An Optimized Superpixel-based Domain Adaptation Scheme for River Ecosystem Classification

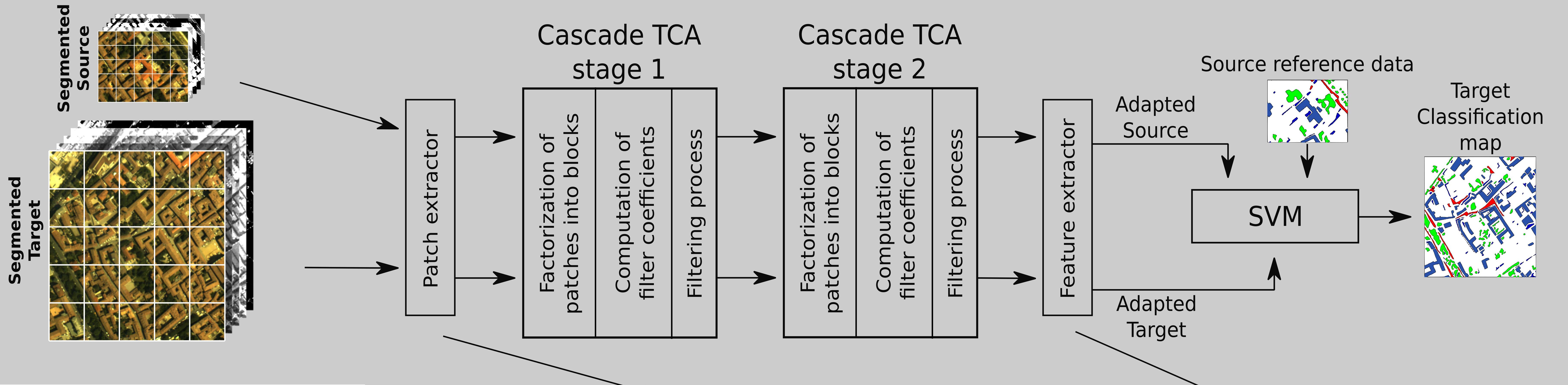
Alberto S. Garea[1], Dora B. Heras[1], Francisco Argüello[2], and Begüm Demir[3]

[1]Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)
[2]Universidade de Santiago de Compostela
[3]Technische Universität Berlin, Berlin

22th International Conference Computational and Mathematical Methods in Science and Engineering — 2022 CMMSE

The availability of new multispectral sensors capable of capturing high resolution images through low altitude flights using drones, provides access to large amounts of information of the Earth Surface at a much lower cost than images captured by other devices. One of the most common tasks performed over those images, in particular, in river ecosystems, is the supervised classification in situations with a scarce number of samples. Domain Adaptation (DA) helps in the classification problem by allowing the classification of images using information from another different image captured by the same sensor over a different location. TCANet is a scheme for unsupervised DA that simulates the behavior of a convolutional network but for which the computation of the filter coefficients is performed directly through TCA, a kernel-based feature extraction technique specially designed for DA. The high computational cost of TCA together with the large size of the high resolution datasets makes the use of both parallelization techniques and the application of spatial information extraction algorithms indispensable to solving the problem. In this paper, we propose an optimized superpixel-based DA technique for river ecosystem classification using high-resolution multispectral images.

