

# Processing Attribute Profiles as Scale-series for Remote Sensing Image Classification

Melike Ilteralp<sup>1</sup> <sup>a</sup> and Erchan Aptoula<sup>2</sup> <sup>b</sup>

<sup>1</sup>Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey

<sup>2</sup>Institute of Information Technologies, Gebze Technical University, Kocaeli, Turkey

**Keywords:** Attribute Profiles, Hyperspectral Images, Supervised Classification, Long Short-term Memory Network.

**Abstract:** Attribute profiles (APs) are among the most prominent “shallow” spatial-spectral pixel description methods, providing multi-scale, flexible and efficient pixel descriptions, even with modest amounts of training data. In this paper, we investigate their collaboration with long short-term memory networks (LSTMs). Our hypothesis is that a profile can be viewed as a “scale-series” and LSTMs can exploit their sequential nature, akin to temporal series. Plus, feeding a deep network with input of already strong descriptive potential (such as APs) can help them produce advanced features more efficiently w.r.t. training from scratch. Moreover, contrary to the state-of-the-art, we report the results of experiments conducted with *non-overlapping training and testing sets*, highlighting a significant boost of performance through the combined use of APs with LSTMs.

## 1 INTRODUCTION

Pixel-level classification with the end goal of land use/cover mapping constitutes the basis of several remote sensing applications. Despite numerous reported pixel description methods, it represents a long-standing challenge due to constant sensor technology advances. Ever-increasing spatial, spectral and temporal remote sensing image resolutions have amplified the need for efficient and effective spatial-spectral pixel description and classification approaches (Landgrebe, 2003).

Deep learning is known for its state-of-the-art performance across many domains, including remote sensing (Audebert et al., 2019). However, its performance is tightly bounded by its need for a large number of labeled samples as well as for high label precision. In addition, labeled remote sensing datasets are relatively scarce w.r.t. computer vision, since generating them is an arduous and expensive task. Consequently, it is not surprising that shallow approaches are still widely employed, as in the recent IEEE GRSS Data Fusion Contest (Yokoya et al., 2020).

At the front of “shallow” descriptors, Morphological Attribute Profiles (APs) stand out as a prominent approach, even with modest amounts of training data.

By exploiting the hierarchical tree-based representations of an input image, they produce multi-scale, efficient object-based pixel descriptions, w.r.t. arbitrarily chosen criteria (Dalla Mura et al., 2010b). They have been studied extensively in the past decade, in terms of alternative tree representations (Bosilj et al., 2017; Cavallaro et al., 2016), threshold selection techniques (Bhardwaj et al., 2019; Derbashi and Aptoula, 2020), pre-processing (Dalla Mura et al., 2011) and post-processing extensions (Pham et al., 2018). Nevertheless, APs have their inconveniences too, as they possess notoriously difficult to set parameters such as thresholds and attributes.

An in-between approach to shallow and deep pixel description has appeared as their combination in an effort to harness the advantages of both strategies. It consists in providing as input to relatively small deep networks, easy to calculate shallow features; examples include the combination of Gabor filtered (Chen et al., 2017) and attribute filtered images with convolutional neural networks (CNNs) (Aptoula et al., 2016). The underlying motivation is to avoid training from scratch (and hence circumvent the need for large training sets), by starting from mid-level features and to produce more advanced outputs through the ability of deep networks.

In this paper, we explore further the aforementioned pixel description paradigm of combining shallow and deep strategies. More specifically, since a

<sup>a</sup>  <https://orcid.org/0000-0003-2685-914X>

<sup>b</sup>  <https://orcid.org/0000-0001-6168-2883>

pixel's values across an attribute profile constitute a numerical sequence across scales, it can be considered as a "scale series". Consequently, we have selected a long short-term memory (LSTM) network in order to exploit its sequential nature, akin to temporal series analysis (Hochreiter and Schmidhuber, 1997).

Our second contribution addresses a validation mis-conduct encountered very often in the AP related literature, where a single tree representation is calculated from the entire scene of the ground truth. As a result, the same tree encodes the pixels of both training and testing sets, leading very possibly to pixels of the same connected components used both for model development and validation. Evidently, this can lead to unrealistically high classification performances and deceptively high generalization impressions (Audebert et al., 2019).

