This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

International Journal of Remote Sensing
Vol. 00, No. 00, 00 Month 201X, 1–15

RESEARCH ARTICLE

HypeRvieW: An Open Source Desktop Application for Hyperspectral Remote Sensing Data Processing

Alberto S. Garea[a], Álvaro Ordóñez[a], Dora B. Heras[a]*

and Francisco Argüello[b]

[a] *Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), Universidade de Santiago de Compostela*;
[b] *Departamento de Electrónica e Computación, Universidade de Santiago de Compostela*;
(*Received 00 Month 201X; final version received 00 Month 201X*)

In this paper, we present a desktop application for the analysis, **reference data** generation, **registration and supervised** spatial-spectral classification of hyperspectral remote sensing images through a simple and intuitive interface. Regarding the classification ability, **the different classification schemes are implemented by using a chain structure as a base. It consists of five configurable stages that must be executed in fixed order: preprocessing, spatial processing, pixel-wise classification, combination and postprocessing.** The modular implementation makes its extension easy **by adding new algorithms for each stage or new classification chains**. **The tool has been designed as a platform which is open to the incorporation of algorithms by the users interested in comparing classification schemes. As an example of use** a classification scheme based on the Quick Shift (QS) algorithm for segmentation and on Extreme Learning Machines (ELM) or Support Vector Machines (SVMs) for classification is also proposed. The application is license-free, runs on the Linux operating system and **was** developed in C language using the GTK library, as well as other free libraries to build the graphical user interfaces.

Keywords: tool, remote sensing, classification, registration, hyperspectral data

1. Introduction

The wide availability of satellite and airborne imagery available from different sources has turned remote sensing image analysis and interpretation in a very active research and work field (van der Meer et al. 2012; Landgrebe 2005). The most widely used commercial software tools for Geographical Information Science (GIS) and, in particular for remote sensing, are Ecognition and ENVI. Both allow to perform the most relevant tasks associated to the analysis of geospatial images such as fusion of data provided by different sensors, segmentation or classification.

In the case of freely available and open source software the website FreeGIS.org currently lists 11 remote sensing software tools. Some of the free and open source software can provide more functionality than proprietary low-end GIS products (Steiniger and Bocher 2009). One of the most complete free and open source options

*Corresponding author. Email: dora.blanco@usc.es

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

is GRASS GIS (Neteler et al. 2012), a desktop application used for management and analysis, image processing, generation of graphs and maps, spatial modeling, and visualization, which has more than 350 modules for rendering images and maps. It is supported by the OSGeo foundation as in the case of gvSIG (Anguix and Carrión 2005). gvSIG is a large project in terms of financial and development resources founded by the Regional Council for Infrastructures and Transportation of Valencia (Spain) that began in 2003 and that is focused on viewing, editing and analysis capabilities. Other approaches can be integrated with commercial tools, for example, imageS-VM (Janz et al. 2007), that performs classification by Support Vector Machine (SVM) (Melgani and Bruzzone 2004), and ImageRF (Waske et al. 2012), that focuses on the classification and regression analysis with Random Forests. Both tools are available in EnMAP-Box, a platform-independent software that can be integrated with ENVI. Another freely available tool is MultiSpec (University 2014), a multiplatform software package that also allows the analysis of medical image data. More recently (Bernabé et al. 2013) introduced a web-based application to classify images using methods such as K-Means and Isodata that is validated over images from the repository provided by Google Maps among others. The tool offers good results but only two methods for classification are available.

All tools mentioned above are very complete for analyzing remote sensing data from different sources. Nevertheless, they do not provide capabilities for including the user own developments related to spectral-spatial classification in order to try different configurations and to compare them. In this work we propose a free desktop application for hyperspectral remote sensing image processing that allows to inspect the data in detail, to build **reference data** for classification, to register pairs of hyperspectral images, and to perform segmentation or classification of an image.

In the application different **supervised** spatial-spectral classification schemes for remote sensing hyperspectral data can be applied by using **chains** consisting each one of a different ordering of configurable stages. **In this momente one chain structure of five stages that must be executed in a predefined order is implemented. The modularity of the tool makes it easy for the user to introduce his own algorithms for a stage of the classification chain.** The tool will also offer **in the future** new classification chains.

The remainder of this paper is organized as follows. Section 2 describes the methodology applied for the system development as well as the architecture of the application, including the relevant modules. Section 3 describes the current capabilities of the application. Details of the classification chain and its different stages are described in Section 4. Some examples of GUI use are described in Section 5, where a new classification scheme is also proposed and evaluated using the tool. Finally, Section 6 concludes with some remarks.

2. System Architecture

[PLEASE INSERT FIGURES 1 AND 2 HERE]

This section describes first the methodology adopted for the development of the system identifying the main tasks required for the implementation. Then, the

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

system architecture is detailed.

This is a license-free desktop application, runs on the Linux operating system and **was** developed in C language using the GTK library to build the graphical user interfaces. Regarding the methodology, **four** main tasks required for the implementation of the tool were identified, as it can be observed in Fig. 1. The first task is data input. **The system works with images that must be in one of the** three following storage formats: raw, MATLAB or the new proposed format called *hrw*. If the user wants to classify the image, **reference data for supervised classification** are necessary and they could come from a public repository or be generated by the user through a module of the application.

Another relevant task is to interact with the user through a Graphical User Interface (GUI). The GUI is intuitive and allows an inexperienced user to perform all the tasks available in the application. Different examples of GUI use are described in the next section.

The image processing task can be split in two, analysis and classification. By analysis we refer to all the processes involved in inspecting the images, rotating or scaling them, **producing classification statistics** and generating **reference data**. Meanwhile, the classification tasks process the dataset following the different available algorithms that the user chooses for each stage of the classification chain including algorithms for registering pairs of images. **Finally, the last task is data saving that stands for result saving including classification results, registered images, segmented images or reference data, depending on the user selection. All the results can be later reloaded.**

The architecture of the system is displayed in Fig. 2. Regarding the GTK library, the GTK modules are the following. The module named analysis tools provides different tools to explore the image data. There are tools for registering pairs of images, analyzing the image band by band, saving individual bands, performing zoom of a band, analyzing and comparing pixel-vectors of the image and generating **reference data. In addition, statistics on a user defined region of the classification map are displayed.** The classification tools module allows the user to choose the configuration for the different stages of the chain including the possibility of discarding any of the stages. It also executes the classification stages. There is the possibility of loading results of a previous classification or saving the results of a new classification. Each algorithm selected for a particular stage is loaded as an independent library.

