Interactive Pansharpening and Active Classification in Remote Sensing

P. Ruiz, J. Mateos, G. Camps-Valls, R. Molina and A. K. Katsaggelos

Abstract This paper presents two multimodal prototypes for remote sensing image classification where user interaction is an important part of the system. The first one applies pansharpening techniques to fuse information from different satellite sensors to obtain a high resolution (HR) multispectral image. Once the HR image has been classified the user can interact with the system to select a class of interest. The pansharpening parameters are then modified to increase the system accuracy for the selected class without deteriorating the performance of the classifier on the other classes. The second prototype utilizes Bayesian modeling and inference to implement active learning and parameter estimation in binary kernel-based multispectral classification problems with possibly infinite dimensional feature spaces. In the prototype we developed three different strategies for selecting the more informative pixel to be included in the training set. In the experimental section, the prototypes are described and applied to two real multispectral image classification problems.

1 Introduction

Remote sensing images are of great interest in numerous applications. Map drawing, delimitation of parcels, studies on hydrology, forest or agriculture are just a few

P. Ruiz, J. Mateos and R. Molina
Dpt. Computer Science and A.I., University of Granada,
e-mail: {mataran, jmd, rms}@decsai.ugr.es

G. Camps-Valls
Image Processing Laboratory (IPL), Universitat de València
e-mail: gustavo.camps@uv.es

A. K. Katsaggelos
Dpt. of Electrical Engineering and Computer Science, Northwestern University, Evanston,IL
e-mail: aggk@eecs.northwestern.edu

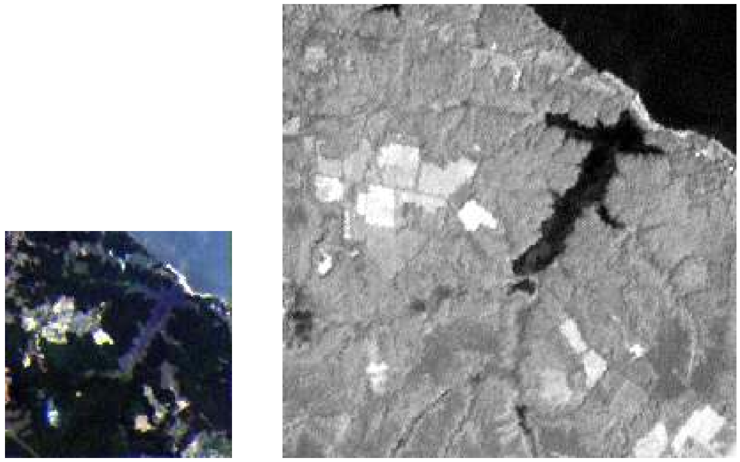

Fig. 1 Region of interest of the observed multispectral image (left) and the corresponding region of interest of the observed panchromatic image (right).

examples where these images are used [10, 11, 7]. Many of these applications involve the classification of pixels in an image into a number of classes. In supervised classification, the user provides the label of a set of samples to train the classifiers. Usually, the bigger the training set, the better the classification results but more expensive (in time or money) the construction of such a set is.

In this paper we present two prototypes that use multimodal data and user interactivity to obtain more accurate classification results or obtain them at a lower cost. Both prototypes deal with the same multimodal remote sensing image classification problem although they consider user interaction from two different but complementary points of view: user feedback and system adaptation.

On one hand, the first prototype deals with the problem of obtaining more accurate classification results on remote sensing images for specific classes. Due to physical and technological constrains, satellites usually have multimodal sensors that capture two types of images. One sensor captures a multispectral (MS) image composed of several spectral bands with low spatial resolution (LR). For instance, Landsat ETM+ captures 6 spectral bands in the visible and infrared spectrum with each pixel covering an area of 30×30 meters. The other sensor captures a high spatial resolution (HR) image with a low spectral resolution, named panchromatic (PAN) image, that in the case of Landsat has a resolution of 15×15 meters per pixel. Figure 1 shows an example of an Landsat LR MS and the HR PAN images. The LR MS and PAN images are fused using pansharpening techniques to obtain an image with the spectral resolution of the MS image and the spatial resolution of the PAN image. A complete review of techniques to carry out the pansharpening procedure can be found in [2].

Bruzzone *et al.* [5] showed that the use of pansharpening methods that do not introduce significant spectral distortion helps the classifier to obtain higher accuracy, especially for pixels at the borders of objects. While pansharpening has traditionally only been used as a preprocessing step, that is, before using supervised classification, in the first prototype we address the problem of interactively modifying the pansharpening parameters to improve the figures of merit of the supervised classification of a class of interest, selected by the user. Hence, in this case, the user interaction consists of selecting a class of interest and then the system adapts the pansharpening to obtain better classification results for the selected class.

On the other hand, the second prototype implements *active learning* concepts exploiting the Bayesian modeling and inference paradigm to tackle the problem of binary kernel-based multispectral image classification.