To avoid this, we have conducted experiments with two real-world datasets, where non-overlapping training and test subsets are used, or in other words, distinct tree representations are calculated for training and testing. The results confirm this performance discrepancy, while the proposed combination of APs with LSTMs achieves significantly higher generalization performance.

## 2 PROPOSED APPROACH

This section will first recall the basics of APs, hierarchical tree representations and LSTMs, and then elaborate on the proposed approach for pixel description.

### 2.1 Attribute Profiles

APs are efficient, multiscale image descriptors that have been introduced in order to overcome the structuring element related limitations of the older morphological profiles (Dalla Mura et al., 2010b). APs are based on attribute filters (AFs) which are morphological connected filters that either preserve or remove connected components (CC) of grayscale images via checking whether that CC satisfies a binary predicate. The predicate is typically a comparison of a CC attribute (e.g. area, elongation, etc) against a numerical threshold value. APs consist of a series of images that are generated via stacking the outputs of AFs using progressively increasing threshold values.

More formally, let  $f : D \rightarrow \mathbb{Z}$  be a grayscale image where  $D \subseteq \mathbb{Z}^2$ . Let  $T$  be a binary predicate and  $\{\kappa_i\}_{1 \leq i \leq L}$  a set of  $L$  thresholds. Let  $\gamma^{\kappa_i}$  and  $\phi^{\kappa_i}$  be the morphological thinning and thickening filters with threshold  $\kappa_i$ . The AP of  $f$  is obtained as follows:

$$AP(f) = \{\phi^{\kappa_L}(f), \phi^{\kappa_{L-1}}(f), \dots, \phi^{\kappa_1}(f), f, \gamma^{\kappa_1}(f), \dots, \gamma^{\kappa_{L-1}}(f), \gamma^{\kappa_L}(f)\} \quad (1)$$

Extended attribute profiles (EAPs) (Dalla Mura et al., 2010a) represent the extension of APs to hyperspectral images. EAPs are generated via applying a dimensionality reduction technique to a hyperspectral image  $\mathbf{f}$  and then stacking the calculated AP of each remaining band:

$$EAP(\mathbf{f}) = \{AP(band_1), AP(band_2), \dots, AP(band_n)\} \quad (2)$$

In both cases (AP and EAP), individual pixels are described with the sequence of values they acquire across the filtered images. In addition, the spatial detail level associated with each value in this sequence decreases as the employed threshold value increases. Hence, an AP can be considered as a pixel-level scale-series.

### 2.2 Hierarchical Tree Representation

Even though AF have been known for a long time (Breen et al., 1996), the computational complexity of CC calculation has hindered their widespread use. This changed with the manifestation of hierarchical tree-based image representations (Salembier et al., 1998), where CCs are encoded as tree-nodes. Consequently, it becomes possible to efficiently manipulate entire CCs and perform object based filtering through mere tree pruning operations.

In more detail, the hierarchical representation of an image is formed via stacking the union of image regions at different scales (Bosilj et al., 2018). The tree structure stands forward as a prominent hierarchical image representation since each tree node corresponds to an image region and the parent-child relation between the tree nodes indicates the inclusion relation of the image regions of different scales. In this tree structure, which is called a component tree, the leaf nodes correspond to the finest regions of the image and the region covered by a node grows as moving higher (towards the root node) in the tree where the root node corresponds to the whole image. There are two types of component tree categories, the inclusion trees and the partition trees where the distinction between them is how the hierarchy is built (Bosilj et al., 2018).