The system libraries module includes PANGO and CAIRO to show graphical and text information respectively, Glib to effectively manage data within C language programs, and Zlib for compressing data. All the GTK modules need the *Main library*, which provides several functions to access and manage the data.

Finally, the module called segmentation-classification algorithms includes the algorithms that are options for each stage of the classification chain. In particular, the **chain available** in the system will be described in Section 4.

The modular structure of the application simplifies the comparison among classification schemes by allowing the addition of new algorithms for each stage of the classification chain, new stages and new complete classification chains. The same can be applied to the analysis capabilities of the application.

3. Capabilities of the tool

The application provides capabilities mainly related to data exploration and analysis in order to numerically and visually inspect their characteristics, and also

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

related to the segmentation and classification of the images:

- **Data loading**. The hyperspectral datasets must be loaded by the user in raw format, MATLAB format or hrw format. The images are characterized by two spatial dimensions and one spectral dimension. This last one is the number of spectral bands corresponding to the reflectance values at different wavelengths. So, each pixel of the image can be considered a pixel-vector with a number of components equal to the number of spectral bands.

  The standard data storage formats for the hyperspectral datasets are raw and MATLAB, which present some drawbacks. In the case of the raw format it requires from the user information on the dimensions of the image, data type and data format when the image is read. In addition, the information is uncompressed, so the files are long. In the case of the MATLAB format, a well known proprietary format by MathWorks, data are compressed but its main drawback is that it is a proprietary format. So, we propose hrw, a format that combines the best characteristics of raw and MATLAB: it is free, it can be read without the need of additional information from the user, and data are compressed.

  Specifically, the hrw format stores first 3 bytes for the three characters hrw identifying the format, then 4 bytes to store the width of the image, 4 additional bytes for the height, and 4 more bytes for the number of bands. The next byte encodes the way in which the data are organized, band wise or pixel-vector wise. The next 4 bytes indicate the size of the remaining compressed data. Finally, the data compressed by Zlib are stored.
- **Inspection of data bands**. The visualization and storage of independent bands of the image is allowed by going directly to the band number, or by sequentially going through the graphical representation of the bands using the mouse.
- **Zoom**. Given the large size of the remote sensing datasets, a zoom of the areas selected by the user can also be performed, which can help, for example, in identifying border irregularities between regions.
- **Rotating and scaling**. The input dataset can be rotated and/or scaled according to an angle and a scaling factor decided by the user.
- **Reference data generation**. **Reference data** of the classification is required for the supervised classification algorithms used in the application. Reference data can be loaded from a file in MATLAB, raw or hrw format or generated by the user. The tool offers a rgb graphical representation of the image that can be zoomed and over which the user defines a new class identified by a color and a label and selects the regions and pixels over the image associated to the class. Reference data is stored in one of the three data formats of the application and it can be reloaded and modified.
- **Pixel spectrum**. In order to compare different pixels of the image, a graphical representation showing the numerical values of the different components of a pixel (vector-pixel information) can be displayed, and the comparison among different pixels shown. This capability is specially useful when the reference data is generated.
- **Registration**. Image registration is a process required before change detection or image compression (Dufaux and Konrad 2000). In our case it consists in finding the adequate values of translation, rotation and scaling parameters in order to align a pair of hyperspectral images. The tool applies the algorithm to the pairs of images selected by the user and produces two registered

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

images as output. The algorithm available in the current version of the tool is described in Section 4.

- **Classification**. The classification module is organized as chains of stages that must be executed in a prefixed order. In the current version of the tool a chain consisting of five stages where the user can choose among different algorithms for each stage is available, as described in Section 4.
- **Classification statistics**. **Once the classification is performed, the user is allowed to graphically select a region of the image in the emerging window of results, and statistics on the classification results are shown. The statistics refer to the number and percentage of pixels in the selected region belonging to each class.**
- **Classification evaluation**. **The quality of the classification is evaluated in terms of execution time**, class accuracy and accuracy over the whole image. The accuracy is evaluated over the labeled samples of the reference data excluding the training samples. The execution time for each stage of the classification chain is shown in the results window and then stored in the output text file. The accuracy is evaluated by using the standard accuracy parameters Class Accuracy (CA), Overall Accuracy (OA), Average Accuracy (AA) and Kappa statistics (K) (Richards and Jia 1999). **The class-specific accuracy represents the percentage of correctly classified pixels for a given class. The OA is the percentage of correctly classified pixels in the whole image and the AA is the mean of the class-specific accuracy for all the classes. Kappa is the percentage of agreement corrected by the amount of agreement that could be expected due to chance alone (Viera, Garrett et al. 2005).**

  Additionally, Quantity Disagreement (QD) and Allocation Disagreement (AD) are also given. They measure the disagreement between a classification map and the reference data in terms of proportion (QD) and spatial allocation (AD) of the classes (Pontius and Millones 2011). In order to compute these values, and to train the unsupervised pixel-wise classifiers, **reference data** in one of the file formats mentioned above must also be loaded. The classification map is shown as a result in order to visually compare to the **reference data for the classification**.

The computational efficiency when transforming the massive amount of remote sensing data into scientific understanding is critical. This is the reason for the widespread use of HPC systems in this area in recent years (Lee et al. 2011). In this regard, one of the future capabilities of the tool will focus on reducing the execution time by providing the user with different parallel and distributed implementations of the algorithms.

