Lattice Computing and Hyperspectral Image Processing for Human Detection and Identification

By

Ion Marqués Bailón

Submitted to the department of Computer Science and Artificial Intelligence in partial fulfillment of the requirements for the degree of Doctor of Philosophy

PhD Advisor:

Prof. Manuel Graña Romay
at The University of the Basque Country (UPV/EHU)

Euskal Herriko Unibertsitatea
Universidad del País Vasco
Donostia - San Sebastián
2014

eman ta zabal zazu

Universidad del País Vasco Euskal Herriko Unibertsitatea

AUTORIZACION DEL/LA DIRECTOR/A DE TESIS
PARA SU PRESENTACION

Dr/a. ____con N.I.F.__________________________

como Director/a de la Tesis Doctoral: ____

____

____

realizada en el Departamento ____

____

por el Doctorando Don/ña. ____ ,

autorizo la presentación de la citada Tesis Doctoral, dado que reúne las condiciones necesarias para su defensa.

En __________________a _____de ____________________de _____

EL/LA DIRECTOR/A DE LA TESIS

Fdo.: _________________________________

eman ta zabal zazu

Universidad del País Vasco   Euskal Herriko Unibertsitatea

CONFORMIDAD DEL DEPARTAMENTO

El Consejo del Departamento de ____

____

en reunión celebrada el día ____ de ____________ de _____ ha acordado dar la conformidad a la admisión a trámite de presentación de la Tesis Doctoral titulada: ___________

____

____

dirigida por el/la Dr/a. ____

____

y presentada por Don/ña. ____

ante este Departamento.

En _________________a _____de ____________________de _____

Vº Bº DIRECTOR/A DEL DEPARTAMENTO   SECRETARIO/A DEL DEPARTAMENTO

Fdo.: ___________________________________   Fdo.: ______________________________

eman ta zabal zazu

Universidad del País Vasco   Euskal Herriko Unibertsitatea

ACTA DE GRADO DE DOCTOR

ACTA DE DEFENSA DE TESIS DOCTORAL

DOCTORANDO DON/ÑA. ____
TITULO DE LA TESIS: ____
____
____

El Tribunal designado por la Subcomisión de Doctorado de la UPV/EHU para calificar la Tesis Doctoral arriba indicada y reunido en el día de la fecha, una vez efectuada la defensa por el doctorando y contestadas las objeciones y/o sugerencias que se le han formulado, ha otorgado por____________________la calificación de:
unanimidad ó mayoría

Idioma/s defensa: ____

En __________________a _____de ____________________de _____

EL/LA PRESIDENTE/A,   EL/LA SECRETARIO/A,

Fdo.:   Fdo.:

Dr/a: ______________________   Dr/a: ________________________

VOCAL 1º,   VOCAL 2º,   VOCAL 3º,

Fdo.:   Fdo.:   Fdo.:

Dr/a: ______________________Dr/a:________________________Dr/a:_____________________

EL/LA DOCTORANDO/A,

Fdo.: ________________________

Acknowledgments

On one of my first days as his doctoral student Prof. Manuel Graña told me: *"It's not the same to say that you* want *as it is to say that you* need*. Words matter."* I do indeed want to give special thanks to him, for his initiative, encouragement and commitment.

I am very grateful to every researcher that has helped me on this tangled road; particularly Prof. Guangbin Huang, Prof. Hong Wang and the guys from the Autonomous Robotics Research Laboratory at Nanyang Technological University in Singapore and Prof. Miguel Vélez-Reyes and his team from The University of Texas at El Paso in El Paso, USA. I also want to give a big shout-out to my colleagues at the Computational Intelligence Group (GIC).

I'd like to thank all my closest friends. They have always been there for me, with a nice word, a big smile and a jar of beer.

Finally, I would like to thank my family -specially my parents and brother- for their unending support.

Ion Marqués Bailón

"I don't wear a suit because I am blue-collar. I am a proletarian of science."

Prof. Manuel Graña

Lattice Computing and Hyperspectral Image Processing for Human Detection and Identification

by
Ion Marqués Bailón

Submitted to the Department of Computer Science and Artificial Intelligence on October 10th, 2014, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

The thesis has a main application topic, directly related to face based biometric identification, which proceeds in two substantial steps, first face localization in the image, second face classification based on features extracted from the image. Regarding face localization, we have focused on the problem of skin identification in a novel imaging device, namely the hyperspectral cameras that allow a fine sampling of the light spectrum, so that the information gathered at each pixel is a high dimensional vector. The problem can be treated as classification problem, where we have proposed the use of active learning strategies to provide an interactive robust solution able to provide high accuracy in a short training/testing cycle. Also it can be treated from the point of view of spectral unmixing, wher endmember induction algorithms find close representatives for the decomposition of the image into regions of high abundance of skin-related spectra. Of special interest is the contribution in this thesis regarding the application of lattice computing algorithms for endmember induction, which are combined with sparsity numerical methods in order to perform endmember selection which competitive with other classical algorithms in terms of quality of skin detection. Regarding face recognition, we have contributed new methods for feature extraction, based on lattice computing algorithms and hybridizations with linear techniques, as well as the robust application of extreme learning machines, as a new paradigm of artificial neural networks. Experimental results on benchmark databases are competitive with state of the art approaches.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This introductory chapter is aimed to provide a quick overlook of the thesis. It provides the motivation and an overview of the Thesis contents in Section 1.1. Section 1.2 lists the contributions of the Thesis. 1.3 enumerates the publications achieved along the works. Section 1.4 describes the actual contents of the chapters in the Thesis.

1.1 Motivation

The works leading to the realization of this thesis have been a meandering path of collaborations, as can be quickly ascertained perusing through the list of publications. The main thread of these collaborations and publications have been the application of Machine Learning approaches to image processing, though some tangential works, such as reward prediction in Multi-Agent Reinforcement Learning or the modeling of subconscious social intelligence, have also ended in successful collaboration and publications. At the time of writing this thesis, the aim has been to focus on what we believe are the most relevant lines of work. Therefore, the thesis has a practical approach focused on contributing on the following Artificial Intelligence applications:

- Detection of the presence of humans on images, specifically hyperspectral images

- Biometrics: Human identification via face recognition using Machine Learning algorithms.