The inclusion trees (such max/min trees) consist of the partial partition of the image and the leaf nodes correspond to the small regions of the image such as local minima or maxima. The new nodes are formed via including the same intensity pixels (flat zones) to the leaf nodes and this inclusion of pixels forms the

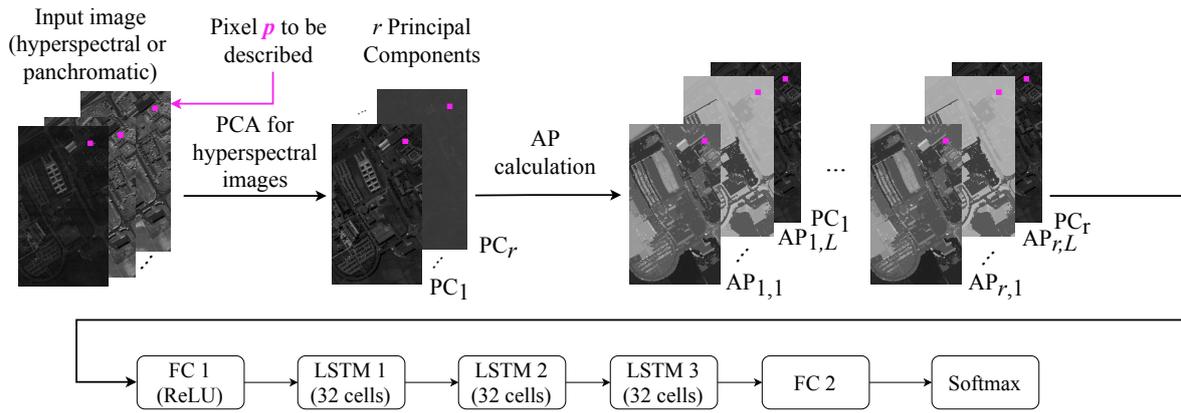


Figure 1: Outline of our pixel description strategy and the architecture of the model where PCA, PC, FC and ReLU, denote principal component analysis, principal component image, fully connected layer and rectified linear unit respectively.

parent-child relationship between them. Inclusion of the pixels to the nodes and hence the formation of the new nodes continues until the root node where the root node corresponds to the whole image.

The partition trees (such as  $\alpha$ -trees, binary partition trees) comprise the full partition of the image and the leaf nodes containing the finest image regions are merged as moving higher in the tree where the root node corresponds to the whole image (Bosilj et al., 2017). The partition at any level of the partition tree represents the whole image.

### 2.3 Long Short-Term Memory Networks

LSTMs are known to achieve state-of-art performance with various sequential data tasks (Hochreiter and Schmidhuber, 1997; Ma and Hovy, 2016; Sogaard and Goldberg, 2016). A LSTM is a specialized recurrent neural network (RNN) architecture that is designed to process sequential data. RNN uses the information of previous events to make inference about the future event along with the current one by retaining the past knowledge with its chain-like structure. However, RNNs suffer from the inability to learn long-term dependencies between the events with large time gaps, due to vanishing gradients. LSTMs have been in fact introduced to address specifically this issue.

In more detail, a LSTM cell receives a sequential data sample  $x_t \in \mathbb{R}^n$ , a hidden state  $h_{t-1}$  and the cell state  $C_{t-1}$  of the previous cell as input at time  $t$  and calculates the current cell state  $C_t$  and hidden state  $h_t$  with forget, input and output gates within it. The forget gate  $f_t$  calculates how much of the previous and current information will be kept. The input gate  $i_t$  calculates which values are important for updating the

cell state  $C_t$ . The output gate  $o_t$  calculates the current hidden state  $h_t$ . Formally, the output of a LSTM cell at time  $t$  is calculated as follows:

$$\begin{aligned}
 f_t &= \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \\
 i_t &= \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \\
 o_t &= \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \\
 \tilde{C}_t &= \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \\
 C_t &= f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \\
 h_t &= o_t \circ \tanh(C_t)
 \end{aligned} \tag{3}$$

where  $\sigma$  denotes the sigmoid function,  $\cdot$  denotes the matrix multiplication,  $\circ$  denotes the dot product,  $W_*$  and  $U_*$  are weight matrices and  $b_*$  represent the bias terms specific to the gates or cell state (Hochreiter and Schmidhuber, 1997).

### 2.4 APs as Input to LSTMs

As explained in the previous section, an AP based pixel feature vector constitutes a numerical sequence. Each sequence element describes the pixel in terms of an arbitrary pre-selected attribute (e.g. area) of the CC containing it, at a certain spatial scale. And the scale varies depending on the used threshold. Furthermore, the values across an AP have no particular reason for being independent of each other since they are generated via filtering the same input image. Therefore, a neural model that can capture the eventual dependencies within this scale series can potentially contribute to classification performance.