Optimized Domain Adaptation Classifier based on TCANet



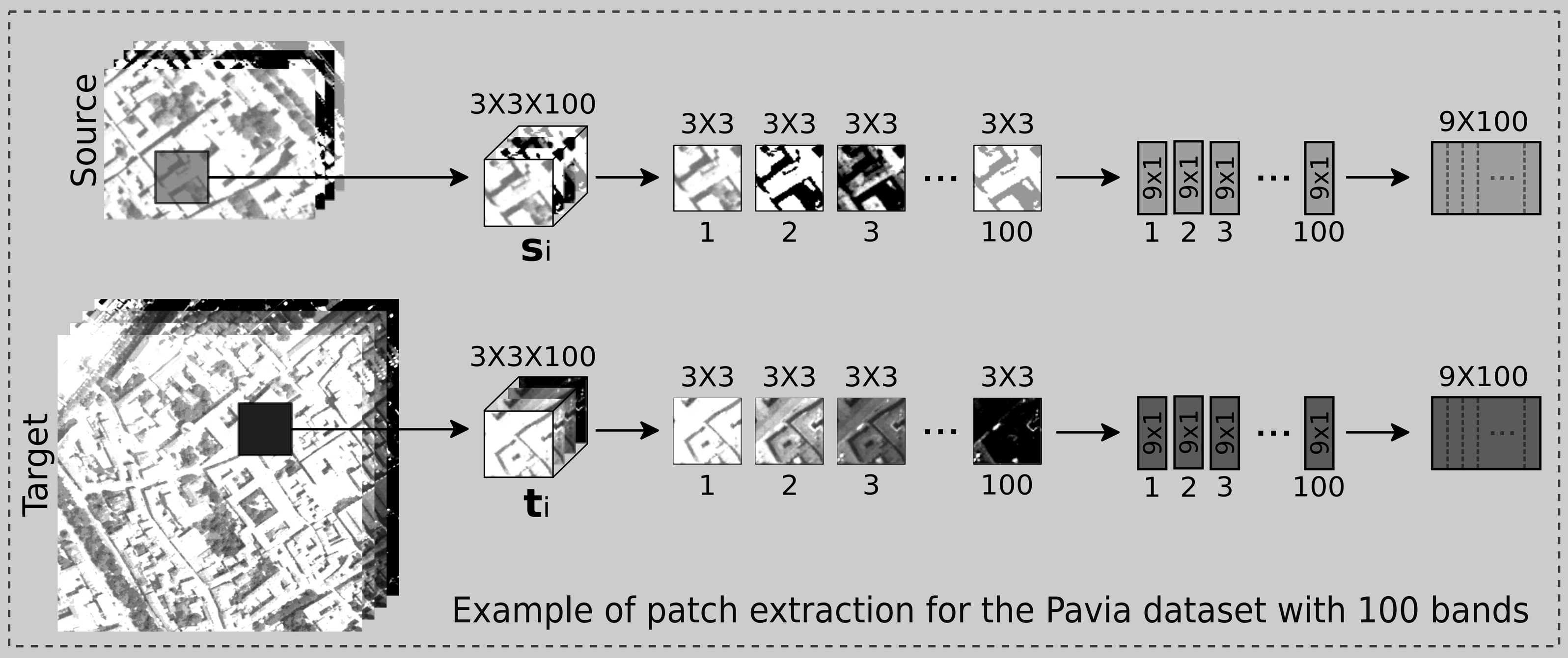
Algorithm 1: Cascade TCA stage 1

Input: Source and target extracted patches
Output: Convolution stage 1
Parameters:
- $H \times V$: patch size
- N_1: number of convolution filters
- $F_1 \times F_2$: block size of filters

1. Preprocessing
- 1.1 Extract blocks from each patch
- 1.2 Join source and target extracted data

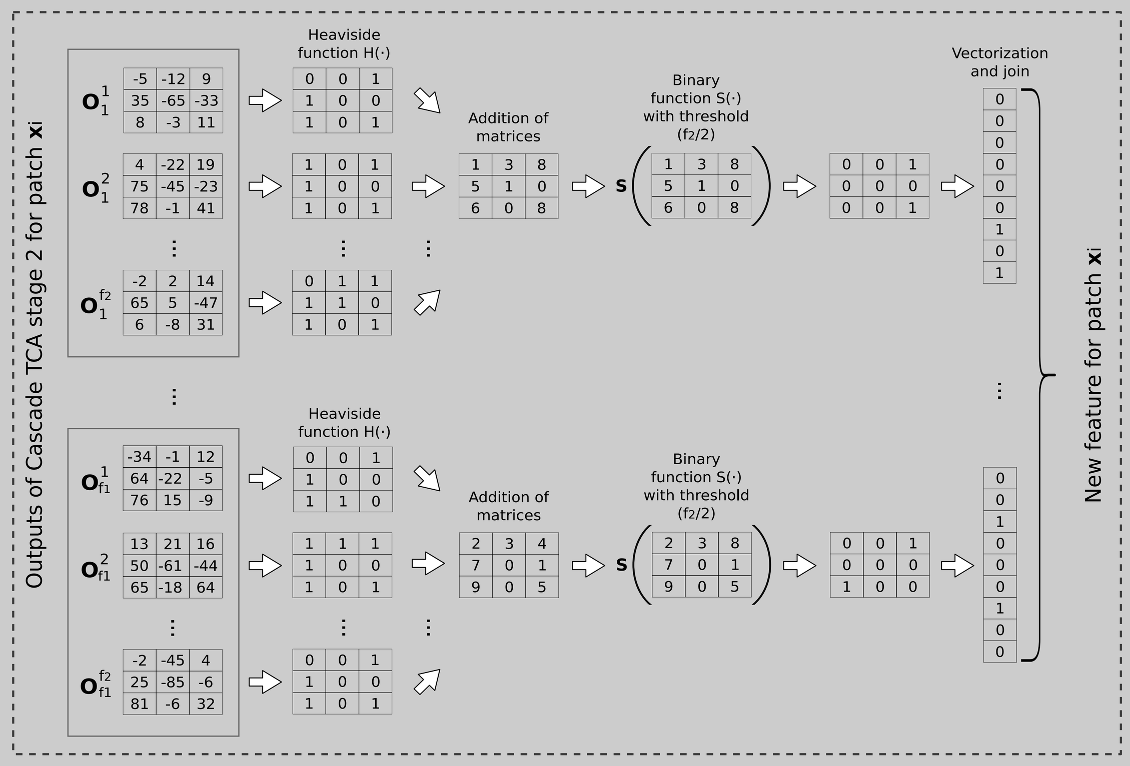
2. Compute TCA filter
- 2.1 Normalize input data
- 2.2 Compute eigenvectors