The works presented here lead to new computational methods to achieve better face recognition under dire circumstances and to enable person detection using hyperspectral imagery. From the point of view of computational innovations, the thesis follows the line of research of the Computational Intelligence Group on the development of Lattice Computing algorithms and applications, specifically for hyperspectral image unmixing and feature extraction in grayscale images. Another big research interest of the thesis has been the application of a relatively new approach to neural network training, that of Extreme Learning Machines.

1.1.1 Person detection

Detection of people is a practical need in many circumstances. A good example is survivor detection after natural disasters. In that kind of scenario, relying on normal photography and classical computer vision techniques may not be enough. The work presented here proposes the use of hyperspectral imagery to solve this task. However, there is a notable lack of publicly -even privately- available databases useful for this task. Therefore, we have laid and followed an experimental pipeline that starts collecting and preprocessing the data and ends proposing effective computational tools.

Appendix A explains the hyperspectral image capture process as well as the preprocessing steps leading to the collection of the experimental dataset used in this thesis. Then, the research forks into two main goals:

- Developing semi-supervised Active Learning techniques that allow us to label the data. These experiments simultaneously demonstrated that skin can in fact be segmented in hyperspectral images. We proposed a uncertainty measure to be used by an ensemble of classifiers in order to select the most interesting samples. This allowed to quickly and consistently segment the skin regions of the data.

- Studying endmember induction and unmixing techniques. We have throughly explored the state of the art endmember induction algorithms and compared them with the Lattice Computing methods. This works also proposes a endmember selection method that reduces the dimensionality of the output from Lattice Auto Associative Memories (LAAM).

These two lines of research allowed to conclude that we can perform skin detection on hyperspectral images. Moreover, lattice methods show equal if not better performance compared to classic algorithms.

1.1.2 Face recognition

Face recognition [22] is one of the most relevant applications of image analysis. The challenge to build an automated system which equals human ability to recognize faces has been a central case study in Artificial Intelligence since its inception.

Many industrial applications, most of them in the security field, require increasing system accuracy and robustness. Two main problems have been identified in face recognition: (1) face authentication, which is a binary classification problem, stated as providing confirmation that a user is who claims to be, and (2) face identification, which is a multiclass classification problem stated as the finding the identity of a user searching through a face image database corresponding to many potential user identities. This Thesis works focus on the face identification problem. This initial problem can be extended to gaze, expression or mood recognition [122]. Taken as pattern recognition problem, face recognition provides a perfect benchmarking framework to test feature extraction techniques and classifiers.

There are many challenges facing the classic yet unsolved problem of face recognition. The real life applications usually record face data under less that ideal circumstances. This challenge calls for versatile methods. This Thesis is focused on developing Lattice Computing techniques that no only are suitable for recognizing faces, but also can overcome the problems common in unbalanced databases. The computational techniques that make use of lattice algebra are not mainstream. They have been used mainly in three or four dimensional data, e.g hyperspectral or medical imagery. Therefore, the practical problem at hand has been tackled using Lattice Computing techniques and studying the fusion of Lattice and linear techniques.

Feature Extraction.

This Thesis has studied feature extraction techniques based on Lattice Computing. We have used Lattice Independent Component Analysis (LICA), and developed a linear-lattice feature fusion scheme that improves face recognition. We have studied the interplay between these linear and lattice feature extraction methods empirically, obtaining promising results. We present a fusion of feature extraction methods that greatly improves recognition.

Classification.

Another novel computational tool are Extreme Learning Machines (ELMs). They are fast and accurate Neural Network learning tools. For the first time, we have explored ELMs capabilties for face recognition, showing very good performance. Moreover, we have combined the Lattice Computing algorithms with ELMs, departing from classical face recognition approaches. This thesis explores the ELMs performance versus the state of the art algorithms like Random Forests and Support Vector Machines. We also compared the ELM learning approach with other methods like Back Propagation or Scaled Conjugate Gradient learning.

1.2 Contributions

This Thesis is very application-oriented. Therefore, the main contributions are practical:

- A new hyperspectral database. The scenes have subjects on them an were taken under diverse conditions.
- Extensive experimentation on person detection using hyperspectral imaging techniques.
- Extensive application of Lattice Computing to biometrics and hyperspectral imaging.
- Exhaustive study of face recognition under undesirable conditions.
- First time application of ELMs to face recognition.
- Exploration of fusion of Lattice Computing methods to linear features.

The contributions of the Thesis from a methodological point of view are the following ones:

- Provides a review of the state of the art in three research areas: active learning, hyperspectral image unmixing and face recognition algorithms.
- Provides a experimental methodology that has a common denominator: trying to learn things from unbalanced and noisy data. The experimental setting of this thesis is not ideal, and the contributions are consciously directed towards utilizing computational intelligence techniques under difficult undesirable circumstances.
- The presented results and methods are compared with well known benchmark databases and algorithms respectively. The metrics and parameters used to evaluate the results are thoroughly explained in the different chapters and in appendix B. We have tried to be as transparent as possible, and all the algorithms are freely available on-line.

From a computational point of view, this Thesis has the following contributions:

- A novel Active Learning scheme using Random Forests and uncertainty calculation that allows fast accurate semi-supervised image segmentation.
- A endmember selection methodology to reduce the output of the Lattice Computing endmember induction algorithm WM.

- Various fused feature extraction methods formed by LICA and linear methods.

- A robust feature extraction and classification scheme, using LICA along with ELMs, that provides great results when working with unbalanced databases.

1.3 Publications

1. Ion Marques, Manuel Graña, Anna Kaminska-Chuchmala, Bruno Apolloni, “An Experiment of Subconscious Intelligent Social Computing on household appliances”, Neurocomputing (accepted) 2014.

2. Borja Ayerdi, Ion Marqués and Manuel Graña, “Spatially regularized semisupervised Ensembles of Extreme Learning Machines for hyperspectral image segmentation”, Neurocomputing (in press) 2014.

3. Ion Marques, Manuel Graña, “Hybrid Sparse Linear and Lattice Method for Hyperspectral Image Unmixing”, Proceedings of HAIS 2014, Salamanca, Spain, Lecture Notes in Computer Science, vol 8480, pp 266-273, 2013.