Active learning aims at building efficient training sets by *iteratively* improving the model performance through sampling. When applied to remote sensing image classification active learning methods start with a few pixels from each class whose labels are known and, then, iteratively select, using a given criterion, pixels from the rest of the image. The classifier interactively asks the oracle, i.e., the user, for the class of the selected pixels. The feedback obtained at each step of the interaction process is converted into new, fresh training information, useful for improving the classifier. In the presented prototype we developed three different strategies for selecting the most informative pixel to be included in the training set. Additionally, if the user inputs a set of test samples, that is, a set of pixels of known class that can be used to evaluate the quality of the classification, the system can provide some numerical performance information. The objective is to obtain the best classification accuracy with the minimum number of queries to the oracle. A survey of active learning algorithms for supervised remote sensing image classification can be found in [20].

The remainder of the paper is outlined as follows. Section 2 describes the theory behind the interactive pansharpening based classification prototype. The Bayesian active learning techniques used in the second prototype are described in section 3. Section 4 presents the prototypes description and experimental results and, finally, section 5 concludes the paper.

2 Interactive Pansharpening based Classification

Multispectral images allow for an accurate recognition of several land cover classes but, due to their low resolution, information on the objects shape and texture may be lost. In contrast, panchromatic images allow for a better recognition of the objects in the image and their textures but provide no information about their spectral properties. Pansharpening is a technique that jointly processes multispectral and panchromatic images in order to obtain a new multispectral image that, ideally, exhibits the spectral characteristics of the observed multispectral image and the resolution of the panchromatic image.

In this prototype, we propose the use of pansharpening techniques to increase the performance of a classification system. The pansharpening method provides a high resolution multispectral image that is used, along with the observed panchromatic image, as input of the interactive classification algorithm. The pansharpening method depends on a set of parameters that are automatically estimated from the data and which determine the quality of the obtained pansharpened image.

Even when using a sophisticated pansharpening technique, the classification results may not fulfill the expectations of the user. Hence, we propose to adapt the pansharpening method to improve the classification of a specific class of interest, chosen by the user, by adjusting its parameters to perform this specific class. This represents an application-specific pansharpening approach, where the application of interest is binary classification.

2.1 Pansharpening

The pansharpening methods included in the prototype are formulated within the Bayesian paradigm because it provides good reconstructions and allows for the adaptation of the pansharpening to the needs of the user. More precisely we have implemented the following two pansharpening methods in the prototype:

- **SAR (Simultaneously Autoregressive) based pansharpening:** The method proposed in [13] assumes that each band of a MS image is a degraded (blurred and decimated) version of the original HR MS image, and the PAN image is a linear combination of HR MS image bands. It uses a classical Simultaneous Autoregressive (SAR) model [14] to impose smoothness on the HR MS image. The method depends on a series of parameters that are modeled as Gamma distributions since they allow for the incorporation of information from the data as well as prior information about the value of the parameters. Both the HR MS image and the associated parameters are automatically estimated by the method without user intervention.
- **CONTOURLET based pansharpening:** It is a novel method, proposed in [1], that assumes that the observed LR MS image has the same spectral properties than the HR MS image but with a lower level of details. The PAN image, on the other hand, contains the details of the HR MS image but lacks the spectral information. The method uses the non-subsampled contourlet transform (NSCT) to decompose the images into residual and detail bands and models the relations between the details of the HR MS image and those of the observed LR MS and PAN images in the contourlet domain. Since the NSCT detail bands are composed of relatively smooth regions separated by strong edges, Total Variation (TV) [15, 21] is used as prior model. The needed parameters are automatically estimated at each level of decomposition and direction for each band, providing a sound way to control the noise and preventing color bleeding.

It is worth noting that in both cases the estimation of the parameters and the HR MS image is performed in an iterative manner where, at each iteration, a new set of parameters is estimated from all the pixels of the current estimation of the HR MS image, and those parameters are used to obtain a new estimation of the pansharpened image.

2.2 Classification

Once the pansharpened image has been obtained, the classification procedure is carried out using the pansharpened and PAN images as input data. The user has to provide the class of a number of pixels to form the training set. Additionally, a validation/test set is also provided by the user to evaluate the performance of the classification process. In the presented prototype we have implemented two different classification techniques: linear discriminant analysis (LDA) and support vector machines (SVM), which have been largely exploited in remote sensing applications [6, 3].

LDA is an effective subspace technique that optimizes Fisher's score [9]. Subspace methods are algorithms focused on finding projections of an original hyperdimensional space to a lower dimensional space where the classes have maximum separation. LDA is related to Fisher's linear discriminant and, roughly speaking, both aim at finding a linear combination of features that characterize or separate two or more classes.