3. Convolutional layer
- 3.1 For each patch
  - 3.1.1 Extract blocks
  - 3.1.2 Multiply by the filters

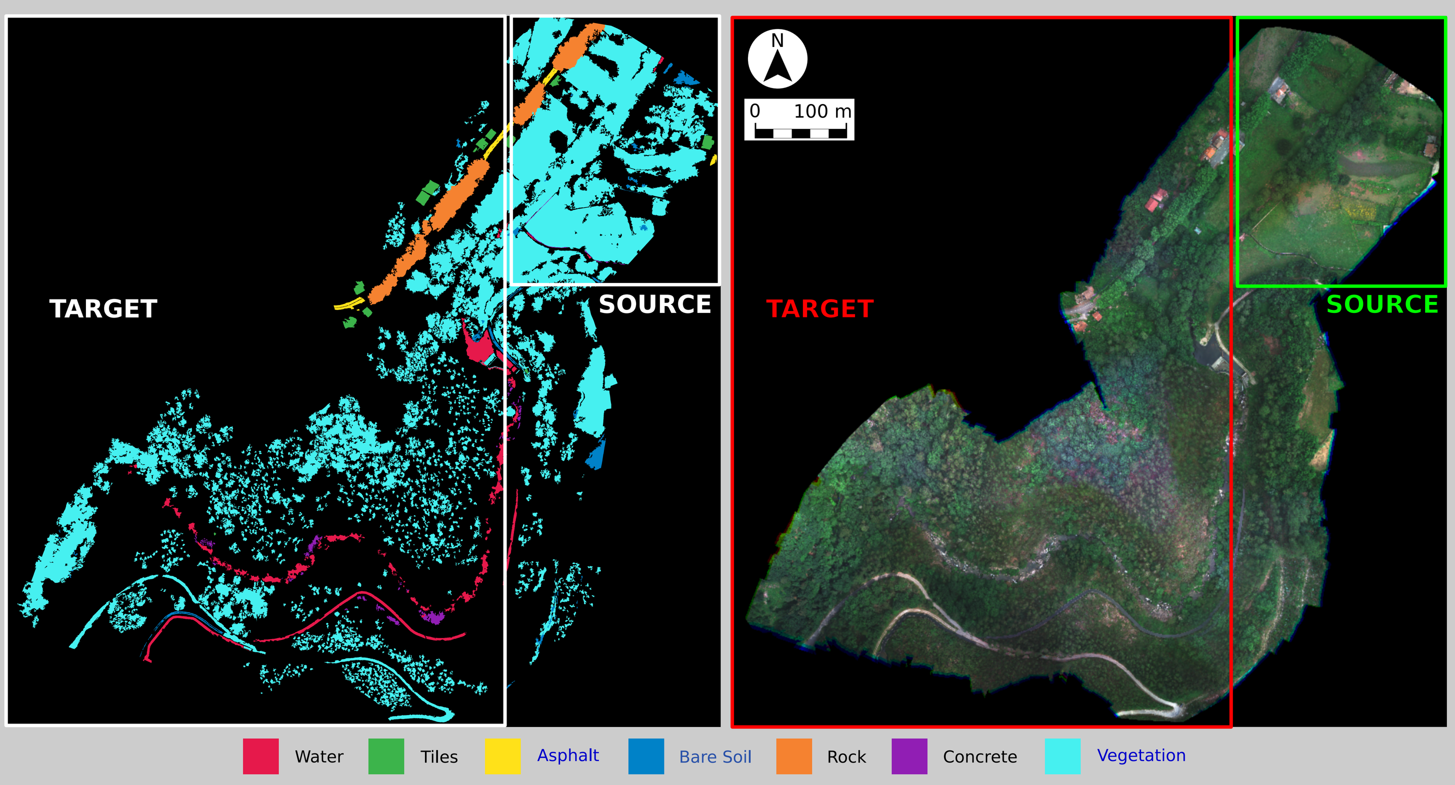


Example of patch extraction for the Pavia dataset with 100 bands

Datasets	Parameters						
	H	V	N_1	F_1	F_2	f_1	f_2
Pavia	3	3	2	9	50	2	16
Oitaven	3	3	2	9	5	2	16



Datasets



Oitavén River

Class	Source-Pixels	Target-Pixels
Water	3944	283219
Tiles	16787	61093