4. Iñigo Barandiaran, Odei Maiz, Ion Marqués and Manuel Grana, “ELM for Retinal Vessel Classification”, Proceedings of ELM 2013, Beijing, China, Adaptation, Learning, and Optimization, vol 16, pp 135-143, 2013.

5. Borja Fernandez-Gauna, Ion Marqués, Manuel Graña, “Undesired State-Action Prediction in Multi-Agent Reinforcement Learning. Application to Multicomponent Robotic System control”, Information Sciences, vol 232, pp 309–324, 2013.

6. Ion Marqués, Manuel Graña, “Greedy sparsification WM algorithm for endmember induction in hyperspectral images”, Proceedings of the IWINAC 2013, Mallorca, Spain, Lecture Notes in Computer Science, vol 7931, pp 336-344, 2013.

7. Manuel Graña, Ion Marqués, Alexandre Savio, Bruno Apolloni, “A domestic application of Intelligent Social Computing: the SandS project”, Proceedings of the SOCO 2013, Salamanca, Spain, Advances in Intelligent Systems and Computing, vol 239, pp 221-228, 2013.

8. Ion Marqués, Manuel Graña, “Fusion of lattice independent and linear features improving face identification”, Neurocomputing, vol 114, pp 80-85, 2012.

9. Ion Marqués, Manuel Graña, “Image security and biometrics: A review”, Proceedings of the HAIS 2012, Salamanca, Spain, Lecture Notes in Computer Science, vol 7209, part II, pp 436-447, 2012.

10. Ion Marqués, Manuel Graña, “Face recognition with Lattice Independent Component Analysis and Extreme Learning Machines”, Soft Computing, vol 16, num 9, pp1525-1537, 2012.

11. Ion Marqués, Manuel Graña, “Experiments on Lattice Independent Component Analysis for Face Recognition”, Proceedings of the IWINAC 2011, La Palma, Spain, Lecture Notes in Computer Science, vol 6678, pp 286-294, 2011.

12. Ion Marqués, Manuel Graña, “Face processing for security: a short review”, Proceedings of the CISIS 2010, León, Spain, Advances in Intelligent and Soft Computing, vol 85, pp 89-96, 2010.

1.4 Structure of the Thesis

The contents of the Thesis are divided in two blocks. The first one is centered on hyperspectral imaging and person detection. The second block encompasses the works on face recognition. All the chapters are self-contained, each having an Introduction and Theoretical background, sections describing Experimental Design and Results, and a Conclusion. The chapters of the Thesis are organized as follows:

1. Chapter 2 reports a review on Active Learning and proposes a iterative image segmentation process applied to the task of partitioning skin regions on hyperspectral images.

2. Chapter 3 surveys the hyperspectral unmixing problem. It presents several endmember induction algorithms with the goal of characterizing skin pixels. It reports the application of several state of the art and Lattice Computing techniques.

3. Chapter 4 is centered on exploring different feature extraction methods for face recognition. Feature fusion is explored and a Lattice-linear fusion scheme is proposed.

4. Chapter 5 expands the reach of the previous chapter, exploring the capabilities of different algorithms to correctly recognize faces under undesirable conditions. It is demonstrated that a combination of Lattice-based feature extraction and non-iterative neural network learning shows the most promising results.

5. Appendix A presents the hyperspectral database developed for person detection. It explains the capture process and also describes the employed preprocessing techniques.

6. Appendix B offers more ample explanations of some methodological details from the experiments presented in chapters 2 and 3.

7. Appendix C gives a theoretical overview of Lattice Computing.

8. Appendix D reports the development of ELMs and explains the algorithm concisely.

Chapter 2

Skin Segmentation via Active Learning

Determining what hyperspectral image pixels belong to skin regions is a first necessary step in process of people detection in an image. In this Chapter this task is approached as an interactive segmentation problem. The motivation of the works reported in this Chapter is explained in Section 2.1. The computational methods used in the experiments are detailed in Section 2.2. Section 2.3 describes the particulars of the experimental design. The experimental results of this work are exposed in section 2.4. Finally, section 2.5 offers a concluding discussion on the matter at hand.

2.1 Introduction

Image segmentation is the process of partitioning visual data into meaningful pieces. It is one of the big challenges that remains open in computer vision. The ideal scenario, that in which the segmentation is done without human interaction, is practically unfeasible. There is not an objective function nor a measure of success that could be used to assess the performance of a segmentation algorithm. This means that there will be a human interaction step at some point. For once, human visual assessment of the data is necessary to add semantics to the image. That translates into manually developing a ground truth, or ideal segmentation of the data. Therefore, the performance of any method will be measured by comparing the segmentation results with that ideal segmentation created by the human. This approach is addressed in the literature as interactive image segmentation.

This supervised machine learning problem can be addressed using Active Learning. It is a useful tool when dealing with data containing scarce labeled samples, making the application of conventional learning techniques based on a static dataset unusable. The Active Learning approach, formally described in section 2.2.2, consists of two elements: a training algorithm and a query or method. The training algorithm is used to build classifiers from the small set of labeled data. The query method is used to select unlabeled which will be labeled by an oracle and added to the training set. This iterative process goes on until stopping criteria are met. The mentioned oracle can be a human expert, although the human input can be substituted by a computational method. Learning without reliable labeling information was proposed recently in [34]. They explored an scenario where the non-expert oracle was asked "whether a pair of instances belong to the same class".

In some cases data is received sequentially. Thus, the sample selection is done one by one with some threshold value. Some recent researches focus on *batch mode* Active Learning, where a batch of data points is simultaneously selected from an unlabeled set [16]. On the other hand, pool-based sampling occurs when a set of data is available and the samples must be chosen from that pool. There are two main querying strategies to choose the samples for training [35]. If a single learner is used the choice depend on the selected measuring strategy. This is called *querying by a single model*. Some Active Learning schemes use ensembles of classifiers and they apply a *query by committee* method: Each member of the committee presents a labeling hypotheses. There is a voting of the label of the candidates. The sample whose labeling decision shows the biggest disagreement within the committee will be queried and labeled by the oracle.