However, convolutional neural networks don't treat data as a sequence and therefore, cannot exploit "past" information. LSTMs on the other hand, stand out as prominent DL architectures that can receive sequential data as input and capture eventual dependencies within it.

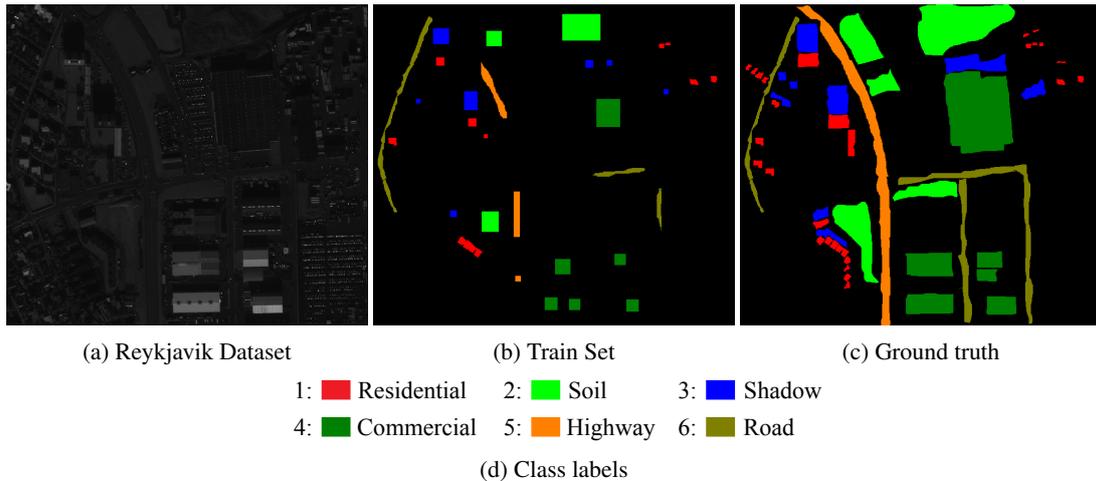


Figure 2: The Reykjavik dataset, training set and ground truth, as often encountered in the state-of-the-art.

Moreover, the combination of APs with deep learning (DL) techniques can also help the latter by alleviating their need for a large amount of labeled data, since APs constitute mid-level features and can thus ease training by not forcing it to start from scratch (Aptoula et al., 2016; Chen et al., 2017).

To achieve our goal, we adapted the LSTM model of (Chevalier, 2016) which predicts the labels of sequences in a many-to-one manner that is similar to our problem where each sequence of pixel description has a label. The proposed architecture consists of 3 LSTM layers where each of them has 32 hidden units. The choice of architectural hyperparameters is detailed in Section 3.2.

The network architecture is not tuned for any one dataset/tree representation since it's intended for general use. The outline of the proposed pixel description strategy and the classification model is shown in Figure 1.

### 3 EXPERIMENTS

The aim of the experiments performed in this section is to test our hypothesis that the combination of APs, as mid-level features, with LSTMs yields superior generalization performance through the exploitation of their sequential nature as scale-series.

#### 3.1 Datasets

The experiments have been conducted on two real remote sensing datasets, one being hyperspectral and the other being a panchromatic dataset, in order to show that our approach works for both types. The

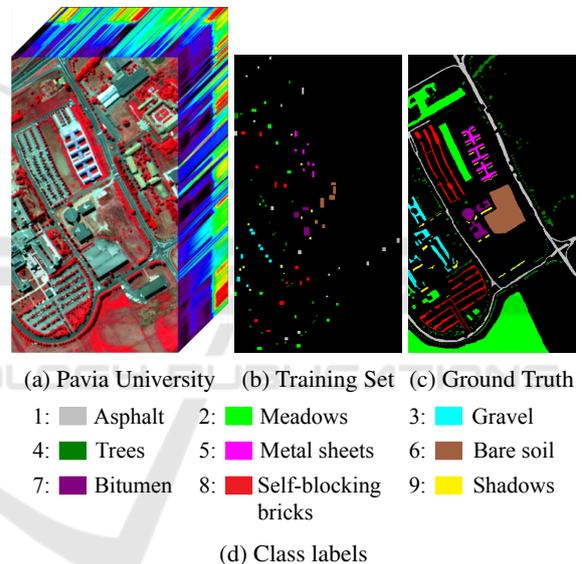


Figure 3: The Pavia University dataset, training set and ground truth, as often encountered in the state-of-the-art.