SVM is one of the most successful examples of kernel methods, being a linear classifier that implements maximum margin separation between classes in a high dimensional Hilbert space $\mathscr{H}$. Kernel methods embed the data observed in the input space $\mathscr{X}$ into a higher dimensional space, the feature space $\mathscr{H}$, where the data are more likely to be linearly separable. Therefore, it is possible to build an efficient linear classifier in $\mathscr{H}$, that translates into a nonlinear classifier in the input space. Kernel methods compute the similarity between training samples $\{\mathbf{x}_i\}_{i=1}^n$ using inner products between mapped samples instead of computing the dot product in the higher dimensional space explicitly. The so-called kernel matrix $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle$ contains all necessary information to perform many classical linear algorithms in the feature space, which are non-linear in the input space [19]. The radial basis function (RBF), $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_K^2\right), \sigma_K \in \mathbf{R}^+$ is selected in this work. To implement SVM for multiclass problems we used the one-versus-all strategy given the particular characteristics of the proposed scheme.

Utilizing one of the described classification methods, the user obtains a classification map. We used the precision and recall scores to assess the model's accuracy:

$$recall = \frac{TP}{TP+FN}; \qquad precision = \frac{TP}{TP+FP} \tag{1}$$

where TP is the number of pixels in the class correctly classified, FN is the number of pixels in the class incorrectly classified and FP is the number of pixels not belonging to the class incorrectly classified.

2.3 Interactive Pansharpening based Classification

Using one of the classifiers, chosen by the user, the classification performed on the HR MS image and panchromatic images usually obtains higher accuracy than the one performed using the observed multispectral and panchromatic images, especially for pixels at the borders of objects, as showed by Bruzzone *et al.* in [5]. However, it is possible that the outcome does not fulfill the user's expectations. This may be due to a suboptimal performance in a class of interest. Then, by examining the classification results, both visually and numerically, the user can select a class of interest to be improved.

Following the method in [17], the parameters needed by the pansharpening method are estimated again. Using the already estimated pansharpened image, the parameters for the new reconstruction are estimated utilizing only the pixels belonging to the class of interest in this image. Note that no iteration is required for this case. The new estimated parameters result in a new estimation of the HR MS image whose spectral and spacial characteristics are more accurate for the pixels in the class of interest and, hence, with a boosted classification performance for the elements of the class. Note however that this may imply that for other classes the user is not interested in, the classifier may perform slightly worse over the new pansharpened image [17].

3 Active Learning

One alternative way of handling user interaction is related to the emerging field of *active learning*. Let us assume that we have access to a set of samples for which the corresponding class, although not already known, can be provided by an oracle, i.e., the user. The key is to decide which elements to acquire from the set of possible samples in order to build an optimal compact classifier. Active learning aims at building efficient training sets by *iteratively* improving the model performance through sampling.

The prototype implements the Bayesian active learning procedure described in [16] where the Bayesian modeling and inference paradigm are applied to a binary kernel-based classifier tackling both active learning and parameter estimation for infinite dimensional feature spaces.

3.1 Bayesian Modeling and Inference

The Bayesian active learning procedure implemented in the prototype aims at solving the general two-classes supervised classification problem [4] that uses the classification function

$$y(\mathbf{x}) = \boldsymbol{\phi}^\top(\mathbf{x})\mathbf{w} + b + \varepsilon, \tag{2}$$

where the mapping $\boldsymbol{\phi} : \mathscr{X} \to \mathscr{H}$ embeds the observed $\mathbf{x} \in \mathscr{X}$ into a higher dimensional (possibly infinite) feature space $\mathscr{H}$, $\mathbf{w}$ is the vectors of parameters to be estimated, b represents the bias in the classification function, and ε are independent realizations of Gaussian distributions $\mathscr{N}(0,\sigma^2)$.

Hence, we can model the classification output $y(\mathbf{x}_i)$ associated with the feature samples $\boldsymbol{\phi}(\mathbf{x}_i), i = 1,\ldots,M$, with M the number of samples, as

$$\mathrm{p}(\mathbf{y}|\mathbf{w},\sigma^2) = \prod_{i=1}^{M} \mathscr{N}(y(\mathbf{x}_i)|\boldsymbol{\phi}^\top(\mathbf{x}_i)\mathbf{w} + b, \sigma^2), \tag{3}$$

where $\mathbf{y} = (y(\mathbf{x}_1), y(\mathbf{x}_2),\ldots,y(\mathbf{x}_M))^\top$. Since $\mathbf{x}_i$, $i = 1,\ldots,M$, will always appear as conditioning variable, for the sake of simplicity, we have removed the dependency on $\mathbf{x}_1,\ldots,\mathbf{x}_M$ in the left-hand side of the equation. We note that, for infinite dimensional feature vectors $\phi(\mathbf{x}_i)$, $\mathbf{w}$ is infinite dimensional. Following [4], we assume as prior distribution that each component of $\mathbf{w}$ independently follows a Gaussian distribution $\mathscr{N}(0,\gamma^2)$.