4. Classification chain

In this section the modular construction of the spectral-spatial classification schemes is presented. In general, the accuracy of the classification over hyperspectral datasets is improved when spatial information is considered (Fauvel et al. 2013). This can be considered through the use of extended morphological profiles (EMPs) that are adaptive definitions of the neighborhood pixels (Quesada-Barriuso, Argüello, and Heras 2015; Benediktsson, Pesaresi, and Amason 2003). Spectral-spatial kernels (Camps-Valls and Bruzzone 2005) **were** also used with the

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

same objective. Another important group of methods introduce spatial information carrying out a segmentation of the image (Fauvel et al. 2013). The segmentation is generated through partitional clustering (Tarabalka, Benediktsson, and Chanussot 2009) or watershed (Tarabalka, Chanussot, and Benediktsson 2010) among other algorithms. The resulting information is then fused with a classification map producing an improved classification map. An efficient approach for this last option could consist in the five stages graphically displayed in Fig. 3. **In the current version of the application the chain available is the following, where default values are provided for all the parameters required by the algorithms.**

[**PLEASE INSERT FIGURE 3 HERE**]

- **Preprocessing (stage I in Fig. 3). The input to the stage is the hyperspectral original image and the output is a hyperspectral preprocessed image. There are four preprocessing algorithms available in the tool representing four different groups of tasks: denoising, registering, feature reduction, and profile calculation.**

  The algorithm for noise reduction operates in the 2-D spatial domain by using wavelets and soft-thresholding (Quesada-Barriuso, Argüello, and Heras 2014b). Two input parameters are required from the user: "number of stages", that is the number of times that the wavelet is applied, and the "threshold" used.

  The algorithm available for feature reduction is an approach based on Principal Component Analysis (PCA)(Abdi and Williams 2010). The main characteristics of the spectral signature are retained by the first principal components. The parameter required by this algorithms is "number of PCs", the number of components to be retained and joined to the hyperspectral image.

  The profile calculation algorithm available also begins with PCA calculation. Then, the EMP is constructed by performing openings and closings by reconstruction over each one of the components produced by PCA (Quesada-Barriuso, Heras, and Argüello 2013). The parameters required by the algorithm are: "number of PCs", that indicates the number of principal components considered, "number of openings/closings", the number of openings (equal to the number of closings) that are computed from each component, and, finally, "radius", that indicates the radius of the disk used as structuring element for each one of the opening/closing pairs.

  The last available algorithm is an automatic procedure to register two hyperspectral images. In order to reduce the dimensionality of the two input images, a PCA is performed. Three parameters can be decided by the user: "reduction method", "algorithm" and "number of peaks". The parameter "reduction method" allows selecting which information from the original images is used. The first option is "1st PCA", that indicates that only the the first component of each image is retained, the second option is "PCA average" that stands for the average of all components. If the third option

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

"none" is selected, the first spectral band of each image is taken. The parameter "algorithm" allows to select one of the two available Fourier-based methods that can be applied in order to solve the registration. With the first option named "FFT" we refer to a method proposed by Chen (Chen, Defrise, and Deconinck 1994) that uses the log-polar Fourier transform (Fourier-Mellin transform). The other option is "MLFFT" (Pan, Qin, and Chen 2009). The last parameter called "number of peaks" allows deciding the number of highest peaks that are analyzed in the log-polar map after the phase correlation to determine the scaling, rotation, and translation parameters that allow registering the images.

- **Pixel-wise classification (stage II). From the input dataset that comes from the preprocessing step, and using the reference data, a classification map is produced through a pixel-wise classifier. Two supervised classification algorithms are available for this stage: SVM (Melgani and Bruzzone 2004) and ELM (Huang, Wang, and Lan 2011).**

  The SVM algorithm is executed using the LibSVM library (Chang and Lin 2011). For this algorithm the parameters are the following. "Training samples per class" indicates the number of samples per class in the reference data that are randomly selected for training. Different values can be specified for each class. "Kernel" selects the type of kernel used. Depending on the selected kernel different additional parameters must be fixed (Chang and Lin 2011). For the case of the RBF kernel the parameters "C" and "gamma" can be provided by the user or generated by the application by 5-fold cross validation.

  In the case of the ELM algorithm a sigmoid function is used and the required parameters from the user are: "hidden layer neurons", the number of neurons in the hidden layer, and "training samples per class".
- **Spatial processing (stage III). This stage corresponds to a segmentation process. The input is the preprocessed image and the output is the segmentation map. For segmentation two algorithms are available now, although any other algorithm producing a segmentation map can be introduced in the tool.**

  The first option is a watershed algorithm (Vicent and Soille 1991). It first calculates a Robust Color Morphological Gradient (RCMG) (Evans and Liu 2006) in order to reduce the data dimensionality. The watershed algorithm does not require parameters from the user.

  The other segmentation alternative is Quick Shift (QS) (Comaniciu and Meer 1999; Fulkerson and Soatto 2010). In this case two parameters must be tuned in order to achieve an adequate segmentation (Fulkerson and Soatto 2010): "sigma" and "tau". Increasing the first one smooths the underlying estimate of the density, producing less detailed segmentation maps. Increasing the second one increases the average size of each region.
- **Combination (stage IV). During this stage the classification and the segmentation results are combined producing a final classification map. The inputs to the algorithm are the classification map and the**

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

segmentation map. The output is an improved classification map. The algorithm available in the tool for this stage is based on a plurality vote step implemented by majority vote (MV) (Tarabalka, Benediktsson, and J. 2009; Waske et al. 2012). With this algorithm the classification map produced by the pixel-wise classifier is processed and every pixel inside a segmented region is assigned to the most repeated class within the region. MV does not require parameters.

- **Postprocessing (stage V). The last stage processes the classification map in order to improve the classification accuracy. The input to the stage is the 2-D classification map and the output is the improved 2-D classification map. A spatial regularization (Ayerdi, Marqués, and Graña 2014) is the algorithm implemented for this stage. It is an iterative process where the class of a pixel changes to the class of a percentage of its neighbors fixed by the user. Two parameters can be tuned by the user: "number of neighbors", the number of neighbors considered for each pixel, and "% for majority" that stands for the percentage of neighbors that must have the same label in order to perform an update of the pixel label.**

The classification schemes published in (Heras, Argüello, and Quesada-Barriuso 2014a; Quesada-Barriuso, Argüello, and Heras 2014a,b) can be implemented by combining the algorithms described in the previous paragraphs, i.e. using the described chain.

5. Examples of use

In this section different capabilities of the tool are illustrated. First, in Section 5.1 we show some examples of GUI use showing its registration and classification capabilities. Then, Section 5.2 is devoted to presenting new supervised classification schemes and to show classification results of the schemes for different datasets.

