Detection of Sea Surface Artefacts in Airborne Hyperspectral Images

Stephano Pugliese, Instituto Superior Técnico - University of Lisbon

Abstract—The aim of this work is to detect vessels, oil spills and other pollutants in an oceanographic environment using image processing techniques. Hyperspectral Imaging is a known effective tool for automatic target detection, giving a detailed identification of materials and better estimates of their abundance (avoiding false positives). This technique differs from others by recording over 200 selected wavelengths of reflected and emitted energy, making it possible to exploit the spectral signature of a given material, and distinguish between different types of materials. Furthermore, due to its high spectral and spatial resolution one can clearly identify, for example, shoreline features and areas damaged by oil spills. The work presented herein describes an attempt to perform ocean aerial surveillance via Hyperspectral Imaging, based on state-of-the-art techniques, such as unmixing procedure (e.g. endmember extraction), matching of referenced spectral signatures with pixels. An intensive study is done through a different number of papers related to the subject, in order to learn which hyperspectral methods and techniques are the best. With respect to the datasets used, the Portuguese Air Force and Navy have provided sets of videos with different kinds of vessels and slicks, that one can see in a real life situation. An overview of the methods and techniques of the state of the art is presented throughout the report. More in depth there is a discussion about the use of Support Vector Machines in hyperspectral imagery context. Two different approaches are designed: image classification with Support Vector Machines on raw hyperspectral data cube and image classification with Support Vector Machines on abundance maps collected with the Vertex Component Analysis algorithm (state of the art endmember abundance maps estimation and hyperspectral data dimensionality reduction algorithm). This report documents the evolutive process of finding the best methodologies to face the problem formulated, the strategy planned and its application to attain the desired results.

Index Terms—Hyperspectral Imaging, Support Vector Machines, Vertex Component Analysis, Dimensionality Reduction, Endmembers abundance maps, image classification

I. INTRODUCTION

Ever since the human race saw the first rainbow in the sky, and rationally thought about it, certainly and immediately a wide variety of questions arose. How is possible to see the whole spectrum of visible light? Obviously these questions were answered throughout the years, with the contributions of a great number of scientists and engineers. Image Processing is a field in Computer Vision, where Digital Processing techniques help in manipulation of digital images using computers. These techniques are an important and very useful tool nowadays to solve a quite big range of problems in our daily life.

In this work a system to detect vessels was designed, with further applications in oils spills and other pollutants in oceanographic environments. The main goal was to build a classification system that was able to process hyperspectral imagery and eventually classify with good accuracy. The proposed solution was based on two different approaches described herein. Hyperspectral imaging is a technique able to collect useful information of a certain target across all the electromagnetic spectrum, facilitating the identification of objects. Furthermore, a distinctive characteristic of Hyperspectral Imaging is that it is possible to acquire a wide number of bands, with a narrow spacing in the spectrum. It is a valuable tool in various fields of study, such as, in medicine (early detection and treatment of many life-threatening medical conditions), in agriculture (crops health monitoring) and in food industry (goods and quality assessment of food products). Furthermore, this technology is non-contact and non-destructive making it ideal for means of detection.

Various methods and techniques were applied to the collected data. First, a robust image alignment was performed, so that image classification was the most accurate possible. Additionally, the endmember extraction of the hyperspectral imagery is a very important aspect to take into account, as there are some different algorithms for features extraction and dimensionality reduction. Hence the Vertex Component Analysis algorithm proposed by [1] was adopted to perform endmember extraction. A hyperspectral image cube can be seen as a set of images layered on top of one another, where one specific wavelength band is represented by each of those images, and contains the spatial and spectral information of an object. Through this information it is possible to obtain the single characteristic of a given object referred to as Spectral Signature.