To perform the inference tasks, that is, parameter estimation, prediction and active learning, we will mainly use the marginal distribution of the observations. The marginal distribution of $\mathbf{y}$ can be obtained by integrating out the vector of adaptive parameters $\mathbf{w}$. It can easily be shown, see for instance [4], that

$$\mathrm{p}(\mathbf{y}|\gamma^2,\sigma^2) = \mathscr{N}(\mathbf{y}|b\mathbf{1},\mathbf{C}), \tag{4}$$

with

$$\mathbf{C} = \gamma^2 \boldsymbol{\Phi}\boldsymbol{\Phi}^\top + \sigma^2\mathbf{I}, \tag{5}$$

where $\boldsymbol{\Phi}$ is the design matrix whose i-th row is $\boldsymbol{\phi}^\top(\mathbf{x}_i)$. The estimation of the parameters b, γ^2 and σ^2 is carried out by using the evidence Bayesian approach [12] which amounts to maximizing the marginal distribution in Eq. (4).

It is important to note that we do not need to know the form of $\boldsymbol{\Phi}$ explicitly to calculate this distribution. We only need to know the Gram matrix $\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top$, which is an $M \times M$ symmetric matrix with elements $\mathbf{K}_{nm} = k(\mathbf{x}_n,\mathbf{x}_m) = \phi^\top(\mathbf{x}_n)\phi(\mathbf{x}_m)$, which has to be a positive semidefinite matrix (see [18]).

3.2 Classification

Once the system has been trained, we want to assign a class to a new value of $\mathbf{x}$, denoted by $\mathbf{x}_*$. The conditional distribution $\mathrm{p}(y(\mathbf{x}_*)|\mathbf{y})$ is a Gaussian distribution with mean $m(\mathbf{x}_*)$ and variance $v(\mathbf{x}_*)$ given by:

$$m(\mathbf{x}_*) = b + \gamma^2 \boldsymbol{\phi}^\top(\mathbf{x}_*)\boldsymbol{\Phi}^\top\mathbf{C}^{-1}(\mathbf{y} - \mathbf{1}b) \tag{6}$$

$$v(\mathbf{x}_*) = \sigma^2 + \gamma^2 \boldsymbol{\phi}^\top(\mathbf{x}_*)\boldsymbol{\phi}(\mathbf{x}_*) - \gamma^4 \boldsymbol{\phi}^\top(\mathbf{x}_*)\boldsymbol{\Phi}^\top\mathbf{C}^{-1}\boldsymbol{\Phi}\boldsymbol{\phi}(\mathbf{x}_*). \tag{7}$$

So, we classify $\mathbf{x}_*$ using $m(\mathbf{x}_*)$ defined in Eq. (6) and write

$$\mathbf{x}_* \text{ is assigned to } \begin{cases} \mathscr{C}_1 & \text{if } m(\mathbf{x}_*) > 0.5 \\ \mathscr{C}_0 & \text{if } m(\mathbf{x}_*) < 0.5 \end{cases}. \tag{8}$$

3.3 Active Learning Approaches

As already explained, active learning starts with a small set of pixels whose class is already known. From these observations, the marginal distribution of $\mathbf{y}$, and all the parameters are estimated using the procedure described in the previous sections.

In order to improve the performance of the classifier we want to select, from the pixels of the image, the most informative sample, $\mathbf{x}_+$, and the user will be asked about its label, $y(\mathbf{x}_+)$. Then the classifier is updated using the new information provided by the user. The process continues until a given number of samples has been included in the training set. To select which of the available samples is added to the training set, the prototype implements three methods described in [16], which are reviewed in the following sections.

3.3.1 Maximum differential of entropies

For a sample $\mathbf{x}$ not already present in the training set, the distribution $\mathrm{p}(y(\mathbf{x}_*)|\mathbf{y})$ can be calculated using Eqs. (6) and (7), and consequently we can select the new training sample as the one maximizing the variance of the prediction, that is,

$$\mathbf{x}_+ = \arg\max_{\mathbf{x}} v(\mathbf{x}). \tag{9}$$

This criterion amounts to select the sample the classifier is less certain about the class it belongs to.

3.3.2 Minimum distance to decision boundary

In our classification problem the decision boundary corresponds to the set

$$\boldsymbol{\Pi} = \{\mathbf{x} \in \mathscr{X} : m(\mathbf{x}_*) - 0.5 = 0\}. \tag{10}$$

We can then select the next sample to be included in the training set as the one with minimum distance to the decision boundary by using

$$\mathbf{x}_+ = \arg\min_{\mathbf{x}} \mathrm{d}^2(\mathbf{x}, \boldsymbol{\Pi}) = \arg\min_{\mathbf{x}} (m(\mathbf{x}) - 0.5)^2. \tag{11}$$

Note that this method provides a Bayesian formulation of the SVM margin sampling heuristic (see [20]).

3.3.3 Minimum Normalized Distance

The two active learning methods described above take into consideration only partial aspects of the conditional distribution $\mathrm{p}(y(\mathbf{x}_*)|\mathbf{y})$. While maximum differential of entropies utilizes the variance of this distribution, it does not use the distance to the decision boundary. On the other hand, the minimum distance to the decision boundary criterion is based on the mean of this conditional distribution and does not take into account the uncertainty of the distribution. It is obviously very easy to imagine scenarios where these two criteria will not select the best sample, either because it is too far from the decision boundary and, hence, having large variance does not represent a problem, or because, although the sample is the closest to the decision boundary, its uncertainty is very small and consequently it may not be the best sample to be included in the training set.