All the results obtained using the application were executed on a PC with a quad-core Intel Core2 Quad Q9450 at 2.66 GHz and 6 GB of RAM. The code was compiled using the gcc 4.4.3 version under Linux 2.6.32-64 with GTK 2.20.1-0, Cairo 1.8.10-2, Pango 1.28.0-0, Glade2 2.6.4-1 and Zlib 1.2.8. For the execution of the classification algorithms Lapack 3.4.2 and Blas 1.2-2 are also required.

The hyperspectral datasets **were** chosen from publicly available repositories for comparison purposes. The tests were run on three hyperspectral airborne datasets. First we have a 103-band ROSIS image of the urban area surrounding the University of Pavia, with a spatial size of 610×340 pixels (Pavia Univ.) and a spatial resolution of 1.3 m per pixel. The second one is 220-band AVIRIS image of 145×145 pixels taken over Northwest Indiana (Indian Pine). The spatial resolution of this image is of 20 m per pixel. The last one is a 204-band AVIRIS image of 512×217 pixels taken over the Salinas Valley, California (Salinas) with 3.7 m of spatial resolution. The numbers of classes considered for the classification were taken from the **reference data for** each image, with nine classes for the Pavia Univ. dataset, and sixteen for Indian Pine and Salinas.

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

5.1 *Examples of GUI use*

[PLEASE INSERT FIGURE 4 HERE]

The GUI is the visible part of the application and the one the user has to interact with in order to perform all the operations over the hyperspectral image. In this section the main features of the tool are illustrated for the particular case of the Univ. of Pavia dataset enhancing the usability which guided the design.

Fig. 4 provides a display of the application showing some of the classification features and results offered by the tool. From left to right, first a selected band of the image and **the reference data** assigning different colors to the pixels belonging to different classes are shown. Note that black areas mean that there is no information on the class for the pixels in those areas.

In the middle of the figure the menu for the configuration of the five stages of the classification chain is shown. In particular, for this example, the following algorithms were selected reproducing one of the classification schemes published in (López-Fandiño et al. 2015): there are no preprocessing nor postprocessing stages, the pixel-wise classification is performed by ELM, the spatial processing consists of a RCMG gradient plus a segmentation by watershed. Finally for the combination stage, the MV algorithm **was** selected.

We can also see the menu for selecting some parameters for the ELM algorithm. The classification map with different colors for the different classes is displayed on the right side of the figure. The accuracy results are similar to those previously published by the authors (See ELM+wat con(8) configuration in (López-Fandiño et al. 2015)).

[PLEASE INSERT FIGURE 5 HERE]

Fig. 5 illustrates an example of **reference data** creation. After asking the user for the properties of the sensor used in the image acquisition, a rgb image corresponding to the hyperespectral image is shown. If this information is unknown for the user the tool offers simply one band of the image. The user can then define classes by defining labels and colors as shown in the figure. For each class the user draws points that define polygons associated to the class. He can also add separate pixels or groups of pixels of a size defined by the user to the class or undo the addition. In order to inspect pairs of pixels, a window allows the comparison among them displaying all their components as depicted in the left side of the figure.

5.2 *Classification experiments*

Following the structure described in Section 4, classification results were obtained using the tool. The three datasets for the experiments are those described at the beginning of this section. Results for some configurations considering watershed as segmentation algorithm and SVM or ELM as classification algorithm were already published (Tarabalka, Chanussot, and Benediktsson 2010; López-Fandiño et al. 2015).

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

We have executed the published classification schemes using the tool in order to check that similar accuracy results to those published in the literature are obtained using it. Once this has been checked, we focus on evaluating the application by executing two new **supervised spatial-spectral classification schemes that do not perform preprocessing (no algorithm is selected for stage (I) in Fig. 3), use RCMG and QS for the spatial processing (QS is selected for stage (II)) and ELM/SVM for the pixel-wise classification (stage (III)). A MV algorithm performs the combination stage (stage (IV)) and a final postprocessing consisting of spatial regularization (stage (V)) is performed. We call these schemes ELM-QS and SVM-QS respectively.**

The different experiments are evaluated by the tool in terms of OA, AA, κ, QD and AD as explained in Section 3. The training samples are excluded from the testing process.

In the case of the QS segmentation algorithm (selected for spatial processing, stage II) two parameters must be provided by the user: σ and τ. The values for these parameters were selected trying to achieve a number of regions similar to the number achieved by the watershed algorithm. Finally the values for ("sigma","tau") are: (6,2), (1,2) and (4,2) for Pavia Univ., Indian Pine and Salinas images respectively.

For the experiments using SVM classification (pixel-wise classification, stage III) the "kernel" parameter selects Gaussian Radial Basis Function (RBF). The SVM was trained using values fixed by the user, the same values of "C" and "gamma" and number of training samples for the Pavia image as in(Tarabalka, Chanussot, and Benediktsson 2010): $(128, 0.125)$**. In the case of the Indian Pine and Salinas images, values were selected for "C" of 1024 and 256 and values for "gamma" of 0.5 and 0.125 respectively were determined by 5–fold cross validation by the tool. For these two images the "number of training samples per class" was selected as 10% of the total samples for all the classes as in (Tarabalka, Chanussot, and Benediktsson 2010) and in (Plaza, Plaza, and Barra 2009) for comparison purposes.**

In the experiments using ELM (selected as pixel-wise classification, stage III) the number of training samples ("training samples per class" parameter) is 200 per class, or half the total number of samples in the class if there are not enough samples. The number of neurons in the hidden layer employed ("hidden layer neurons" parameter) are 500 for Pavia Univ., 950 for Indian Pine, and 350 for Salinas in all the cases. All the parameters have been taken from (Heras, Argüello, and Quesada-Barriuso 2014b; López-Fandiño et al. 2015) in order to perform a comparison with the published results.

For the combination stage (stage IV) the majority vote algorithm is selected, that does nor require parameters from the user. For the postprocessing stage (stage V) the spatial regularization algorithm is used. The parameter "neighbors" takes value 8 (8 neighbors per pixel) and "majority" takes a vulue of 50 (50% of the neighbors must have the same label in order to change the label of a pixel).