Asphalt	11845	27413
Bare Soil	32417	29898
Rock	141373	233494
Concrete	5414	40767
Vegetation	2249483	2859168

Sensor: MicaSense RedEdge multispectral camera mounted on a custom UAV.
- Spatial resolution → 8.2cm/pixel
- Height → 120m
- Spectral bands → 475 nm (Blue), 560 nm (Green), 668 nm (Red), 717 nm (Edge), and 840 nm (NIR)

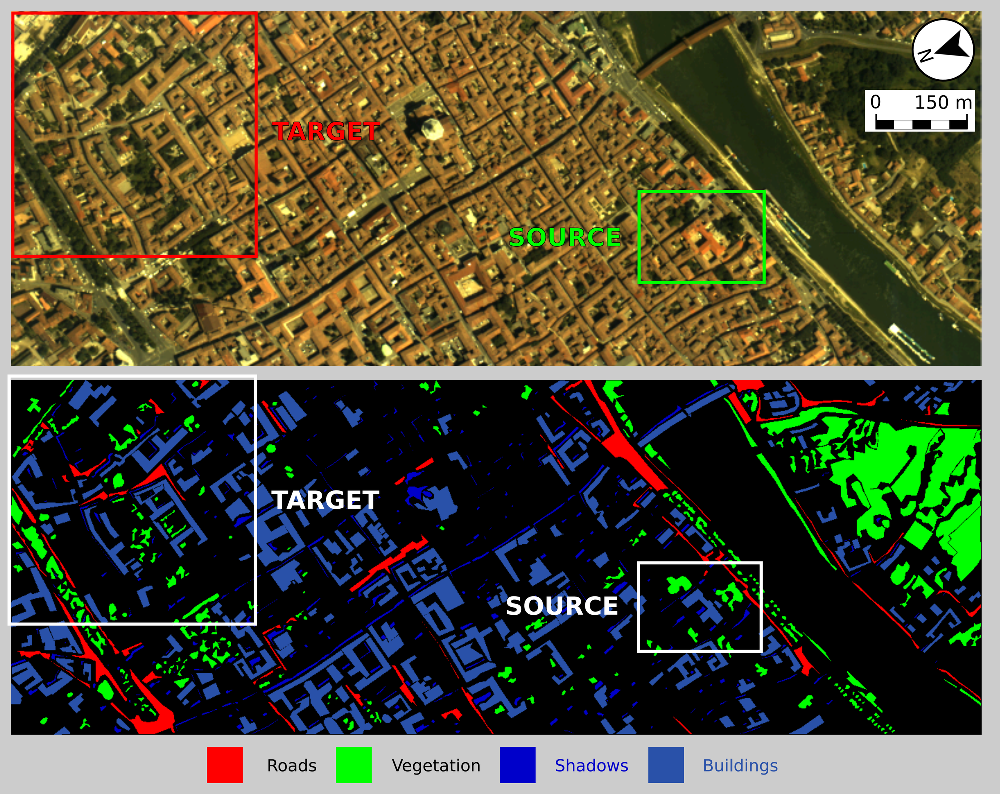
Dimensions:
- Source → 2500 x 1898
- Target → 6689 x 4701