first one is the Pavia University dataset (Figure 3), acquired by the ROSIS-03 sensor over Pavia, Italy. It is an urban area consisting of 103 spectral bands, ranging from  $0.43 \mu\text{m}$  to  $0.86 \mu\text{m}$  with 9 classes. It depicts an area of  $610 \times 340$  pixels and the spatial resolution is 1.3 meters. After applying PCA to the dataset, four principal components have been found to account for 99% of the total variance.

The second one is the panchromatic Reykjavik dataset (Figure 2) acquired from Reykjavik, Iceland with the Ikonos satellite. It shows an area of  $628 \times 700$  pixels with 1 meter spatial resolution and 6 classes.

As observed by (Audebert et al., 2019), using the standard train and test splits that are encountered

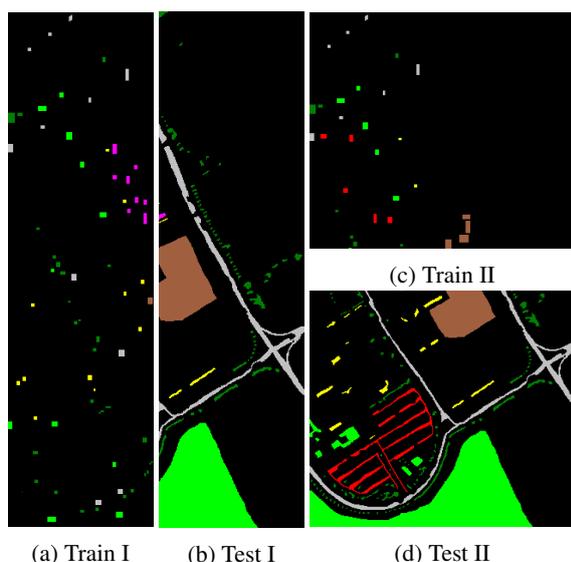


Figure 4: The Pavia University dataset train and test sets, vertical split (a), (b) and horizontal split (c), (d). Note that the samples of the gravel, bitumen and self-blocking bricks classes are eliminated in the vertical split whereas the samples of the gravel, metal sheets and bitumen classes are eliminated in the horizontal split.

widely in the related state-of-the-art, one computes a single tree, containing the nested connected component hierarchy of both training and testing samples. Whereas in actual deployment conditions, the classification models are expected to perform with a distinct tree calculated from the scene to be classified. To overcome this issue, we generated non-overlapping train/test subsets via splitting the ground truth horizontally and vertically into two same-size images. The top/left split is used as the train set and the bottom/left split is used as the test set. The resulting train and test sets depending on split type are shown in Figures 4 and 5 for the Pavia University and

Table 1: The Pavia University dataset, number of class samples in the standard dataset and its horizontal and vertical splits.

Class Id	Labeled Samples per Split					
	Standard		Horizontal		Vertical	
	Train	Test	Train	Test	Train	Test
1	548	6631	210	3499	392	3094
2	540	18649	240	13433	479	9035
3	392	2099	0	0	0	0
4	524	3064	214	1653	524	1176
5	265	1345	0	0	265	43
6	532	5029	372	3530	48	4634
7	375	1330	0	0	0	0
8	514	3682	225	1973	0	0
9	231	947	28	732	191	153