The learner or learners hypothesis can be built with different criteria, which were thoroughly reviewed in [121] and more recently in [35]. One family of methods is *uncertainty reduction*. On this approach the sample whose classification result shows the highest uncertainty is chosen. In the case of probabilistic model for classification, those samples with posterior probability nearest to that of a random sampling would be chosen. It can also be uses information entropy, the margin between the highest and the second highest probability for an instance, expected gradient length, variance reduction or least confidence. The problem with these methods is that they can easily include outliers in the learning process [117]. Avoiding this is the motivation behind other methods that take into account instance correlations . The *expected error reduction* approach tries to predict the future generalization error for all the samples, in order to choose the one that will lower the error the most. It is a computationally expensive approach. Other collection of methods that try to overcome the outliers problem are density-weighted strategies. The main idea is that new instances should not only be chosen considering uncertainty, but also how representative those samples are of the input distribution. Some classic approaches explore the feature correlations, using clustering techniques to pre-select the most representative data samples before the querying. Techniques for combining the representativeness and informativeness of samples have also been recently explored [66].

The methodology proposed on this chapter segments the image iteratively, requiring the input of the user on every step. The idea is to use a few labeled data points and segment the whole image with what we can learn from these points. Then, the user labels some more points considered difficult to segment. The process continues, augmenting the knowledge about the data in an interactive way while focusing on the difficult parts of the image. The advantages of this scheme is that it can be used to assist on the creation of the ground truth of the data. In our case, we previously labeled all the data manually. We have therefore assessed the validity of our method substituting the human interactor with an uncertainty-based query-by-commitee Active Learning setting. The algorithm uses an uncertainty measure as an indicator of which difficult pixels should it focus on the next iteration.

The need of these method rises notably when the data at hand is not labeled. It is the case of this work, as the hyperspectral images were collected in-situ and the ground truth, i.e a binary segmentation of skin and non-skin regions, is not available. Doing a very laborious manual segmentation, it was considered interesting to tests these interactive segmentation processes. Firstly, it effectively allows us to confirm if the methods are accurate. Secondly, it allows us to see the conflictive or difficult areas of the data. Finally, it can be a way of validating the manual segmentation. On the other hand, these first experiments also serve as a starting points towards answering the following question: It is feasible to to detect skin regions in hyperspectral images using the collected data, as described in appendix A, in a semi-supervised way?

2.2 Computational methods

The experimental design involves interactive active learning via classification and quantification of uncertainty. The basics of active learning, the selected classification algorithm and the measuring of uncertainty are explained in this section.

2.2.1 Random Forest Classifiers

Random Forest (RF) algorithm is a classifier ensemble [13] that encompasses bagging [12] and random decision forests [4, 57], being used in a variety of applications. RF captures complex interaction structures in data, and are proposed [13] to be resistant to both overfitting of data when individual trees are very deep and no pruned, and under-fitting when individual trees are too shallow. RF became popular due to its simplicity of training and tuning while offering a similar performance to boosting. Consider a RF as a collection of decision tree predictors

$$\{h(\mathbf{x},\psi_k);k=1,...,K\},$$

where ψ_t are independent identically distributed random vectors whose nature depends on their use in the tree construction, and each tree casts a unit vote to find the most popular class of input $\mathbf{x}$.

Given a dataset of N samples, a bootstrapped training dataset is used to grow a tree $h(\mathbf{x};\psi_k)$ on a randomly selected subset of instances with replacement from the original training sample. RF also employs random feature selection. At each node of the decision tree, $\hat{d}$ variables are selected at random $\hat{d} \ll d$. Each decision tree is grown using CART methodology without pruning. This tree growing approach recursively picks the best data split of each node with the criteria of how well separates the classes contained in the parent node.

The independent identically distributed random vectors ψ_t determine the random dimension selection and data sample bootstrapping prior to tree training, which are the source for individual tree diversity and uncorrelation of their outputs. This uncorrelation between the trees and the strength of the individual trees determine the generalization error of the forest.

The trained RF can be used for classification of a new input $\mathbf{x}$ by majority voting among the class prediction of the RF trees. Note that in RF the Law of Large Numbers insures convergence as $k \to \infty$, therefore avoiding overfitting.

2.2.2 Active Learning fundamentals

The performance of supervised classifiers strongly depend on the information provided by the data used to train the classifier, so that the appropriate selection and labeling of the training set may be a cumbersome task requiring extensive manual inspection and analysis of the data, typically requiring some visualization tool and labeling of each data sample. Besides, noisy samples may interfere the class statistics, which may lead to poor classification performances and/or over-fitting. For these reasons, a training set must be constructed in a smart way, meaning that it must consists of the minimal set of samples allowing to compute correctly the class boundaries, therefore it must contain the most informative data samples. In the machine learning literature this approach is known as Active Learning.

Active Learning [26, 132] focuses on the interaction between the user and the classifier. Let $X=\{\mathbf{x}_i,y_i\}_{i=1}^{l}$ be a training set consisting of labeled samples, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1,\ldots,N\}$. Let be $U=\{\mathbf{x}_i\}_{i=l+1}^{l+u} \in \mathbb{R}^d$ the *pool of candidates*, with $u \gg l$, corresponding to the set of unlabeled samples to be classified. The classifier would be feed with the samples $\mathbf{x}_i$ and the targets y_i in a classic classification scheme. Actively learning on the other hand involves trying to indicate which input vector should be selected from the training set, in order to improve the learning capabilities of the classifier. We can consider that, for a given learning task whose target function is f, there are some areas where the function is more easily learned and some that are more difficult to classify. Measuring how difficult a sample is to classify is not trivial, and is discussed below. The Active Learning approach tries to focus on those difficult to classify regions -similarly to boosting techniques- but interactively.

Generally, an Active Learning process can be summarized as follows: In a given iteration t, the Active Learning algorithm selects from the pool U^t the q candidates that will, at the same time, maximize the gain in performance and reduce the uncertainty of the classification model when added to the current training set X^t. The selected samples $S^t = \{\mathbf{x}_m\}_{m=1}^q \subset U$ are labeled with labels $\{y_m\}_{m=1}^q$ by an oracle, which can be a human operator in interactive segmentation, or the available ground truth when performing cross-validation experiments. Finally, the set S^t is added to the current training set $(X^{t+1} = X^t \cup S^t)$ and removed from the pool of candidates $(U^{t+1} = U^t \backslash S^t)$. The process is iterated until a stopping criterion is met, such as the achieved accuracy reaching a preset threshold θ_{max}.