Table 1 shows accuracy results obtained by using the tool for ELM-QS and SVM-QS comparing to results for similar schemes in the literature. First, results for the raw application of the SVM classification algorithm are shown (Tarabalka, Chanussot, and Benediktsson 2010; Plaza, Plaza, and Barra 2009). This is a classification scheme where no spatial processing is performed. Then next column includes results for the scheme that extracts spatial information from the image through watershed,

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

clasifies the original hyperspectral dataset by using SVM and, finally, combines the information of the classification and the segmentation maps by MV (SVM-wat in the table) (Tarabalka, Chanussot, and Benediktsson 2010). The next columns in the table show results for the same configurations but using ELM for pixel-wise classification instead of SVM (López-Fandiño et al. 2015). As we have mentioned before, results for the configurations using QS as segmentation algorithm were not published previously, so the accuracy results are obtained by the tool. For the other configurations the results in the table are extracted from the bibliography. **There are no execution times for these schemes in the literature, so the execution times included in Table 1 were all obtained using the tool.**

Considering the best values for each image, that are highlighted in bold in the table, the first observation is that for two of the images, for Pavia Univ. and Salinas, the accuracy results obtained by the schemes using QS are slightly better than the results obtained by ELM-wat, the second best scheme for these images. In the case of the Indian Pine image the best results are obtained by SVM-wat and the second best by ELM-wat. For this image the accuracy results are lower than for the other images because it is smaller than the other two in spatial size, so the training is more difficult.**The lowest execution times are those obtained for the schemes including ELM, and the highest ones are those for the schemes including segmentation by QS. A comparison based on the datasets gives the lowest execution times for the smallest dataset: Indian Pine.**

Figures 6, 7 and 8 include first on the left the **reference data** map, then the classification map for the raw classification by SVM without any pre- or postprocessing and without spatial processing (SVM column in Table 1). Finally, on the right, the map produced by the QS-based scheme that gives the best results in the table is shown. The classification maps shown are in all cases obtained by using the tool.

[PLEASE INSERT TABLE 1 HERE]

[PLEASE INSERT FIGURES 6, 7, AND 8 HERE]

Each color in the images corresponds to a different class and it is important to remember that the black areas indicate that there is no reference information on the classification label for those pixels. So, they are not taken into account when the accuracies of the classification schemes are calculated. It can be observed that the maps on the right, corresponding to QS-based schemes present in general more uniform areas with a lower number of isolated pixels. For example, observe for Pavia Univ. in Figure 6 the regions in the bottom of the images or the upper left area in Figure 8.

6. Summary

In this paper a licence-free desktop application for the analysis and **supervised** spectral-spatial classification of hyperspectral remote sensing images is presented. The user can analyze details of the images inspecting them band by band, performing zoom, analyzing the spectrum for each pixel or even rotating and scaling

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

the images. The **reference data** that are required for supervised classification can also be graphically generated. **Statistics of classification results on regions defined by the user are calculated.** It is also possible to register pairs of hyperspectral images.

Different **supervised** classification schemes are available. These are built **using a chain** made up of five different stages that are configurable by selecting among different algorithms through a simple and intuitive interface. The stages are: preprocessing, spatial proccesing, pixel-wise classification, combination, and postprocessing. This modular structure makes the application suitable for comparing different classification schemes visually, in terms of accuracy **or in terms of execution time**. New classification pipelines can be introduced by the user in a simple way, **so the application is designed to be a platform where the user can try new algorithms.** The application runs on the Linux operating system and was developed in C language. For validation, results using the tool are obtained for **published classification schemes and for two new spatial-spectral ones** based on the QS algorithm for segmentation (spatial processing) and on SVM or ELM for pixel-wise classification obtaining accuracy results that achieve OA values of 95.25% for the Pavia Univ. image.

Supplemental data

The underlying research materials for this article can be accessed at http://wiki.citius.usc.es/software/hyperview

Acknowledgements

This work was supported in part by the Ministry of Science and Innovation, Government of Spain, cofounded by the FEDER funds of European Union [grant number TIN2013-41129-P] and by Xunta de Galicia, Programme for Consolidation of Competitive Research Groups [grant number 2014/008].

References

Abdi, Hervé, and Lynne J. Williams. 2010. "Principal component analysis." *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (4): 433–459.

Anguix, A, and G Carrión. 2005. "gvSIG: Open Source Solutions in spatial technologies." *GISPLANET, Estoril, Portugal* .

Ayerdi, Borja, Ion Marqués, and Manuel Graña. 2014. "Spatially regularized semisupervised Ensembles of Extreme Learning Machines for hyperspectral image segmentation." *Neurocomputing* http://dx.doi.org/10.1016/j.neucom.2014.01.068i.

Benediktsson, J. A., M. Pesaresi, and K. Amason. 2003. "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations." *Geoscience and Remote Sensing, IEEE Transactions on* 41 (9): 1940–1949.

Bernabé, S., A. Plaza, P. R. Marpu, and J.A. Benediktsson. 2013. "A new parallel tool for classification of remotely sensed imagery." *Computers and Geosciences* 6 (4): 1934–1948. http://dx.doi.org/10.1016/JSTARS.2012.2230247.

Camps-Valls, G., and L. Bruzzone. 2005. "Kernel-based methods for hyperspectral image classification." *Geoscience and Remote Sensing, IEEE Transactions on* 43 (6): 1351–1362.

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

Chang, Chih-Chung, and Chih-Jen Lin. 2011. "LIBSVM: a library for support vector machines." *ACM Transactions on Intelligent Systems and Technology (TIST)* 2 (3): 27.

Chen, Qin-sheng, Michel Defrise, and Frank Deconinck. 1994. "Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16 (12): 1156–1168.

Comaniciu, D., and P. Meer. 1999. "Mean shift analysis and applications." In *Computer Vision, 1999 IEEE76th International Conference on,* Vol. 21197–1203.

Dufaux, F., and J. Konrad. 2000. "Efficient, Robust, and Fast Global Motion Estimation for Video Coding." *IEEE Trans. Image Processing* 9 (3): 497–501.

Evans, A., and X. Liu. 2006. "A morphological gradient approach to color edge detection." *Image Processing, IEEE Trans. on* 15 (6): 1454–1463. http://dx.doi.org/10.1109/TIP.2005.864164.

Fauvel, Mathieu, Yuliya Tarabalka, Jon Atli Benediktsson, Jocelyn Chanussot, and James C Tilton. 2013. "Advances in spectral-spatial classification of hyperspectral images." *Proceedings of the IEEE* 101 (3): 652–675.