As for image classification it is known that in many cases information classes that are spectrally similar and thus not separable when using standard low-dimensional data, can nevertheless be separated with a high degree of accuracy in higher dimensional spaces. Although the classification of this type of data in higher dimensions could become a real problem for parametric classifiers such as Gaussian Maximum Likelihood. Another aspect to take into account is the large number of variables produced alongside the large number of parameters to be estimated from a limited number of training samples. The usage of non-parametric classifiers is advantageous in terms of the insensitivity to the problem’s dimensionality. It is common sense nowadays that Support Vector Machines are an efficient classifier and that is why it was used in this work. Regarding dimensionality reduction there are a significant number of algorithms from the literature that perform well such as: hyperspectral signal subspace identification by minimum error (HySime) by [2] , second moment linear (SML) by [3]
and noise-whitened HarsanyiFarrandChang (NWHFC) by [4]. This phenomenon is a process that consists in transforming hyperspectral imagery into a reduced form with reduced dimensionality. It is a very powerful way to deal with the problem arose by the curse of dimensionality as stated by [5].

II. Dimension Reduction and Feature Extraction in Hyperspectral Imagery

Dimension reduction (DR) is a process by which a given hyperspectral image data is transformed into a reduced dimensionality form. These kind of methods are very important in a wide range of different areas. An aspect to note is that by performing DR irrelevant variance in the data is detected and eliminated. With respect to feature reduction techniques there are two main categories used very frequently in hyperspectral imagery: feature extraction (FE) and feature selection. Three of the best and most common DR and FE techniques for hyperspectral images are Principal Component Analysis (PCA) by [6], the vertex component analysis (VCA) and the independent components analysis (ICA) by [7], thus the VCA was the chosen one to deal with this work’s hyperspectral datasets. Very briefly, in order to understand how the VCA algorithm works, one should take into consideration two important aspects: an endmember is a pure spectral signature for a given information class; the endmembers are vertices of a simplex; the affine transformation of a simplex is also a simplex. Summarily the algorithm iteratively projects data (spectral vectors) onto a direction orthogonal to the subspace spanned by the endmembers already determined leading to a new endmember signature that is the extreme of that projection. The algorithm iterates until all endmembers are exhausted, i.e, when all the \( p \) endmembers are extracted. Beside the fact that the number of endmembers is much smaller than the number of hyperspectral bands, the abundance maps of the \( p \) endmembers can accurately provide the same spectral information as the hyperspectral raw image with \( N_s \) spectral bands.

III. Image Alignment

Image alignment can be simply described as the process of matching one image called template, with another image. There are several applications for image alignment, such as motion analysis, tracking objects on video etc. [9] proposed at the time a revolutionary technique that used image intensity gradient information to search for the best match between two images named Lucas-Kanade tracker. Their technique, amongst others, takes advantage of the fact that the images are already in approximate registration. Furthermore, it can deal with linear distortions including rotation, scaling and shearing. With this algorithm it was possible to track the boat throughout the different frequencies at the same frame (time-instant). Another technique was put into consideration to perform the desired operation: The Scale-Invariant Feature Transform (SIFT) is a very useful technique widely used in Computer Vision. The main idea behind this method is the usage of a feature detector that will extract from a given image a certain number of frames or designated regions, in a consistent manner depending on the variations of the illumination and other viewing conditions. The feature descriptor will eventually associate the regions with a signature that identifies their appearance in a very robust and compact fashion. Applying this method to the dataset would have helped to perform image alignment, but unfortunately, after several tests, the technique turned out to be unsuccessful with the aforementioned dataset due to the lack of enough saliences in the scene. Alongside the fact that the image scene is mainly composed of regions of sea and simply a boat, it was very difficult for the algorithm to perform well. Given the difficulty of the process, and since image alignment was not the focus of the thesis, it was decided that the most efficient way to do the alignment was to do it manually. This is described next:

1) Initiate the Control Point Selection Tool in MATLAB and start by choosing a set of points in the reference band that are visible in all the other bands, and mark the corresponding set of points in the other bands. Once again, it is important to note that the alignment was performed taking into consideration band 12 as the central band that is, the alignment was performed with the pairs band12 and band1, band12 and band2, band12 and band3 · · · until band12 and band24;
2) After a proper choice of the set of points for the first pair, export the set of points to the workspace. The points are exported in a .mat file containing the values of the fixedPoints and the movingPoints of the current pair. Repeat the process until all pairs are exhausted.
3) Create a cell array that stores each pair
4) For each non-reference band \( i \), calculate the geometrical difference between the marked point coordinates and the reference points with a simple subtraction operation as follows:
\[
\Delta x_i^p = fixedPoints_i^p(1,1) - movingPoints_i^p(1,1);
\]
\[
\Delta y_i^p = fixedPoints_i^p(1,2) - movingPoints_i^p(1,2),
\]
creating two new variables called \( \Delta x_i^p \) and \( \Delta y_i^p \), where \( p \) is the point index;
5) After calculating the geometrical difference between the two images, one should now perform image translation, that is, add or subtract a certain value line and column-wise so that the pixels move accordingly. This can be done, for instance, with the MATLAB function imtranslate. For each image I, from 1 to 25, the respective correction will be done using a 2-dimensional translation vector, that is composed by the two mentioned variables making it possible to translate the image accordingly and eventually change the image matrices’ content position.

After carefully observing the output images of the previous process, it is perceived that some residual translation remained, since the alignment was not as perfect as wanted. That being so, an ultimate technique was implemented that would certainly perfect the manual alignment. A method was developed enabling alignment via images sitting on different RGB channels as shown below:

1) A white canvas with 3 channels was created in order to contain the images to be aligned;
2) Using the moving arrows, it was possible to dispose the
images in order to align them:
3) The resulting figure presented green or red pixels in the scene as the images were moving to find the best match, as seen in 1 (a):
4) If a good match between two images was found, the figure should display the pixels in yellow as shown in 1 (b)

![Images not aligned](image1)

(a) Images not aligned

![Images consistently aligned](image2)

(b) Images consistently aligned

Fig. 1. Channels R (RED) and G (GREEN) containing the images to be aligned

The image alignment was accomplished with success and the hyperspectral data was properly prepared to be used in the coming applied methods such as the endmember extraction and abundance maps estimation as well as the image pixel classification via Support Vector Machines.

IV. IMAGE PROCESSING TECHNIQUES

A. VCA - Vertex Component Analysis

The Vertex Component Analysis is the algorithm proposed for estimating the abundance maps of endmembers and to reduce the dimension of hyperspectral data. Performing the unmixing of hyperspectral data is the decomposition of the pixel spectrum into a linear mixture of the spectra of different chemical constituents, commonly referred as endmembers spectral signatures and their corresponding abundance fractions [10]. It is important to note that since the number of endmembers is much lower than the number of spectral bands, these abundance maps can be considered as a compact representation of spectral information provided by the hyperspectral image [1] and [11]. To geometrically understand the algorithm, let one assume that a given spectral vector $r \in \mathbb{R}^L$, where $L$ is the number of bands, is a linear combination of $p$ signatures $m_j$, weighted by the respective abundance fraction $\alpha_j$, for $j = 1, ..., p$. Assuming this linear mixing scenario, it yields to:

$$ r = x + n = M\gamma\alpha + n $$

where $M \equiv [m_1, m_2, \ldots, m_p]$ is the mixing matrix, $p$ is the number of endmembers present in the covered area, $s \equiv \gamma\alpha$ ($\gamma$ is a scaling factor that models the variation of illumination due to surface topography), $\alpha \equiv [\alpha_1, \alpha_2, \ldots, \alpha_p]^T$ is the abundance vector containing the fractions of each endmember and $n$ models the additive noise. VCA is a fast method that comprehends two different facts:
1) the endmembers are vertices of a simplex
2) the affine transformation of a simplex is also a simplex

Summarily the algorithm iteratively projects data (spectral vectors) onto a direction orthogonal to the subspace spanned by the endmembers already determined leading to a new endmember signature that is the extreme of that projection. The algorithm iterates until all endmembers are exhausted, i.e., when all the $p$ endmembers are extracted. Beside the fact that the number of endmembers is much smaller than the number of hyperspectral bands, the abundance maps of the $p$ endmembers can accurately provide the same spectral information as the hyperspectral raw image with $N_s$ spectral bands. Pseudo-codes of some important procedures and the actual implementation of the VCA algorithm are presented in algorithms