2.2.3 Classification uncertainty in RF classifiers

RF classifiers allow a committee approach for the estimation of unlabeled sample uncertainty [132]: assume that we have built a committee of k base classifiers, i.e. a RF with k trees. The output of the committee members provide k labels for each candidate sample $\mathbf{x}_i \in U$. The data sample class label is provided by the majority voting. Our heuristic is that the standard deviation $\sigma(\mathbf{x}_i)$ of the class labels is the measure of the classification uncertainty of $\mathbf{x}_i$. Let us consider an ordering of the pool of candidates $U^* = \left\{\mathbf{x}_{j_i}\right\}_{i=l+1}^{l+u}$, where $\sigma(\mathbf{x}_{j_i}) > \sigma\left(\mathbf{x}_{j_{i+1}}\right)$. The *standard deviation query-by-bagging* heuristic selection of samples to be added to the train set is stated as the following selection:

$$S^t = \left\{\mathbf{x}_{j_m}\right\}_{m=1}^q \tag{2.1}$$

Standard deviation of predicted class labels is a natural multiclass heuristic measure of classification uncertainty. A candidate sample for which all the classifiers in the committee agree has a zero prediction standard deviation, thus its inclusion in the training set does not bring additional information. In other words, belongs to an “easy” to classify region of the data. On the contrary, a candidate with maximum disagreement between the classifiers results in maximum standard deviation, so it is “difficult” to classify it. Therefore, its inclusion will be highly beneficial.

2.3 Experimental design

The goal is to classify image pixels into two classes, the target region and the background. The Active Learning system returns to the user the unlabeled pixels whose classification outcome is most uncertain with the current classifier. After manual labeling by the user, pixels are included into the training set and the classifier is trained again. On our experiment, we first manually label all the images. We use this ground truth to simulate the manual input of the user. This shortcut avoids intermittent work-flow.

For each RF, we begun with 5 training samples. The RF consisted on $k = 100$ trees, sampling $\sqrt{d}$ variables as candidates at each split. We selected on each run $q = 20$ uncertain pixels candidates using the criterion defined above. The ideal situation is that in which the performance measuring statistics converge towards the optimum value when the number of active learning runs increases. We run each image up to 40 times to asses this principle. The selection of 40 maximum runs is justified by time constrains. Additionally, the tendency of our accuracy parameters is clear with just 40 runs, as can be seen in figures 2.1 and 2.2.

The experiments on active learning have some peculiarities which do not encourage the use of a cross validation scheme. There is a small initialization step, followed by an iterative growth of the training set, based on some criterion. Thus, we have performed 100 repetitions of each experiment and reported the mean and standard deviation values of CR, precision and sensitivity. The pipeline shown in figure 2.3 serves as a visual help to explain the conducted Active Learning experiments.

2.4 Experimental results

The 7 images with the 4 preprocessing schemes show high Correct Rate (CR), as shown in table 2.1. There are not noticeable changes from one coordinate system to the other. There are no big differences, due to the size disparity of background and skin regions. Variances, shown in parenthesis, are low for all images, except for image C5 reflectance with hyperspherical coordinates. Table 2.2 shows the precision (precision) of the learning process. Overall, cartesian coordinates seem to show the best capability of avoiding false positives. Image A1 seems to drop the worst results. It is noticeable that when using hyperspheric coordinates, image C5b drops significantly worse results when using reflectance. We can see that generally the precision for reflectance value in hyperspheric space are worse than their radiance counterparts. This difference does not appear in cartesian coordinates. Regarding Sensitivity, looking at table 2.3 it is clear that image A1 is the most difficult to segment. It also reinforces the conclusion that hyperspheric coordinates

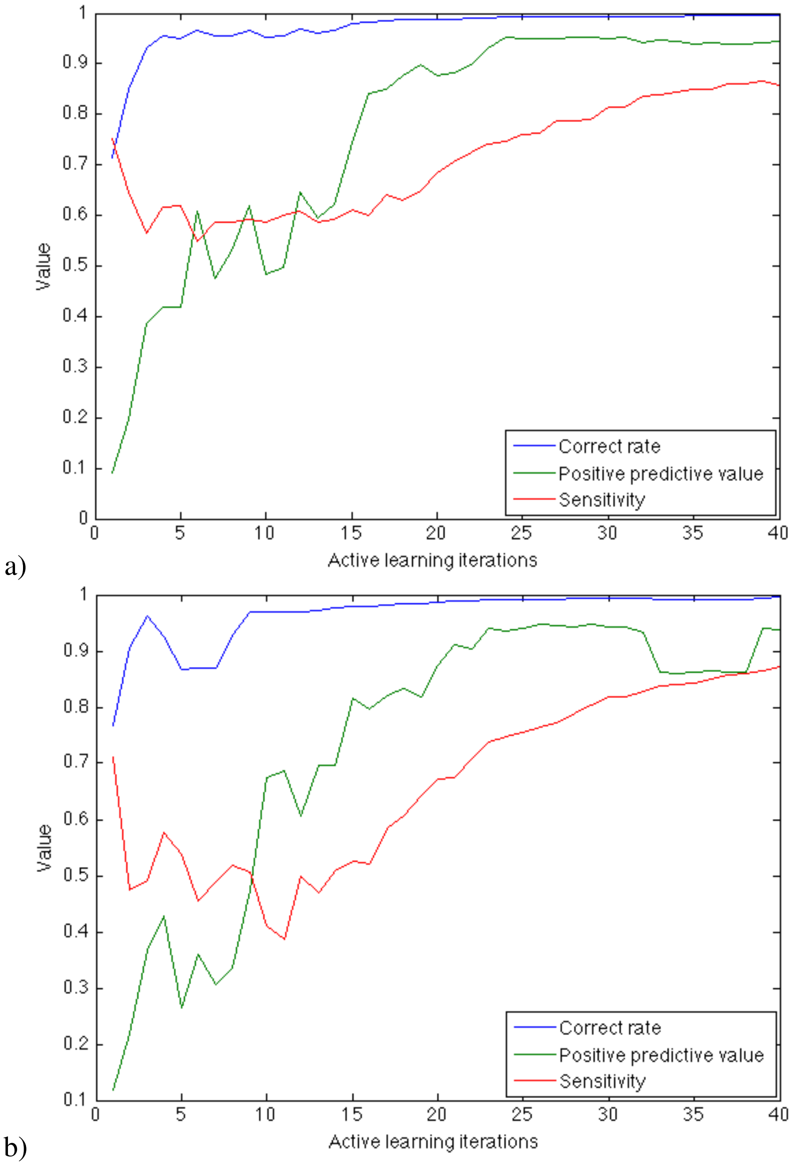

Figure 2.1: Performance indicator on increasing active learning iterations for a) radiance and b) reflectance values, averaged over the 7 images.