Fulkerson, Brian, and Stefano Soatto. 2010. "Really quick shift: Image segmentation on a GPU." In *Trends and Topics in Computer Vision,* 350–358. Springer.

Heras, Dora B., Francisco Argüello, and Pablo Quesada-Barriuso. 2014a. "Exploring ELM-based spatial–spectral classification of hyperspectral images." *International Journal of Remote Sensing* 35 (2): 401–423.

Heras, Dora B., Francisco Argüello, and Pablo Quesada-Barriuso. 2014b. "Exploring ELM-based spatial–spectral classification of hyperspectral images." *International Journal of Remote Sensing* 35 (2): 401–423.

Huang, Guang-Bin, Dian Hui Wang, and Yuan Lan. 2011. "Extreme learning machines: a survey." *International Journal of Machine Learning and Cybernetics* 2 (2): 107–122.

Janz, Andreas, Sebastian Van Der Linden, Björn Waske, and Patrick Hostert. 2007. "ImageS-VM: a user-oriented tool for advanced classification of hyperspectral data using support vector machines." *Proceedings of the EARSeL SIG Imaging Spectroscopy, Bruges, Belgium* .

Landgrebe, David A. 2005. *Signal theory methods in multispectral remote sensing.* Vol. 29. John Wiley & Sons.

Lee, C. A., S. D. Gasster, A. Plaza, C. I. Chang, and B. Huang. 2011. "Recent developments in high performance computing for remote sensing: A review." *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* 4 (3): 508–527.

López-Fandiño, J., P. Quesada-Barriuso, D. B. Heras, and F. Argüello. 2015. "Efficient ELM Based Techniques for the Classification of Hyperspectral Remote Sensing Images on Commodity GPUs." *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* http://dx.doi.org/10.1109/JSTARS.2014.2384133.

Melgani, Farid, and Lorenzo Bruzzone. 2004. "Classification of hyperspectral remote sensing images with support vector machines." *Geoscience and Remote Sensing, IEEE Transactions on* 42 (8): 1778–1790.

Neteler, M., M. H. Bowman, M. Landa, and M. Metz. 2012. "GRASS GIS: A multi-purpose open source GIS." *Environmental Modelling and Software* 31: 124–130. http://dx.doi.org/10.1016/j.envsoft.2011.11.014.

Pan, Wei, Kaihuai Qin, and Yao Chen. 2009. "An adaptable-multilayer fractional Fourier transform approach for image registration." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31 (3): 400–414.

Plaza, J., A. J. Plaza, and C. Barra. 2009. "Multi-channel morphological profiles for classification of hyperspectral images using support vector machines." *Sensors* 9 (1): 196–218.

Pontius, R. G. Jr, and M. Millones. 2011. "Death to kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment.." *International Journal of Remote Sensing* 32 (15): 4407–4429.

Quesada-Barriuso, Pablo, Francisco Argüello, and Dora B. Heras. 2014a. "Computing Efficiently Spectral-Spatial Classification of Hyperspectral Images on Commodity GPUs." In *Recent Advances in Knowledge-based Paradigms and Applications,* 19–42. Springer.

Quesada-Barriuso, Pablo, F Argüello, and Dora B. Heras. 2014b. "Spectral–spatial clas-

sification of hyperspectral images using wavelets and extended morphological profiles." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)* 7 (4): 1177–1185.

Quesada-Barriuso, Pablo, Francisco Argüello, and Dora B. Heras. 2015. "Wavelet-based classification of Hyperspectral Images Using Extended Morphological Profiles on Graphics Processing Units." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8 (6): 2962–2970.

Quesada-Barriuso, Pablo, Dora B. Heras, and Francisco Argüello. 2013. "Efficient 2D and 3D watershed on graphics processing unit: block-asynchronous approaches based on cellular automata." *Computers & Electrical Engineering* 39 (8): 2638–2655.

Richards, John A., and Xiuping Jia. 1999. *Remote sensing digital image analysis*. Vol. 3. Springer.

Steiniger, Stefan, and Erwan Bocher. 2009. "Free and open source GIS software for building a spatial data infrastructure." *International Journal of Geographical Information Science* 23 (10): 1345–1370.

Tarabalka, Y., Benediktsson, and J. J., Chanussot. 2009. "Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques." *Geosci. and Remote Sensing, IEEE Trans. on* 47 (8): 2973–2987. http://dx.doi.org/10.1109/TGRS.2009.2016214.

Tarabalka, Yuliya, Jón Atli Benediktsson, and Jocelyn Chanussot. 2009. "Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques." *Geoscience and Remote Sensing, IEEE Transactions on* 47 (8): 2973–2987.

Tarabalka, Y., J. Chanussot, and J. Benediktsson. 2010. "Segmentation and classification of hyperspectral images using watershed transformation." *Pattern Recognition* 43 (7): 2367–2379. http://dx.doi.org/10.1016/j.patcog.2010.01.016.

University, Purdue. 2014. "MultiSpec software and documentation." https://engineering.purdue.edu/ biehl/MultiSpec/.

van der Meer, F.D., H.M. van der Werff, F.J. van Ruitenbeek, C.A. Hecker, W.H. Bakker, and M.F. Noomen. 2012. "Multi- and hyperspectral geologic remote sensing: A review." *International Journal of Applied Earth Observation and Geoinformation* 14 (11): 112–128. http://dx.doi.org/10.1016/j.jag.2011.08.002.

Vicent, L., and P. Soille. 1991. "Watersheds in digital spaces: an efficient algorithm based on immersion simulations." *Image Processing, IEEE Trans. on* 13: 583–598. http://dx.doi.org/10.1109/34.87344.

Viera, Anthony J, Joanne M. Garrett, et al. 2005. "Understanding interobserver agreement: the kappa statistic." *Fam Med* 37 (5): 360–363.

Waske, B., S. van der Linden, C. Oldenburg, B. Jakimow, A. Rabe, and P. Hostert. 2012. "ImageRF A user-oriented implementation for remote sensing image analysis with Random Forests." *Environmental Modelling and Software* 35: 192–193. http://dx.doi.org/10.1016/j.envsoft.2012.01.014.

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

Table 1. Classification accuracies in percentage **and CPU execution times in seconds**. The best accuracy values are indicated in bold.