We can then use the following active learning procedure which combines precision and proximity to the decision boundary

$$\mathbf{x}_+ = \arg\min_{\mathbf{x}} \mathrm{E}\left[\frac{(y(\mathbf{x})-0.5)^2}{v(\mathbf{x})}\right], \tag{12}$$

where the expected value is calculated utilizing the conditional distribution $\mathrm{p}(y(\mathbf{x})|\mathbf{y})$. Notice that since

$$\mathrm{E}\left[\frac{(y(\mathbf{x})-0.5)^2}{v(\mathbf{x})}\right] = 1 + \frac{(m(\mathbf{x})-0.5)^2}{v(\mathbf{x})}, \tag{13}$$

we can rewrite this criterion as

$$\mathbf{x}_+ = \arg\min_{\mathbf{x}} \frac{(m(\mathbf{x})-0.5)^2}{v(\mathbf{x})}. \tag{14}$$

4 Prototypes Description

We have developed two prototypes implementing the proposed methodologies, that is, interactive pansharpening based classification and Bayesian active learning. The purpose of these prototypes is to show how adaptation and human interaction can be used to improve the performance of classification techniques. Let us now describe in detail each one of the mentioned prototypes.

4.1 Interactive Pansharpening Based Classification

In this prototype, developed in Matlab®, whose graphical user interface is shown in Fig. 2, we address the problem of adaptively modifying the pansharpening parame-

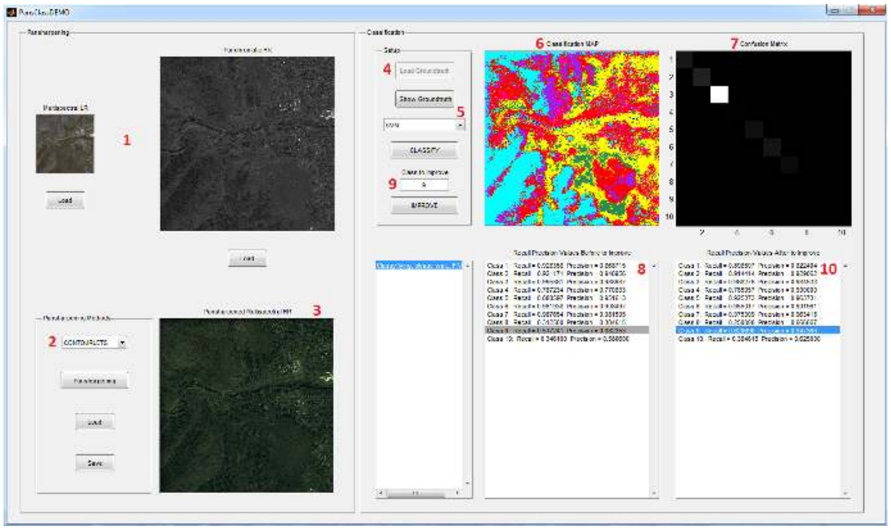

Fig. 2 Interactive Pansharpening prototype interface.

ters in order to improve the precision and recall figures of merit of the classification of a given class without significantly deteriorating the performance of the classifier over the other classes. The workflow of the prototype is as follows: First, the input LR MS and HR PAN images are loaded. Currently, the prototype only classifies Quickbird images, which have four spectral bands covering the blue, green, red, and infrared bands at 2.44×2.44 meters per pixel and a panchromatic band at 0.61×0.61 meters per pixel, although it can be easily adapted to use other remote sensing images. Then, a pansharpening method is selected from the pull-down list (marked with the number 2 in Fig. 2) and the pansharpening is performed by pressing the button "Pansharpening". The user can select between the two pansharpening methods, SAR and CONTOURLETS, described in section 2.1. We used the Matlab nonsubsampled contourlets toolbox[1] for the contourlets based method. The computed pansharpened image is depicted in the area marked as 3 in Fig. 2. This pansharpened image can be saved to a file, together with some metadata as the used pansharpened method and the value of the parameters used to obtain the image, for its later use. Alternatively, the user can load a previously computed image by pressing the "Load" button in the "Pansharpening Method" area.