V. SUPPORT VECTOR MACHINE CLASSIFIER APPLICATION ON PROVIDED DATASETS

A large set of classification methods are available in the literature. All of them are divided into two distinct groups: Supervised Methods (eg. Multilayer perceptrons, Naive Bayes classifiers, Decision Trees, Maximum Likelihood, Spectral Angle Mapper and Support Vector Machine) and Non-Supervised Methods (e.g. K-means clustering algorithm, Expectation-Maximization Algorithm and Principal Component Analysis). Kernel-based methods are proved to be very powerful algorithms for hyperspectral image classification thanks to developments conducted by [8]. To perform image classification some preceding steps had to be considered. Image alignment was very important in this work due to the fact that the camera used for acquiring the hyperspectral imagery had the ability to collect spectral information from a wide range of different bands, in different time instants, so that the hypercube could be formed. Given the difficulty of the process, and since image alignment algorithms from the literature were not as effective as one needed the most efficient way to do the alignment was to do it manually. SVMs are very useful to deal with large input spaces efficiently, since it can produce
Algorithm 1: Endmember extraction using the VCA algorithm

1 function data (band);
   Input : Each band, already properly aligned, numbered from 0 to 24
   Output: x, Columns, Lines, L, wavlen, Bands, that are the used arguments in VCA algorithm
2 Set the number of columns, lines and bands of the imagery;
3 for i = 0 : 24 do store all bands in an array
4    im{i + 1} = imread(aligned_band_i);
5    images{1 + 1} = reshape((imi + 1), 1, total_number_pixels);
6    r_RGB(i + 1,:) = images{1 + 1}
7 end
8 Store all created variables;
9 function functional_algorithm (x, Columns, Lines, L, wavlen, Bands);
10 Input : Output from function data and user definition of the number of endmembers p
11 Output: A estimated mixing matrix;
12 Index pixels that were chosen to be the most pure;
13 Rp: Data matrix R projected;
14 S: estimated abundance fraction;
15 All the aforementioned outputs are variables from the actual VCA algorithm designed by [1]
16 Read the number of pixels from the input data;
17 Set the desired positive integer number of endmembers in the scene p;
18 Set the desired signal to noise ratio;
19 Make sure the input data x is of type double;
20 Run the VCA algorithm with inputs: x, p, SNR;
21 Using the output variables from VCA, calculate the estimated abundance fraction using equation 1:
22 S = pinv(A) * Rp;
23 Store the estimated abundance fraction in a variable to use afterwards in classification or training;

Algorithm 2: VCA algorithm

1 function VCA (R, p, SNR);
2 Input : matrix with dimensions L x N (where L is the number of bands and N the total number of pixels) R, positive integer number of endmembers in the scene p
3 Output: estimated mixing matrix A, pixels that were chosen to be the most pure index, Data matrix R projected
4 SNRth = 15 + 10log10(p)dB;
5 if SNR is higher than SNRth then
6    project the data onto a subspace of dimension p;
7    X := U_p^T R;
8    U_d is obtained by SVD from RR^T / N, where
9    R := [r_1, r_2, ..., r_N];
10   [Y]_{i,j} := [X]_{i,j} / ([X]_{i,j}^T u) is the projective projection;
11 else
12    project the data onto a subspace of dimension p − 1;
13    U_d is obtained by PCA;
14 end
15 Auxiliary matrix A with p x p dimensions, is initialized, which stores the projection of the estimated endmembers signatures:
16 A := [e_a [0] ... [0]]; e_a = [0, ..., 0, 1]^T;
17 for i := 1 to p do
18   Vector f orthonormal to the space spanned by the columns of A is randomly generated;
19   Data is projected onto f;
20   store the endmember signature corresponding to the maximum between pure pixel a and pure pixel b,
21   assuming that the pure endmembers occupy the vertices of a simplex: a ≤ f^T [Y]_{i,1} ≤ b, for
22   i = 1, ..., N
23 end
24 if SNR is higher than SNRth then
25   compute the columns of matrix \( \hat{M} \) that contain the estimated endmembers signatures in the
26   L-dimensional space: \( \hat{M} := U_d [X]_{i, indices} \), with dimensions L x p
27 else
28   compute the same matrix \( \hat{M} \) but take into account
29   the mean of R: \( \hat{M} := U_d [X]_{i, indices} + \hat{F} \);