	SVM[a]	SVM-wat[a]	SVM-QS	ELM[b]	ELM-wat[b]	ELM-QS
Pavia Univ.						
OA	81.01	85.42	**95.85**	86.75	95.09	94.91
AA	88.25	91.31	**94.95**	89.55	95.14	94.71
κ	75.86	81.30	**94.46**	82.61	93.44	93.28
QD	–	–	1.47	–	–	2.57
AD	–	–	2.69	–	–	2.51
t	332.92	342.05	371.30	31.74	41.40	71.00
Indian Pine						
OA	78.76	**92.48**	89.42	80.72	91.41	90.92
AA	69.66	77.26	78.19	85.48	**93.91**	82.68
κ	75.75	91.39	87.89	77.70	**89.98**	89.73
QD	–	–	4.45	–	–	5.33
AD	–	–	6.11	–	–	3.75
t	30.36	32.49	32.59	27.46	29.57	32.01
Salinas						
OA	84.17	–	91.88	91.55	93.31	**93.96**
AA	–	–	95.56	95.97	96.52	**96.85**
κ	–	–	90.92	90.55	92.51	**93.26**
QD	–	–		–	–	1.60
AD	–	–		–	–	4.44
t	384.72	–	398.45	22.22	33.29	38.50

[a]Accuracy data for SVM and SVM-wat are taken from (Tarabalka, Chanussot, and Benediktsson 2010) except SVM results for Salinas image that are taken from (Plaza, Plaza, and Barra 2009).

[b]Accuracy data for ELM and ELM-wat are taken from (López-Fandiño et al. 2015).

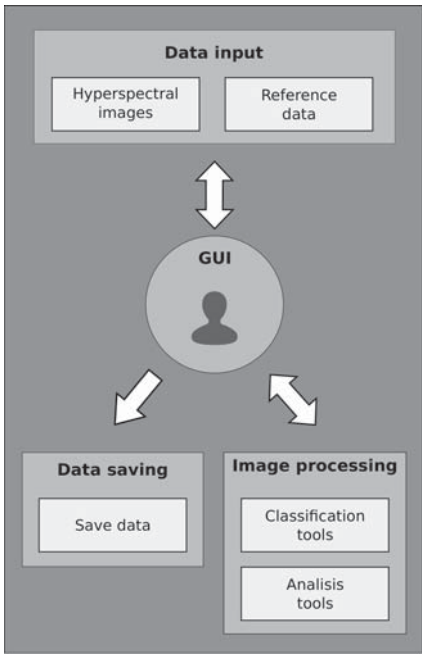

Figure 1. Methodology applied for the system development.

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

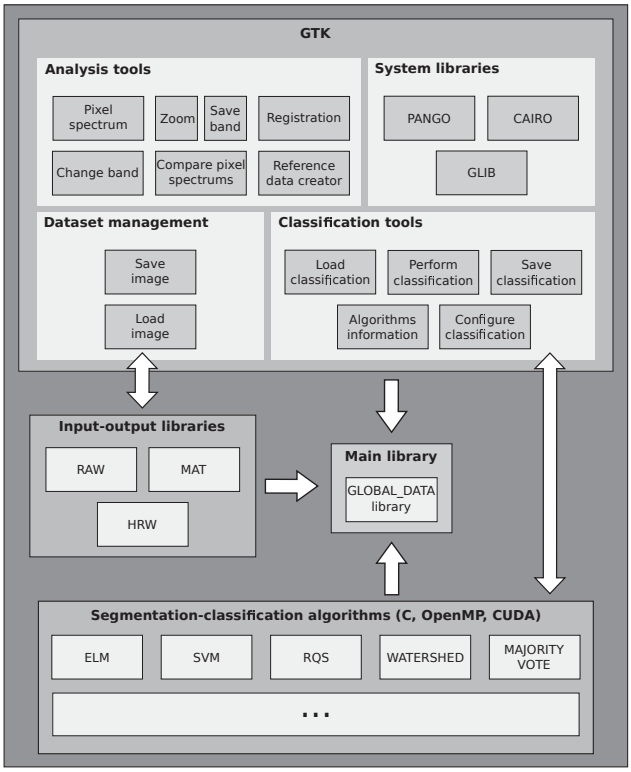

Figure 2. Architecture of the proposed system.

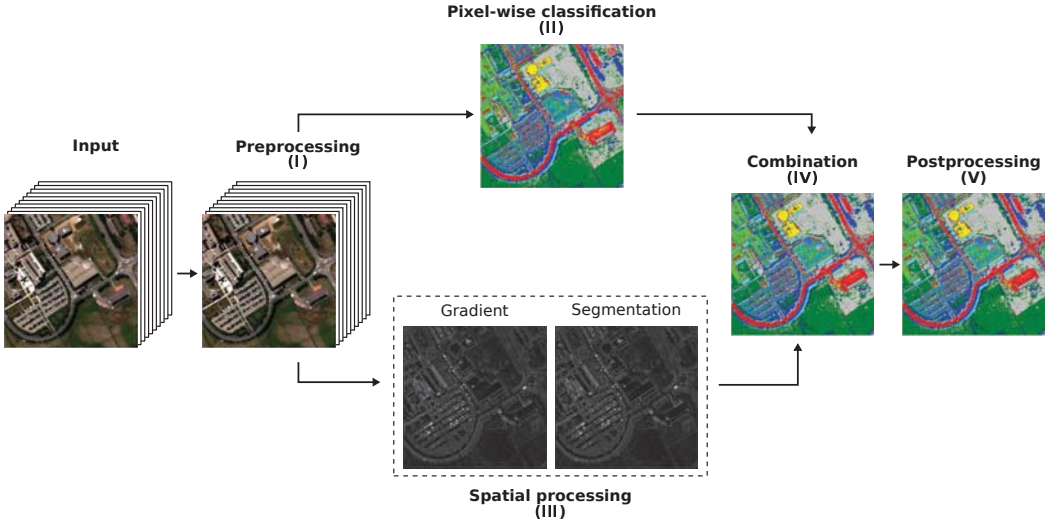

Figure 3. Five-stage spectral-spatial classification chain implemented in the application.