The next step is to select the training and test sets. In our prototype this task is achieved by loading, using the button marked as #4 in Fig. 2, a groundtruth image, that is, an image of the same size of the PAN and pansharpened image containing, at each pixel position, the class of this pixel. In the example shown in Fig. 2, ten different classes (cars, water, forest, ...) were used. Note that only a few pixels may be classified in this groundtruth image. Pixels with unknown class have label equal to zero. From this groundtruth image, the PAN and the pansharpened images,

[1] http://www.mathworks.com/matlabcentral/fileexchange/10049

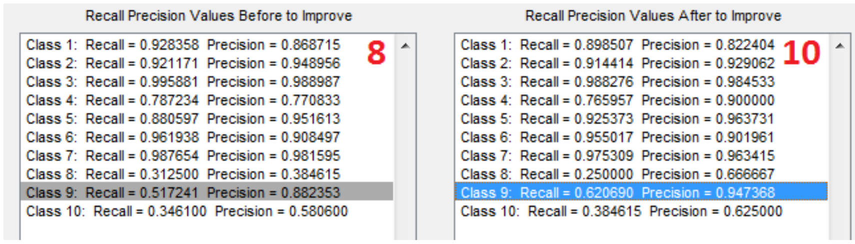

Fig. 3 Detail of the recall and precision values before and after improvement.

a training and a test set are obtained by randomly selecting, among the pixels with known class, 20% for training and the rest for testing. The features of each pixel consist of its panchromatic value and its four band values (that is, a vector with five components) together with the same five components of its four nearest neighbours. So, at each pixel location the corresponding feature vector has 25 components. In the pull-down list marked as #5 in Fig. 2 the classification method can be selected between two options: LDA and SVM, implemented by the Statistics toolbox in Matlab and libSVM[2], respectively.. By pressing the button ”CLASSIFY”, the classifier produces a Classification Map by classifying all the pixels in the image (marked as #6), and a Confusion Matrix (marked as #7) and the precision-recall values for each class (marked as #8) from the pixels in the test set.

As described in section 2.3, the user can examine the classification results, both visually and numerically, and can select a class of interest to be improved by typing its number in the text box marked as #9 in Fig. 2. By pressing the button ”IMPROVE”, the pansharpening procedure is repeated estimating the parameters from the pixels of the selected class in the training set and the new classification map and confusion matrix are obtained and displayed. The new precision-recall values are shown in the window marked as #10 in Fig. 2 side by side to the previously obtained values to help the user compare them. For a better visual evaluation of the results, Figure 3 depicts both precision and recall windows. Notice that the precision-recall values for the class of interest (class 9) have increased from (0.517, 0.882) to (0.621, 0.947). Notice also that some other classes also increased their precision or recall (see, for instance, classes 5 and 10) without significantly decreasing the figures of merit for the other classes.

4.2 *Bayesian Active Learning Remote Sensing*

The second prototype implemented exploits the Bayesian modeling and inference paradigm to tackle the problem of kernel-based remote sensing image classification. The particular problem of active learning is addressed by proposing an incremental/active learning approach based on three different methods: the maximum differ-

[2] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

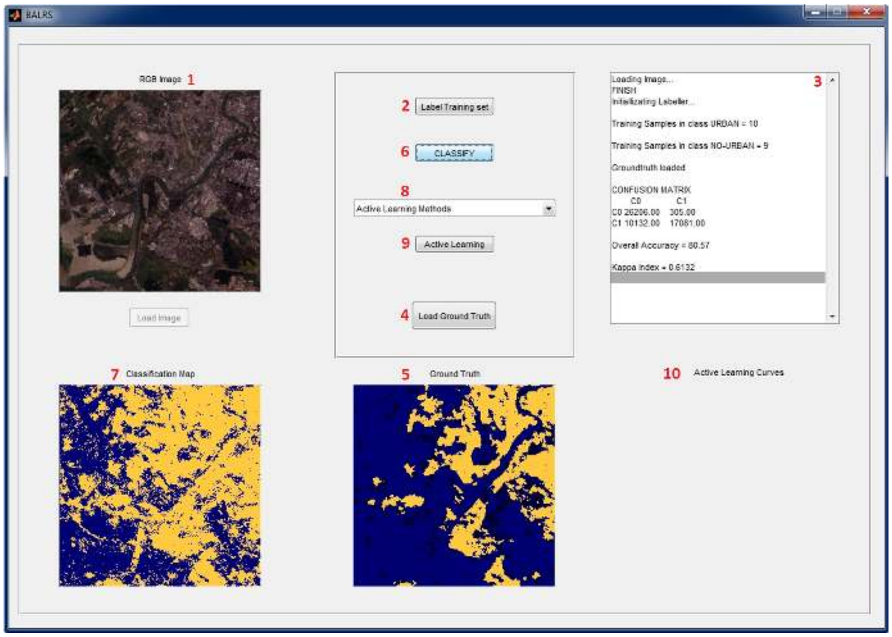

Fig. 4 Bayesian Active Learning prototype interface.

ential of entropies, the minimum distance to decision boundary, and the minimum normalized distance as described in section 3.3.

The Matlab® prototype, whose interface is shown in Fig. 4, is designed to guide the user on the challenging problem of remote sensing land cover classification from multispectral data, and in particular for urban monitoring applications. It can be easily adapted to handle other binary classification problems. Initially, the user loads a Landsat TM MS image whose bands RGB are show in window ”RGB Image” (marked as #1 in Fig. 4). Then, the user labels as urban or non-urban an initial set of pixels that are used as initial training set by pressing the button ”Label Training Set” (marked as #2 in Fig. 4). For this task, the labeler-tool depicted in Fig. 5 has been developed. This tool allows to enlarge the image, move through it, select a pixel, its class and label it by pressing the ”Label” button. In addition, it allows to load a training set from a file utilizing the button ”Load Training Set”. Once a small number of pixels of each class are labeled, the user can close the labeler tool, returning to the interface main window that will show the number of labeled samples for each class in log window (marked as #3 in Fig. 4).