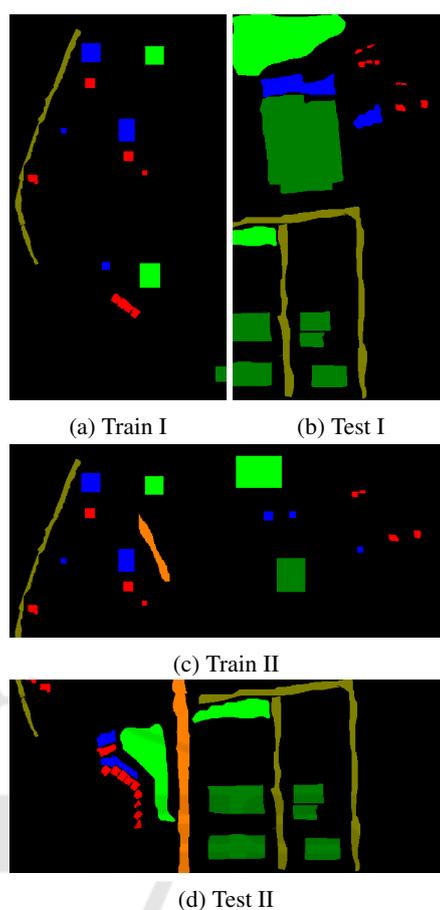


Figure 5: The Reykjavik dataset train and test sets, vertical split (a), (b) and horizontal split (c), (d). Note that the samples of the highway class are eliminated in the vertical split.

the Reykjavik datasets respectively. Afterward, the model is tested on both overlapping (Standard) and non-overlapping (Horizontal and Vertical) train and test subsets.

Some of the classes in the non-overlapping subsets do not have a sufficient number of samples for training or testing due to the split operation. To overcome this situation, the ratio of the number of sam-

Table 2: The Reykjavik dataset, number of class samples in the standard dataset and its horizontal and vertical splits.

Class Id	Labeled Samples per Split					
	Standard		Horizontal		Vertical	
	Train	Test	Train	Test	Train	Test
1	1863	6213	1195	1875	1446	669
2	6068	28144	4748	8885	2220	14419
3	2619	10610	2450	1278	2173	4078
4	5599	29768	2530	11893	450	27465
5	2489	12051	1300	5509	0	0
6	4103	11940	2147	9784	2828	8558

ples of a class w.r.t. the total number of samples in the ground truth was enforced to be at least 0.1%. As a result, the samples of the classes gravel, bitumen and self-blocking bricks in the vertical split and the samples of the classes gravel, metal sheets and bitumen in the horizontal split of the Pavia University dataset have been eliminated. Likewise, samples of the highway class in the vertical split of the Reykjavik dataset have been eliminated as well. The resulting number of samples per class depending on split type are shown in Tables 1 and 2 for the Pavia University and the Reykjavik datasets respectively.

### 3.2 Implementation Details

The area attribute has been selected in this study as a geometric attribute for the sake of comparability reasons with published studies (Dalla Mura et al., 2010b; Bosilj et al., 2017; Cavallaro et al., 2016; Bhardwaj et al., 2019; Derbashi and Aptoula, 2020; Dalla Mura et al., 2011; Pham et al., 2018; Aptoula et al., 2016). The CCs are calculated using two different types of tree representations in the name of comprehensiveness. The first is a component (min-max) tree from the family of inclusion trees and the other is an  $\alpha$ -tree from the family of partition trees. As far as threshold selection is concerned, we used automatically computed thresholds for the Pavia University and Reykjavik datasets as recommended respectively by (Ghamisi et al., 2014) and (Cavallaro et al., 2016):

$$\begin{aligned} \lambda_{Pav} &= \{770, 1538, 2307, 3076, 3846, 4615, 5384, 6153, \\ &\quad 6923, 7692, 8461, 9230, 10000, 10769\} \\ \lambda_{Rey} &= \{25, 100, 500, 1000, 5000, 10000, 20000, \\ &\quad 50000, 100000, 150000\} \end{aligned} \quad (4)$$

The proposed approach has been compared against two alternatives in order to better quantify its effect:

- **AP+RF (without LSTMs):** APs are classified using a Random Forest (RF) classifier as is very often realized in the state-of-the-art.
- **Spectral Signature + LSTM (without APs):** For the hyperspectral Pavia University dataset, a LSTM was trained using its full 103-dimensional spectral signature in order to observe the effect of AP contribution w.r.t. to the plain use of LSTMs.