sparse solutions and work solidly well with noisy samples. There are some aspects regarding the automatic analysis of hyperspectral data to bare in mind such as the increasing data dimensionality, atmospheric effects, small ratio between the number of available training samples and the number of features and the Hughes phenomenon [12]. The usage of non-parametric classifiers like SVM’s is advantageous in terms of the insensitivity to the hyperspectral data’s dimensionality.

A. Datasets Labeling, Training Sets and Test Sets

The first dataset of two consisted in a sequence of images (bands) of a given boat with a resolution of 648 by 1024 pixels. One had access to a set of the given frames for 25 different bands captured by a Rikola Snapshot Hyperspectral Camera. It is important to note that with this camera all pixels are true image pixels (no interpolation is used i.e. real spectral response). It has the capacity to acquire a maximum of 380 different spectral bands, for different time instants. The second dataset consisted also in a sequence of images in 25 different bands with a resolution of 648 by 1024 pixels, containing a boat (although different in terms of shape) in another relative position in the oceanographic scene. Both datasets are shown in Fig. 2. In order to test the assembled SVM classifier, the second dataset was used. Two information classes were defined: Class "sea" assigned as 1, and class "boat" assigned as −1. A pixel by pixel labeling was performed. To do so, some image processing techniques were employed such as specifying a polygonal region of interest, in this case pixels that represented the boat class, so that it was possible to get
the indexes of those pixels and label them accordingly in the dataset provided.

The training data was composed by part of the hyperspectral data cube, attained after manual image alignment of the data, with 1000 pixels of boat samples and other 1000 samples of sea pixels, in 25 different features. Also a labeling array was designed with 2000 by 1 dimensions. The Training variable was set with 26 columns (25 predictors and one label column): 2000 x 26. This variable was used in the Classification Learner app, a built-in MATLAB application. After some tests and experiments for different classifiers the Fine Gaussian Support Vector Machine proved to be the best classifier. The testing set is a key factor to evaluate whether a classifier is robust and accurate.

Test Set: 25 aligned bands with a resolution of 518 by 1007 pixels, with a 25 by 521626 dimensional double type array. In order to test the classifier, a structure was retrieved from the Classification Learner app, that contained the classifier itself. A simple line of code was enough to get the classification results that will be presented in chapter VI. To correctly and robustly analyse the results after performing image classification via Support Vector Machines, some statistical measures were collected. There is a brief description of two parameters in special, amongst others: precision or positive predictive value, that gives an idea of how relevant are the results and recall or sensitivity, that measures the proportion of positives that are correctly identified as such. In this particular case, for precision, where a "true positive" is the event where a pixel was correctly identified as boat or sea, and the test confirms it, is the fraction of pixels that the system detected correctly. The values TruePositives (TP), TrueNegatives (TN), FalsePositives (FP) and FalseNegatives (FN) are part of the so called confusion matrix and are very crucial to calculate the statical measures such as: Precision, Recall, Specificity, Fallout, Accuracy, F-Score, Markedness and Informedness.

To better understand the relation between recall and precision on one hand and sensitivity and specificity on the other hand, it is useful to see the aforementioned matrix. It is more difficult to get a good precision than a good specificity while keeping the sensitivity/recall constant. There is usually a trade-off between sensitivity and specificity (or recall and precision). For instance, in information retrieval, if one cast a wider net, it will detect more relevant documents/positive cases (higher sensitivity/recall) but it will also get more false alarms (lower specificity and lower precision). Considering all this, whether a classifier is "good" depends a varied number of factors.