Optionally, a groundtruth image can be loaded by clicking on the button ”Load Ground Truth” (marked as #4 in Fig. 4) allowing to obtain, in addition to the classification map, numerical results about the classification performance. If loaded, the Ground Truth image is depicted in the area marked as #5 in Fig. 4. In the image, the urban class is shown in bright color, the non-urban class in dark color and the background in black.

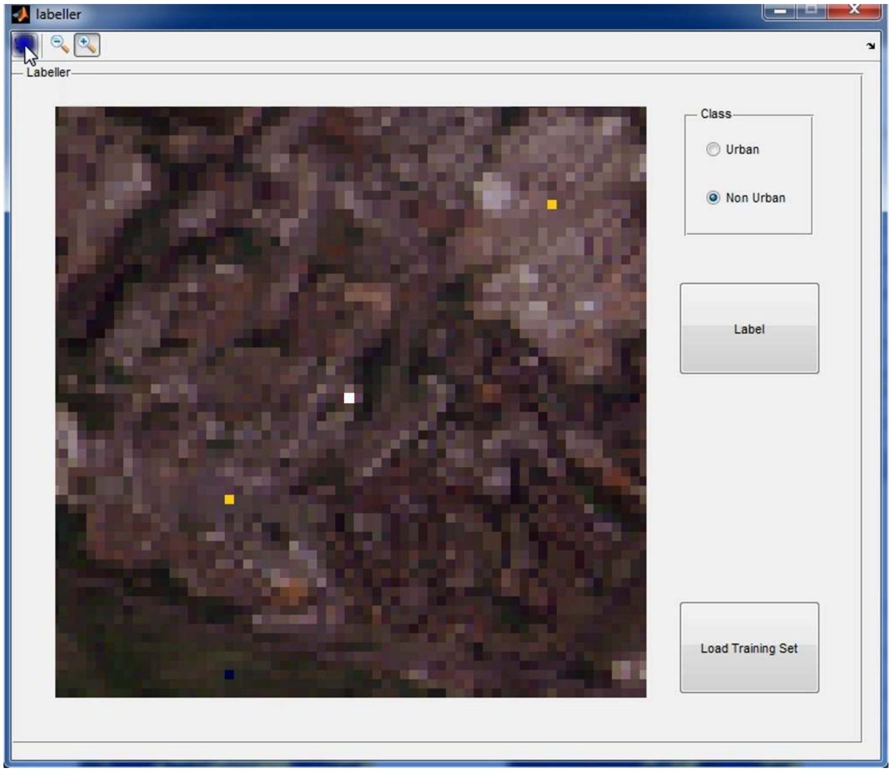

Fig. 5 Manual labeling tool interface.

Utilizing the button ”CLASSIFY”, marked as #6 in Fig. 4, the classifier is trained with the initial set and the whole image is classified. The ”Classification Map” area (#7) shows the obtained classification map and, if the ground truth was loaded, the log window shows the confusion matrix, and overall accuracy and the estimated Cohen’s kappa statistic [8] as measures of accuracy and class agreement, respectively.

From this initial classifier, the implemented methods help the user to improve the classifier performance by using one of the active learning method described in section 3.3. They can be selected from the pull-down list marked as #8 in Fig. 4. By clicking the ”Active Learning” button (#9) the most informative pixel according to the chosen active learning method is selected and the labeler automatically shows up to allow the user to label the pixel, shown in white as can be seen in Fig. 5. Once the user labels the pixel and closes the labeler, the sample is incorporated into the training set and the classifier is updated. The process continues by alternatively clicking on the ”Active Learning” button and labeling the sample. For each iteration, the learning curve, plotted in the area ”Active Learning Curves”, marked as #10 in Fig. 4, is updated. An example of the learning curve after 150 iterations is shown in Fig. 6. As can be seen in the figure, the learning curve grows significantly after a few samples have been included into the training set.

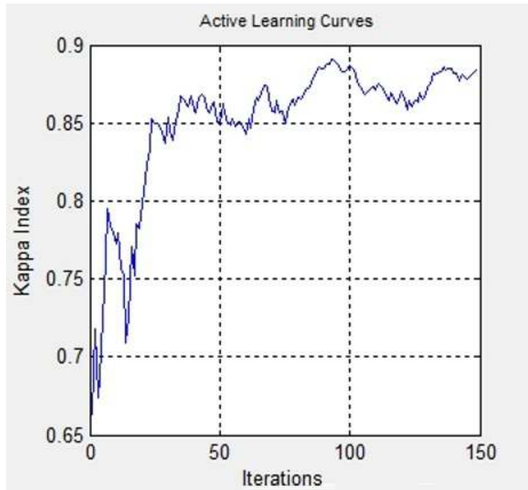

Fig. 6 Obtained Learning Curve. The kappa statistic is measured after each user query.