The RF classifier has been trained with 100 trees. The same hyper-parameters have been used consistently in the LSTM model across all experiments: a learning rate of 0.0025, a batch size of 1500, and a number of epochs of 9440 which are selected through

grid search. We fixed a seed=128 for reproducibility of the results. The time step of the LSTM model is selected as 1 to be fair since the length of a pixel description varies depending on the number of bands of the dataset and the tree representation. The results that have been obtained are shown in Table 3 and Table 4.

### 3.3 Results & Discussion

The following observations can be made from the results of Table 3 and Table 4: when the standard (overlapping) train/test split is used, the RF classifier is observed to achieve a superior performance w.r.t. the LSTM model for both datasets, unless spectral signatures are used. A possible explanation could be that when using the standard train/test splits, many samples of the same connected components end up separated across the train and test sets. A deep network is more prone to overfit w.r.t. RF, especially given the relatively small dataset sizes.

On the contrary, when non-overlapping train/test splits are used, the classification performance of RF drops drastically and LSTM outperforms RF in all non-overlapping image experiments for both datasets. A possible explanation could be that when two pixels of the same class from different sets (one from train set and one from test set) are described by the APs calculated from different trees, these descriptions will be less similar to each other compared to the case where both of them reside in the same tree. As LSTM treats these descriptions as sequential data, not like independent values as RF does, its performance is more robust than RF since it is able to exploit the sequential nature of the information within the input sequence, even though they are constructed using distinct trees. This outcome also confirms the observations of (Audebert et al., 2019).

As verified by the results of the experiments in Table 3 and Table 4, the benefit of combining APs with LSTM is greater than the other approaches for the classification performance on non-overlapping splits.

## 4 CONCLUSION

In this study, we explored the combination of APs with LSTMs in an effort to exploit their sequential nature, that often goes under-exploited via the widespread use of RF. For this purpose, we introduced the collaboration of APs with LSTMs. We also investigated the usage of non-overlapping training and testing sets and proposed a way to overcome their drawback on generalization performance.

Table 3: Pixel classification performances in terms of the kappa statistic and  $F_1$ -score  $\times 100$  for the Pavia University dataset where AP+LSTM is the proposed approach.

Split	Spectral Signature				MinMax-tree				$\alpha$ -tree			
	$\kappa$		$F_1$ -score		$\kappa$		$F_1$ -score		$\kappa$		$F_1$ -score	
	RF	LSTM	RF	LSTM	AP+RF	AP+LSTM	AP+RF	AP+LSTM	AP+RF	AP+LSTM	AP+RF	AP+LSTM
Standard	65.6	<b>71.67</b>	72.9	<b>81.19</b>	<b>87.79</b>	83.94	90.7	<b>90.91</b>	<b>85.04</b>	75.88	<b>88.7</b>	87.04
Vertical	36.83	<b>44.28</b>	52.5	<b>62.91</b>	17.22	<b>49.7</b>	34.5	<b>65.56</b>	15.69	<b>33.49</b>	26	<b>55.06</b>
Horizontal	33.57	<b>42.07</b>	35	<b>68.64</b>	25.08	<b>59.89</b>	39.3	<b>71.49</b>	45.61	<b>60</b>	58.4	<b>74.53</b>

Table 4: Pixel classification performances in terms of the kappa statistic and  $F_1$ -score  $\times 100$  for the Reykjavik dataset where AP+LSTM is the proposed approach.

Split	MinMax-tree				$\alpha$ -tree			
	$\kappa$		$F_1$ -score		$\kappa$		$F_1$ -score	
	AP+RF	AP+LSTM	AP+RF	AP+LSTM	AP+RF	AP+LSTM	AP+RF	AP+LSTM
Standard	<b>76.46</b>	58.26	<b>82.5</b>	57.07	<b>71.56</b>	65.61	<b>77.8</b>	65.34
Vertical	15.73	<b>17.45</b>	<b>25.5</b>	21.86	7.17	<b>22.97</b>	19	<b>39.46</b>
Horizontal	1.38	<b>13.07</b>	17.9	<b>27.26</b>	25.31	<b>28.28</b>	31.4	<b>40.14</b>

We tested our approach on two real remote sensing datasets and APs have been calculated with two different hierarchical tree representations. In all the experiments where the training and testing sets don't overlap, we observed the collaboration of APs with LSTM enabling a significant boost in classification performance w.r.t. using AP or LSTMs alone.