Pavia

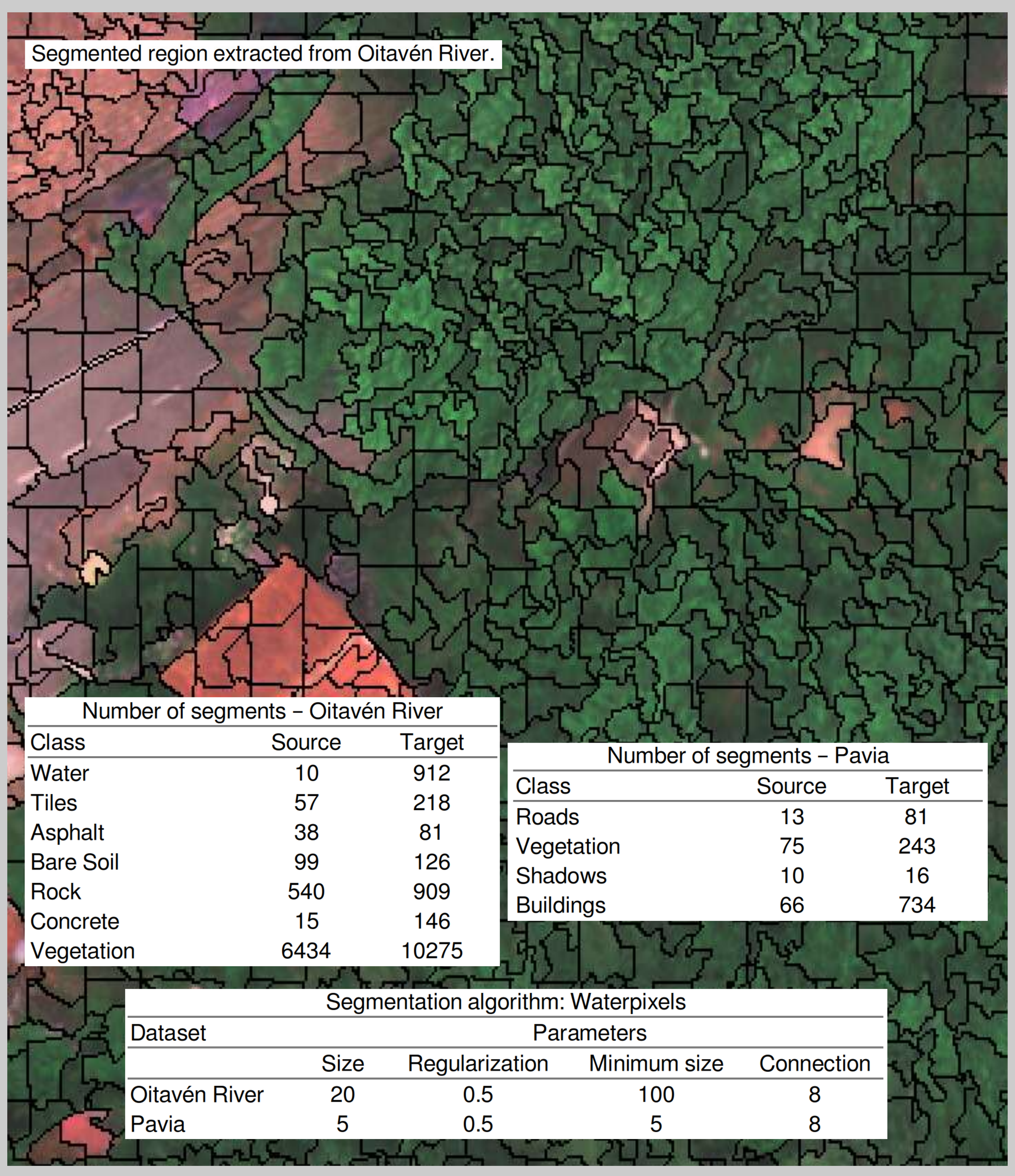
Class	Source-Pixels	Target-Pixels
Roads	326	2549
Vegetation	1793	6406
Shadows	514	1638
Buildings	1465	17501

Sensor: Rosis-03.
- Angular field of view (FOV) → 16°
- Instantaneous field of view (IFOV) → 0.56 mrad
- Number of pixels per line → 512
- Scan principle → Pushbroom
- Ground resolution → 1m – 6m
- Radiometric resolution → 14 bits
- Spectral range → 430 nm – 800 nm
- Spectral sampling → 4 nm
- Inflight calibration → 0.2 nm

Dimensions:
- Source → 124 x 173
- Target → 350 x 350



Segmentation



Number of segments – Oitavén River		
Class	Source	Target
Water	10	912
Tiles	57	218
Asphalt	38	81
Bare Soil	99	126
Rock	540	909
Concrete	15	146
Vegetation	6434	10275

Number of segments – Pavia		
Class	Source	Target
Roads	13	81
Vegetation	75	243
Shadows	10	16
Buildings	66	734

Segmentation algorithm: Waterpixels				
Dataset	Parameters			
	Size	Regularization	Minimum size	Connection
Oitavén River	20	0.5	100	8
Pavia	5	0.5	5	8

Speedup results and experimental setup

Initial version, segment version, and C++ segment version execution times in seconds for Oitavén River and Pavia datasets.