5 Conclusions

In this chapter we presented two prototypes to multimodal interaction in remote sensing image classification problems. The first one proves that pansharpening techniques can be used to increase the performance of classification methods when applied to MS images. We have addressed the problem of adaptively modifying a pansharpening method in order to improve the precision and recall figures of merit of the classification on a given class without deteriorating the performance of the classifier over the other classes. The validity of the proposed technique has been demonstrated using a real Quickbird image. The second prototype implements a non-parametric Bayesian active learning approach based on kernels for remote sensing image classification. We presented three different approaches for active learning: the maximum differential of entropies, the minimum distance to decision boundary, and the minimum normalized distance. The proposed prototype, dealing with the urban monitoring problem from multispectral data, show the validity of the proposed approach.

Acknowledgements

This work has been supported by the Spanish Ministry for Education and Science under projects, TIN2010-15137, EODIX/AYA2008-05965-C04-03 and the Spanish research program Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

References

1. Amro, I., Mateos, J., Vega, M.: Parameter estimation in the general contourlet pansharpening method using Bayesian inference. In: 2011 European Signal Processing Conference (EUSIPCO 2011), pp. 1130–1134 (2011)
2. Amro, I., Mateos, J., Vega, M., Molina, R., Katsaggelos, A.: A survey of classical methods and new trends in pansharpening of multispectral images. EURASIP Journal on Advances in Signal Processing **2011**(79) (2011)
3. Bandos, T.V., Bruzzone, L., Camps-Valls, G.: Classification of hyperspectral images with regularized linear discriminant analysis. IEEE Transactions on Geoscience and Remote Sensing **47**(3), 862–873 (2009)
4. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (2007)
5. Bruzzone, L., Carlin, L., Alparone, L., Baronti, S., Garzelli, A., Nencini, F.: Can multiresolution fusion techniques improve classification accuracy? In: Image and Signal Processing for Remote Sensing XII, vol. 6365, p. 636509 (2006)
6. Camps-Valls, G., Bruzzone, L.: Kernel-based methods for hyperspectral image classification. IEEE Trans. on Geoscience and Remote Sensing **43**(6), 1351–1362 (2005)
7. Camps-Valls, G., Tuia, D., Gómez-Chova, L., Jiménez, S., Malo, J. (eds.): Remote Sensing Image Processing. Morgan & Claypool Publishers, LaPorte, CO, USA (2011). Collection 'Synthesis Lectures on Image, Video, and Multimedia Processing', Al Bovik, Ed.
8. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement **20**(1), 37–46 (1960)
9. Duda, R., Hart, P.: Pattern classification and scene analysis. Wiley, New York (1973)
10. Liang, S.: Quantitative Remote Sensing of Land Surfaces. John Wiley & Sons, New York (2004)
11. Lillesand, T.M., Kiefer, R.W., Chipman, J.: Remote Sensing and Image Interpretation. John Wiley & Sons, New York (2008)
12. Molina, R., Katsaggelos, A., Mateos, J.: Bayesian and regularization methods for hyperparameter estimation in image restoration. IEEE Transactions on Image Processing **8**, 231–246 (1999)
13. Molina, R., Vega, M., Mateos, J., Katsaggelos, A.: Variational posterior distribution approximation in Bayesian super resolution reconstruction of multispectral images. Applied and Computational Harmonic Analysis, special issue on "Mathematical Imaging", Part II **24**(2), 251–267 (2008)
14. Ripley, B.D.: Spatial Statistics. Wiley (1981)
15. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
16. Ruiz, P., Mateos, J., Camps-Vall, G., Molina, R., Katsaggelos, A.: Bayesian active remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing (2012). Submitted
17. Ruiz, P., Talens, J., Mateos, J., Molina, R., Katsaggelos, A.: Interactive classification oriented superresolution of multispectral images. In: 7th International Workshop Data Analysis in Astronomy 'Livio Scarsi and Vito Di Gesu' (DAA2011), pp. 77–85 (2011)
18. Schölkopf, B., Smola, A.: Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond. MIT Press Series, Cambridge, MA, USA (2002)
19. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
20. Tuia, D., Volpi, M., Copa, L., Kanevski, M., Muñoz-Marí, J.: A survey of active learning algorithms for supervised remote sensingimage classification. IEEE Journal on Selected Topics in Signal Processing **4**, 606–617 (2011)
21. Vega, M., Mateos, J., Molina, R., Katsaggelos, A.: Super resolution of multispectral images using TV image models. In: I. Lovrek, R. Howlett, L. Jain (eds.) Knowledge-Based Intelligent Information and Engineering Systems, *Lecture Notes in Computer Science*, vol. 5179, pp. 408–415. Springer (2008)