In the future, we intend to address threshold-free APs (Derbashi and Aptoula, 2020) via LSTMs capable of admitting varying length input. This will enable us to provide as input to the networks directly the node sequences from the root node to the node containing the pixel under study, eliminating the need for cumbersome thresholds or filtering.

## ACKNOWLEDGEMENTS

This study has been supported by TUBITAK under Grant 118E258.

## REFERENCES

- Aptoula, E., Ozdemir, M. C., and Yanikoglu, B. (2016). Deep learning with attribute profiles for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 13(12):1970–1974.
- Audebert, N., Le Saux, B., and Lefèvre, S. (2019). Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geoscience and Remote Sensing Magazine*, 7(2):159–173.
- Bhardwaj, K., Patra, S., and Bruzzone, L. (2019). Threshold-free attribute profile for classification of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(10):7731–7742.
- Bosilj, P., Damodaran, B., Aptoula, E., Dalla Mura, M., and Lefèvre, S. (2017). Attribute profiles from partitioning

trees. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 381–392. Springer International Publishing.

- Bosilj, P., Kijak, E., and Lefèvre, S. (2018). Partition and inclusion hierarchies of images: A comprehensive survey. *Journal of Imaging*, 4(2):1–31.
- Breen, E. J., , and Jones, R. (1996). Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding*, 64(3):377–389.
- Cavallaro, G., Dalla Mura, M., Benediktsson, J. A., and Plaza, A. (2016). Remote sensing image classification using attribute filters defined over the tree of shapes. *IEEE Transactions on Geoscience and Remote Sensing*, 54(7):3899–3911.
- Chen, Y., Zhu, L., Ghamisi, P., Jia, X., Li, G., and Tang, L. (2017). Hyperspectral images classification with Gabor filtering and convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2355–2359.
- Chevalier, G. (2016). LSTMs for human activity recognition. <https://github.com/guillaume-chevalier/LSTM-Human-Activity-Recognition>, visited 2020-11-12.
- Dalla Mura, M., Benediktsson, J. A., Waske, B., and Bruzzone, L. (2010a). Extended profiles with morphological attribute filters for the analysis of hyperspectral data. *International Journal of Remote Sensing*, 31(22):5975–5991.
- Dalla Mura, M., Benediktsson, J. A., Waske, B., and Bruzzone, L. (2010b). Morphological attribute profiles for the analysis of very high resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10):3747–3762.
- Dalla Mura, M., Villa, A., Benediktsson, J. A., Chausot, J., and Bruzzone, L. (2011). Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis. *IEEE Geoscience and Remote Sensing Letters*, 8(3):542–546.
- Derbashi, U. and Aptoula, E. (2020). Attribute profiles without thresholds through histogram based tree path

- description. In *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium*, pages 29–32.
- Ghamisi, P., Benediktsson, J. A., and Sveinsson, J. R. (2014). Automatic spectral–spatial classification framework based on attribute profiles and supervised feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 52(9):5771–5782.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Landgrebe, D. A. (2003). *Signal Theory Methods in Multi-spectral Remote Sensing*. Wiley, New York.
- Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany.
- Pham, M., Lefèvre, S., and Aptoula, E. (2018). Local feature-based attribute profiles for optical remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):1199–1212.
- Salembier, P., Oliveras, A., and Garrido, L. (1998). Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570.
- Søgaard, A. and Goldberg, Y. (2016). Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany.
- Yokoya, N., Ghamisi, P., Haensch, R., and Schmitt, M. (2020). 2020 IEEE GRSS Data Fusion Contest: Global land cover mapping with weak supervision [technical committees]. *IEEE Geoscience and Remote Sensing Magazine*, 8(1):154–157.