As for the second approach one should take into consideration some important phases. Using the designed training set some new procedures were held. After getting the mixing matrix from the output of the VCA algorithm, where the number of endmembers was defined as \( p = 2 \), one had to calculate the abundance fractions for the training set used. The abundance fractions are calculated derived from equation 2 in the following manner:

\[
S_{\text{train}} = (M)^* \times \text{Vec}_{\text{train}} \tag{2}
\]

where \( S_{\text{train}} \) is the training abundance fractions (regarding dataset #1), \( M^* \) is the pseudo inverse of the training mixing matrix and \( \text{Vec}_{\text{train}} \) is a portion of the first dataset used for training. With respect to the training data, it contained 2000 observations. A suitable label array was constructed and a new variable Training was set with 3 columns (2 predictors and one label column containing the two response classes).

Using the first dataset for training and for testing the classifier was not the initial intent (even though the training set used was only a small portion of the entire set) but due to the lack of different datasets of this type this approach was essential. Regarding the morphological erosion operation, a proper threshold was determined, that one would later use on the second dataset after classification. A comprehensive series of steps describe the previous explanation:

1) Plug the hyperspectral data cube into VCA to calculate training mixing matrix \( M \);
2) Calculate the abundance fractions with: \( S_{\text{train}} = (M)^* \times \text{Vec}_{\text{train}} \);
3) Perform the SVM training for the abundance fractions;
4) With the used dataset for training, test the classifier, and with the output perform morphological erosion and set up a threshold.

As what occurred in the first approach, training for each SVM classifiers was performed, and in this case the best option provided was the Fine Gaussian SVM classifier.

For the classification procedure one had to use the second dataset. In this case the entire set was used so that it was possible to attain a binarized image at the output of the SVM classifier representing the whole image. The first step was to get the abundance fractions for Dataset 2, but using again equation 2, where \( M \) is the training mixing matrix and the \( \text{Vec}_{\text{test}} \) is the second entire Dataset. Using the calculated abundance fractions, plug it into the SVM classifier. On the output of the classifier, perform some morphological operations (erosion) using the threshold determined for the training set.

A comprehensive series of steps are described in the following explanation:

**PROCESS INPUT:** New_Test_Set, mixing matrix \( M \), SVM classifier, threshold

1) Get the abundance fraction for the test set: \( S_{\text{test}} = (M_{\text{training}})^* \times \text{Vec}_{\text{test}} \);
2) Perform classification with test set;
3) Apply morphological operations (erosion) to the output of the classifier with the already determined threshold.
VI. RESULTS

A. SVM classifier on raw hyperspectral data cube

After performing the Testing phase, the output of the classifier is analysed. Despite the fact that the second Dataset was used to test the classifier, results using the first dataset are also presented in this section (although one might say that the Test Set must never contain any data from the Training Set), so that one can observe and analyse the differences between the outputs of each Test Set. The outputs of both datasets are illustrated in figure 3.

(a) Example of vessel detection in an oceanographic airborne imagery, using Dataset #1 and a Fine Gaussian SVM for the classification stage. This are the direct outputs in a binary fashion

(b) Example of vessel detection in an oceanographic airborne imagery, using Dataset #2 and a Fine Gaussian SVM for the classification stage. These are the direct outputs in a binary fashion

As it is possible to observe on both figures, the boat location is evident in the scene, although there are some pixels, FP, mainly sun reflections and foam trail, that affect the statistical measures. In the first case, considering dataset #1 2272 boat pixels in a total of 2408 were correctly classified (TP), yielding to a boat classification score of 94.35%. Considering dataset #2 one retrieved 87.2% of the pixels that belong to the boat information class (2423 pixels in 2778). The latter values are the so-called Recall.

B. SVM classifier on abundance maps

After performing the Testing phase, the output of the classifier is analysed. Results for both datasets are presented in this section, so that one can observe and analyse the differences between the outputs of each Test Set. The outputs of both datasets are illustrated in figures ?? and 4.