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

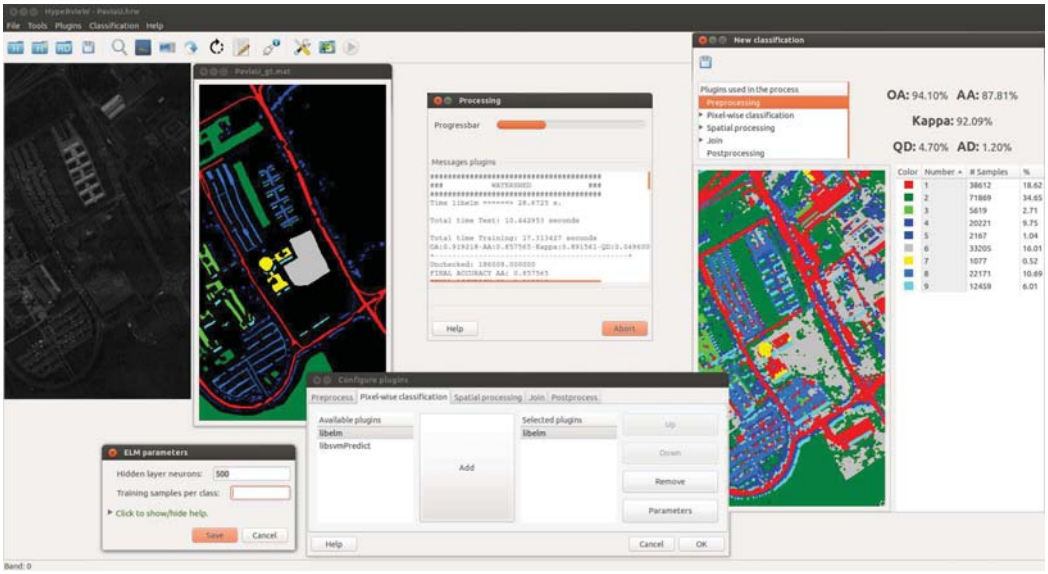

Figure 4. Example of spectral-spatial classification performed by the tool over the Pavia Univ. dataset.

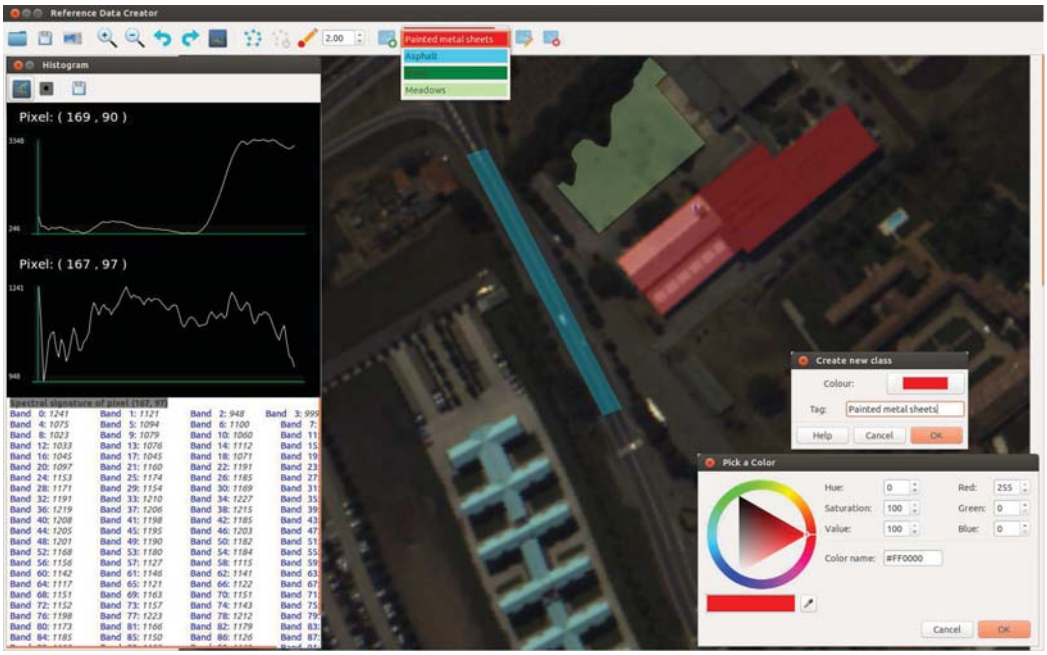

Figure 5. Example of **reference data** creation over the Pavia Univ. dataset.

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Remote Sensing on 24/10/2016, available online: http://www.tandfonline.com/10.1080/01431161.2016.1244363.

Figure 6. From left to right **reference data**, and classification maps for SVM and SVM-QS for the Pavia Univ. image.

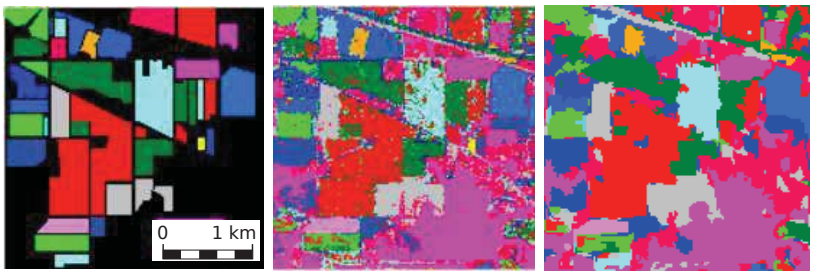

Figure 7. From left to right **reference data**, and classification maps for SVM and SVM-QS for the Indian Pine image.

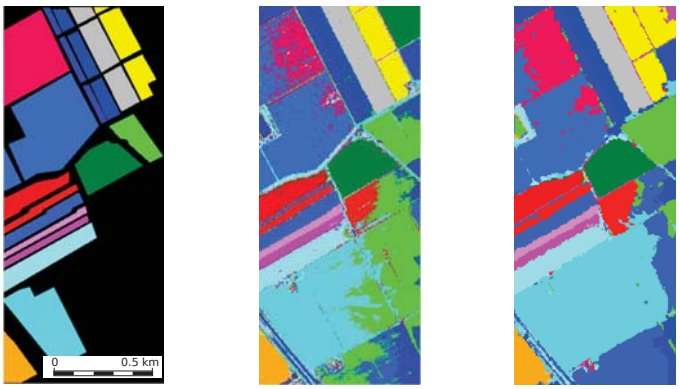

Figure 8. From left to right reference data, and classification maps for SVM and ELM-QS for the Salinas image.