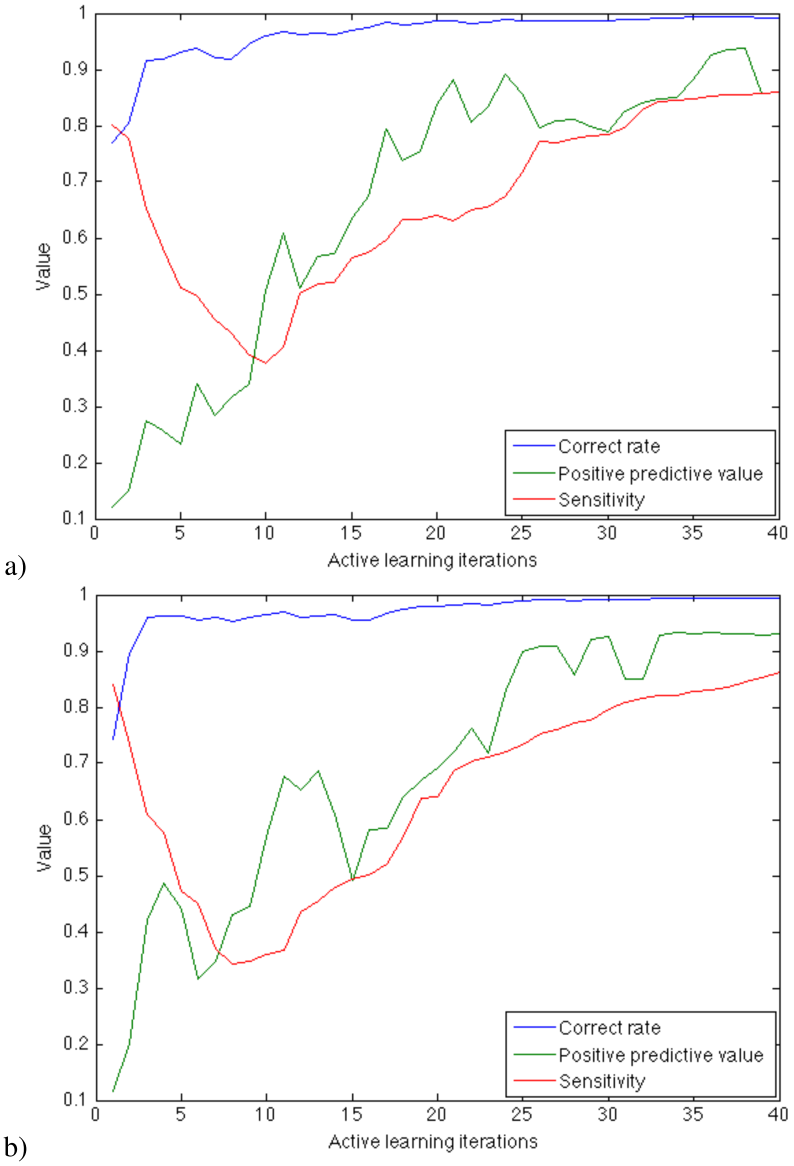

Figure 2.2: Performance indicator on increasing active learning iterations for a) radiance and b) reflectance values, averaged over the 7 smoothed images.

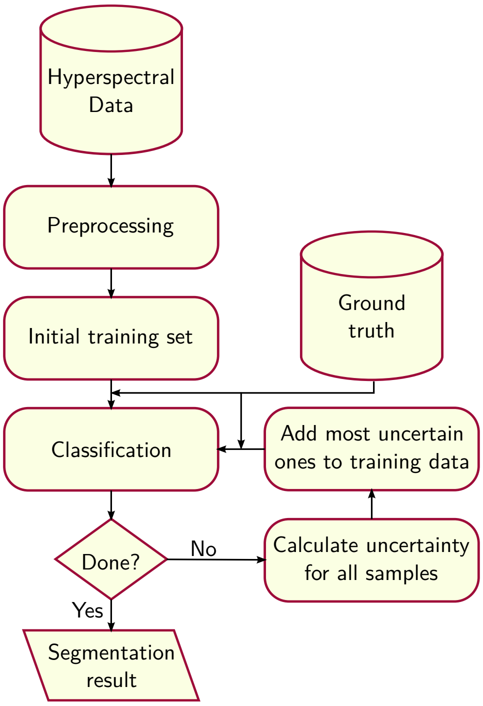

Figure 2.3: Active Learning experimental design flowchart. The preprocessing is detailed in appendix A.

		No smoothing		Smoothing	
		Radiance	Reflectance	Radiance	Reflectance
Cartesian coord.	A1	0.9890 (0.0007)	0.9888 (0.0010)	0.9888 (0.0009)	0.9882 (0.0011)
	B2	0.9891 (0.0015)	0.9890 (0.0027)	0.9919 (0.0008)	0.9922 (0.0004)
	A3	0.9949 (0.0013)	0.9953 (0.0011)	0.9965 (0.0004)	0.9963 (0.0005)
	C4	0.9977 (0.0024)	0.9985 (0.0001)	0.9984 (0.0003)	0.9983 (0.0004)
	C5	0.9935 (0.0025)	0.9948 (0.0005)	0.9961 (0.0014)	0.9965 (0.0005)
	C5b	0.9945 (0.0014)	0.9939 (0.0018)	0.9962 (0.0011)	0.9961 (0.0010)
	A5	0.9985 (0.0002)	0.9974 (0.0027)	0.9984 (0.0001)	0.9984 (0.0001)
	mean	0.9939 (0.0014)	0.9940 (0.0014)	0.9952 (0.0007)	0.9951 (0.0005)
Hyperspherical coord.	A1	0.9863 (0.0007)	0.9838 (0.0006)	0.9872 (0.0005)	0.9854 (0.0004)
	B2	0.9915 (0.0010)	0.9906 (0.0012)	0.9931 (0.0004)	0.9924 (0.0009)
	A3	0.9871 (0.0019)	0.9896 (0.0021)	0.9916 (0.0014)	0.9929 (0.0011)
	C4	0.9989 (0.0000)	0.9989 (0.0000)	0.9989 (0.0000)	0.9990 (0.0000)
	C5	0.9971 (0.0002)	0.9727 (0.0081)	0.9972 (0.0001)	0.9873 (0.0116)
	C5b	0.9970 (0.0001)	0.9964 (0.0002)	0.9971 (0.0001)	0.9965 (0.0003)
	A5	0.9980 (0.0002)	0.9970 (0.0003)	0.9980 (0.0001)	0.9974 (0.0002)
	mean	0.9937 (0.0006)	0.9899 (0.0018)	0.9947 (0.0004)	0.9930 (0.0021)