(a) Example of vessel detection in an oceanographic airborne imagery, using Dataset #1 and a Fine Gaussian SVM for the classification stage. This are the direct outputs in a binary fashion

(b) Example of vessel detection in an oceanographic airborne imagery, using Dataset #2 and a Fine Gaussian SVM for the classification stage. These are the direct outputs in a binary fashion

As it is possible to observe on both figures, the boat location is also evident in the scene, although there are some pixels, FP, mainly sun reflections and foam trail, that will affect the statistical measures. In the first case, considering dataset #1 1992 boat pixels in a total of 2408 were correctly classified (TP), yielding to a boat classification accuracy of 82.72%. Considering dataset #2 one retrieved 79.55% of the pixels that belong to the boat information class (2210 pixels in 2778).

VII. STATISTICAL EVALUATION FOR BOTH APPROACHES AFTER SVM AND AFTER MORPHOLOGICAL OPERATIONS

With the results, a quantitative and qualitative analysis for both classifiers is presented and how these behaved with the different datasets. Starting with the first approach in subsection VII-A and the second approach in subsection VII-B.
A. Approach 1

A SVM Classifier was used to test some hyperspectral data cubes (Dataset #1 and Dataset #2). The statistical measures, already described in subsection V-A, are presented next:

Regarding Dataset #1:
- Precision: 0.1221
- Recall: 0.9435
- Specificity: 0.9685
- Fallout: 0.0315
- Accuracy: 96.84 %

Regarding Dataset #2:
- Precision: 0.0389
- Recall: 0.8722
- Specificity: 0.8912
- Fallout: 0.1088
- Accuracy: 89.11 %

Analysing the results, it is clear that the classifier behaved better when the test set was composed by Dataset #1, which is something that one was waiting to happen. Starting with precision, for the first dataset we have a higher value for this measure, thus both values are relatively low. This quantity denotes the proportion of Predicted Positive cases that are correctly Real Positives, so in this case the proportion is not satisfactory. If Precision, or Confidence are this low, then Recall or Sensitivity must be high, due to their inverse relationship, i.e., as Precision increases, Recall falls and vice versa. In terms of Specificity or Inverse Recall referred as the proportion of Real Negative cases that are correctly Predicted Negative and also known as the True Negative Rate the values are fairly good. This measures how the classifier avoids FP's, which in this case is significantly accurate and a good result as the number of false positives is relatively low (16337 FP's in a total of 521026 pixels for Dataset #1 and 59858 FP's in a total of 552720 pixels for Dataset #2). As for Fallout or False Positive Rate that measures the proportion of Real Negatives that occur as Predicted Positive in both cases the values are relatively low which is good.

As for the first statistical measure, precision, the performance of both datasets is rather satisfactory, just like in the first approach. Thus, a very reasonable explanation to these low values can be given. With respect to the pixels that lay in the boat, we can see, as illustrated in figure 5, that some pixels are denoted as false positives although they are in did part of the boat. The misleading assignment is due to the chosen ground truth. A safety margin in this area was set, that noticeable contains a great percentage of pixels that here are described as FP's. This problem is even more evident with Dataset #1, also in figure 5. Whilst this reason was not mentioned in the first approach, it completely addresses also the problem of precision values in that case.

Recall values for both cases are reasonable, specially in the first dataset, as expected. The true positive rate, the same as recall, can be viewed as the proportion of Real Positive cases that are correctly Predicted Positive, which in this context is quite good. As seen before, the inverse of the latter measure is the true negative rate. With this approach both quantities revealed accurate results, as it is prominent by the number of TN for both datasets 1 and 2: 487169 and 503563 respectively. Addressing again the ground truth choice problem mentioned above, as these quantities measure how the classifier avoids FP's the values are quite reasonable for the number of FP's detected: 32272 FP's in a total of 521026 pixels for Dataset #1 and 46379 FP's in a total of 552720 pixels for Dataset #2. The low values for fallout are good in both cases and accurate regarding what one was waiting.