Step	Oitavén River					Pavia				
	Initial pixel version	Segment version	Speedup	C++ segment version	Speedup	Initial pixel version	Segment version	Speedup	C++ segment version	Speedup
Reading datasets	3.575	3.420	1.04x	1.196	2.86x	0.461	0.427	1.08x	0.123	3.47x
Normalize datasets	1.260	1.266	0.99x	0.323	3.92x	0.16	0.152	1.05x	0.020	7.6x
Extracting source train patches	0.478	0.180	2.65x	0.001	180x	0.162	0.034	4.76x	0.001	34x
Extracting target train patches	1.227	0.926	1.32x	0.001	926x	0.298	0.097	3.07x	0.001	97x
TCA stage 1	9.404	0.280	33.58x	0.138	2.02x	1.255	0.175	7.17x	0.028	6.25x
Filtering stage 1	2.416	0.227	10.64x	0.037	6.13x	5.430	0.781	6.95x	0.502	1.55x
TCA stage 2	30.756	0.577	53.30x	0.279	2.06x	1.516	0.153	9.90x	0.067	2.28x
Filtering stage 2	940.38	98.368	9.55x	0.035	**2810.07x**	56.328	8.783	6.41x	0.939	**9.35x**
SVM train	59.755	1.277	46.79x	0.870	1.46x	4.169	0.439	9.49x	0.146	3x
SVM test – Feature extraction	81647.3	583.71	139.87x	0.384	**1520.07x**	1649.279	59.935	27.51x	10.973	**5.46x**
SVM test - Prediction	6487.3	2.839	2285.06x	0.543	5.22x	29.701	0.439	67.65x	0.078	5.62x
Total	89183.851	693.07	128.67x	3.807	**182.05x**	1748.759	71.415	24.48x	12.878	**5.54x**

Experimental setup

- CPU Intel i5-6600 quad-core:
  - 3.3GHz.
  - 64GB of RAM.
- Ubuntu 18.04 LTS.
- gcc 7.5.0.
- OpenMP 4.0.
- Matlab 2015

	Class	Train Source-pixels	Train Target-pixels	Test Target-pixels	Train Source-segments	Train Target-segments	Test Target-segments
Oitavén River	Water	2000	2000	140609	10	10	902
	Tiles	2000	2000	29546	57	30	188
	Asphalt	2000	2000	12706	38	30	51
	Bare Soil	2000	2000	13949	99	30	96
	Rock	2000	2000	115747	540	60	849
	Concrete	2000	2000	19383	15	10	136
	Vegetation	2000	2000	1428584	6434	100	10175

	Class	Train Source-pixels	Train Target-pixels	Test Target-pixels	Train Source-segments	Train Target-segments	Test Target-segments
Pavia	Roads	200	200	1174	13	20	61
	Vegetation	200	200	3103	50	20	223
	Shadows	200	200	719	10	5	11
	Buildings	200	200	8650	50	20	714

References

- S. Garea, A., Heras, D. B., & Argüello, F. TCANet for domain adaptation of hyperspectral images. Remote Sensing. 2019. 11(19), 2289.
- Tuia, D.; Persello, C.; Bruzzone, L. Domain adaptation for the classification of remote sensing data: An overview of recent advances. IEEE Geosci. Remote Sens. Mag. 2016, 4, 41–57.
- Chan, T.H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A simple deep learning baseline for image classification? IEEE Trans. Image Process. 2015, 24, 5017–5032.
- Pan, S.J.; Tsang, I.W.; Kwok, J.T.; Yang, Q. Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. 2011, 22, 199–210.

Conclusions

- A new optimized version of the TCANet algorithm for DA applied to classification of multi and hyperspectral images is proposed.
- The optimized version focuses mainly on exploiting spatial information by using a superpixel based initial segmentation of the datasets that greatly reduces the execution time by extracting patches centered on segments.
- The code has also been optimized by using BLAS and LAPACKE functions to improve the performance of matrix operations.
- Experiments were performed over images with different size and resolution. The highest speedups (182x) are obtained for the largest image with a highest spatial resolution.