Table 2.1: Correct rate obtained by active learning for each image.

for reflectance values drop the performance of radiance data. Image A3 shows the biggest standard deviation, which indicates that for this image the initial random selection of training pixels is crucial.

Overall, the smoothing process has two effects: effectively enhancing the segmentation performance of the method and reducing the variance between experiment runs. Regarding precision on cartesian coordinates, it halves the standard deviation by halve while enhancing the average smoothing capabilities by 9.44%. Changes in sensitivity are not so noticeable. Results of data in hyperspheric coordinate are also generally better, but thee difference on standard deviation is not as relevant. There are some cases where the smoothing shows clear positive effects. For instance, it enhances precision and sensitivity values for image A3 while reducing the variability.

Figures 2.4 to 2.10 illustrate the segmentation results for cartesian and hyperspectral coordinates. Each image is selected from the best smoothing/not smoothing and reflectance/radiance results for each image as shown in tables 2.3 and 2.2. The ground truth is shown in black (background) and white (skin). Red pixels indicate true positives, while blue pixels denote false positives. We show the image corresponding to the method that dropped the best precision for each hyperspectral cube. Looking at individual images, we can see that A1 has the lower Sensitivity values in table 2.3. It can be observed in figure2.4 that there are noticeable white

		No smoothing		Smoothing	
		Radiance	Reflectance	Radiance	Reflectance
Cartesian coord.	A1	0.8402 (0.0418)	0.8401 (0.0585)	0.8248 (0.0510)	0.7948 (0.0634)
	B2	0.8832 (0.0346)	0.8839 (0.0500)	0.9396 (0.0196)	0.9471 (0.0116)
	A3	0.9295 (0.0128)	0.9299 (0.0113)	0.9419 (0.0160)	0.9347 (0.0177)
	C4	0.9349 (0.1036)	0.9657 (0.0043)	0.9706 (0.0025)	0.9663 (0.0074)
	C5	0.8621 (0.0546)	0.8904 (0.0131)	0.9304 (0.0368)	0.9398 (0.0131)
	C5b	0.8697 (0.0350)	0.8566 (0.0401)	0.9207 (0.0332)	0.9159 (0.0260)
	A5	0.9550 (0.0090)	0.9201 (0.0899)	0.9564 (0.0038)	0.9580 (0.0049)
	mean	0.8964 (0.0416)	0.8981 (0.0382)	0.9263 (0.0233)	0.9224 (0.0206)
Hyperspherical coord.	A1	0.7135 (0.0252)	0.7041 (0.0331)	0.7383 (0.0252)	0.7662 (0.0338)
	B2	0.9283 (0.0253)	0.9118 (0.0284)	0.9607 (0.0066)	0.9502 (0.0183)
	A3	0.7553 (0.0427)	0.8480 (0.0319)	0.8159 (0.0310)	0.8817 (0.0143)
	C4	0.9728 (0.0023)	0.9708 (0.0026)	0.9737 (0.0012)	0.9723 (0.0020)
	C5	0.9536 (0.0066)	0.5555 (0.0836)	0.9550 (0.0025)	0.7743 (0.1653)
	C5b	0.9307 (0.0026)	0.9241 (0.0039)	0.9355 (0.0020)	0.9256 (0.0098)
	A5	0.9351 (0.0081)	0.9121 (0.0109)	0.9385 (0.0046)	0.9286 (0.0086)
	mean	0.8842 (0.0161)	0.8323 (0.0278)	0.9025 (0.0104)	0.8856 (0.0360)

Table 2.2: Precision obtained by active learning.

		No smoothing		Smoothing	
		Radiance	Reflectance	Radiance	Reflectance
Cartesian coord.	A1	0.5706 (0.0210)	0.5658 (0.0267)	0.5769 (0.0133)	0.5752 (0.0169)
	B2	0.8655 (0.0106)	0.8641 (0.0051)	0.8702 (0.0099)	0.8703 (0.0129)
	A3	0.8984 (0.0461)	0.9120 (0.0338)	0.9414 (0.0115)	0.9439 (0.0077)
	C4	0.9349 (0.0195)	0.9405 (0.0084)	0.9243 (0.0200)	0.9242 (0.0284)
	C5	0.9569 (0.0021)	0.9584 (0.0015)	0.9521 (0.0054)	0.9522 (0.0037)
	C5b	0.9689 (0.0025)	0.9690 (0.0018)	0.9611 (0.0066)	0.9639 (0.0040)
	A5	0.9702 (0.0043)	0.9702 (0.0027)	0.9646 (0.0046)	0.9623 (0.0043)
	mean	0.8808 (0.0152)	0.8829 (0.0114)	0.8844 (0.0102)	0.8846 (0.0111)
Hyperspherical coord.	A1	0.5549 (0.0210)	0.3597 (0.0247)	0.5815 (0.0197)	0.4144 (0.0189)
	B2	0.8741 (0.0087)	0.8684 (0.0085)	0.8789 (0.0066)	0.8709 (0.0067)
	A3	0.8542 (0.0404)	0.7991 (0.0543)	0.9343 (0.0079)	0.8850 (0.0314)
	C4	0.9590 (0.0037)	0.9583 (0.0023)	0.9580 (0.0025)	0.9624 (0.0023)
	C5	0.9568 (0.0015)	0.9569 (0.0025)	0.9595 (0.0033)	0.9526 (0.0029)
	C5b	0.9750 (0.0016)	0.9638 (0.0037)	0.9739 (0.0016)	0.9664 (0.0027)
	A5	0.9677 (0.0029)	0.9457 (0.0066)	0.9663 (0.0020)	0.9446 (0.0078)
	mean	0.8774 (0.0114)	0.8360 (0.0147)	0.8932 (0.0062)	0.8566 (0.0104)