As the classifier in the first approach, this one lies above the diagonal, hence it represents a good classifier. Looking now to the other measures, in terms of informedness and markedness we have: Informedness = 0.8482 and Markedness = 0.2862. As for Dataset #2: Informedness = 0.7112 and Markedness = 0.2499. The values in both cases for Informedness are good, thus again higher for the first Dataset. As for markedness both values are low which is good. As for accuracy both have high accuracies, and an even more positive aspect of this results is that both datasets seem to be best classified with the second approach, rather than the first. This was exactly one of the objectives of this classification part. It was expected that the second approach would have been

B. Approach 2

In the same way as the first approach a SVM classifier was used to test both datasets, although in this approach the data used was not hyperspectral raw data cubes, as mentioned in V, but the outputs of VCA algorithm and equation 2. The statistical measures in this case are:

Regarding Dataset #1:
- Precision: 0.0581
- Recall: 0.9117
- Specificity: 0.9379
- Fallout: 0.0621
- Accuracy: 93.78 %

Regarding Dataset #2:
- Precision: 0.0455
- Recall: 0.7955
- Specificity: 0.9157
- Fallout: 0.0843
- Accuracy: 91.51 %
better to classify data, and the presented results demonstrate that consistently.

VIII. MORPHOLOGICAL OPERATIONS

In order to isolate the boat in the scene some coding was applied on the images after exiting the SVM classifier and using the pre-determined threshold (2910), as follows in pseudo-code 3. The application of algorithm 3 is the ultimate operation over the imagery.

Algorithm 3: Algorithm for image treatment

```
1 function clean_image YP;
   Input: A type double array with dimensions N pixels x 1 that is the direct output from the classifier
   Output: Clean eroded image - eroded_image;
2 Reshape the output of the classifier in order to get the image scene correctly;
3 Store it in a variable called result;
4 Create a new variable IM that stores the complemented image of result;
5 Using bwareaopen function, and with the pre-determined threshold, remove all the connected objects that have fewer than P pixels;
6 Store the eroded image in a variable;
```

- Fallout: 0.0594
- Accuracy: 94.01 %

Regarding Approach 2:
- Precision: 0.0644
- Recall: 0.7955
- Specificity: 0.9416
- Fallout: 0.0584
- Accuracy: 94.09 %

An overview over the statistical measures gives an idea of how accurate and robustly the two different approaches produced the displayed results. In terms of accuracy one must highlight the better performance of classification on abundance maps. The main goal, as stated at the beginning of this thesis was to build a system that would be able to process hyperspectral imagery and eventually classify with good accuracy and produce reliable outputs. This objective was met, at least in terms of accuracy. Hence, one can not state that one classifier was better than the other just looking over the accuracy scores. Comparing the statistics after the SVM classification with the statistics after the applied morphological operations a slight, but noteworthy improvement in the results occurs. In terms of the other measures the reasons are very similar to the ones given at the beginning of section VII. It is important to look at the precision values in both cases, that are very low as stated above. In a hypothetic situation where it is not required a high value for recall, i. e., if there are no issues on having more false negatives, the precision measure, because it behaves inversely to the recall, could improve, meaning that at the end of the day one might have less FP’s. Translating this situation to this context, this would mean that the trail of foam behind the boat would be less noticeable, and that some pixels along the borders of the boat would be more misclassified.

The outputs for the system illustrated in figure 6 are presented next in figure 7. The results also just for the second dataset are displayed according to the two different considered approaches.

Regarding the final ROC space illustrated in figure 8 the difference between the two approaches in terms of the classifiers performance is quite small. However, if one would take into
account only this ROC analysis the classifier used in the first approach seems more reliable than the one from the second approach.

This work has a major contribution amongst others. The designed classifier met the goals that were initially entrenched, allowing in the future an improvement always possible in terms of the classifier modelling. By using proper datasets, and by performing the Processing steps described broadly throughout this report good results can be attained. Although the second approach proved to be better than the first, one should not disregard at all the results provided by the first approach. As expected and with what one described thoroughly during this report the Support Vector Machines are very reliable classifiers to use in the Hyperspectral Imagery context.

ACKNOWLEDGMENT

The author would like to thank the effort and commitment showed by Professor Alexandre Bernardino since the first time almost one year ago. His guidance during all the process of this work was meaningful and essential to me.

REFERENCES