Table 2.3: Sensitivity obtained by active learning for each image.

areas. It is confirmed visually that A1 poses the biggest challenge in terms of skin segmentation. The scene was windy, which can move not only the vegetation but also the subjects clothes and hair. It can be observed that big areas of the arms are not correctly segmented. Refer to Appendix A for further details on the image. Regarding precision, we see that image C4 has the highest value, with a precision of 0.9706. This is illustrated by the lack of blue pixels. Notice that in all of the images, but more prominently in images C4, C5, C5b and A5, there are several blue pixels in areas that intuitively we would denominate as skin. This might be consequence of the human error induced in the manual segmentation process. Every human segmentation step involves the possibility of miss-labeling a sample, therefore dragging that error across the Active Learning iterations. Moreover, slight movements of the subjects during the image acquisition can also introduce spatial noise that leads to these erroneous labels.

2.5 Conclusion

The experimental framework presented in this Chapter enabled the exploration of diverse computational aspects when dealing with skin detection. Firstly, it was shown that it is possible to segment skin in hyperspectral images, even in situations where noise is very present. Secondly, an Active Learning methodology was proposed to label and segment the images. Lastly, the accuracy of the proposed system was tested with varying image preprocessing steps. Overall, the results can be summarized as follows:

- The calculation of reflectance normalization using white standards present on the scene adds noise and lowers the segmentation capability of the method.

- Representing the images in cartesian space is better suited than converting the data to hyperspherical coordinates.

- The smoothing of data enhances the segmentation results.

- The segmentation process is not robust to motion noise, such as appears in image A1 due to the wind conditions.

Therefore, obtaining a good skin segmentation depends on choosing a good image representation, preprocessing and computational techniques. However, results show that many errors are located in skin areas bordering non skin regions. This phenomenon can be partially caused by the manual segmentation step, where due to chromatic similarities it is difficult to asses whether one pixel is skin or not.

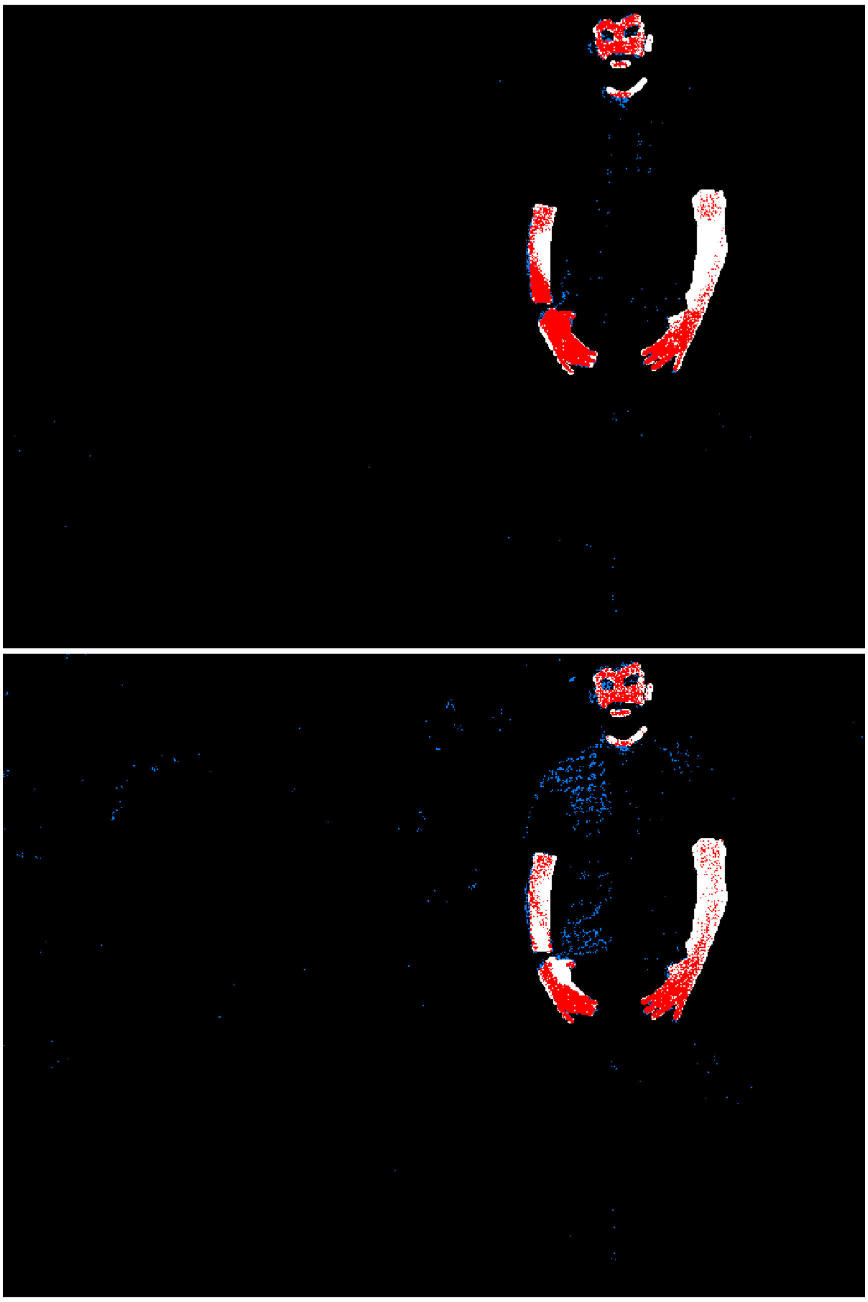

Figure 2.4: A1 active learning segmentation results for cartesian (top) and hyperspherical (bottom) coordinates.

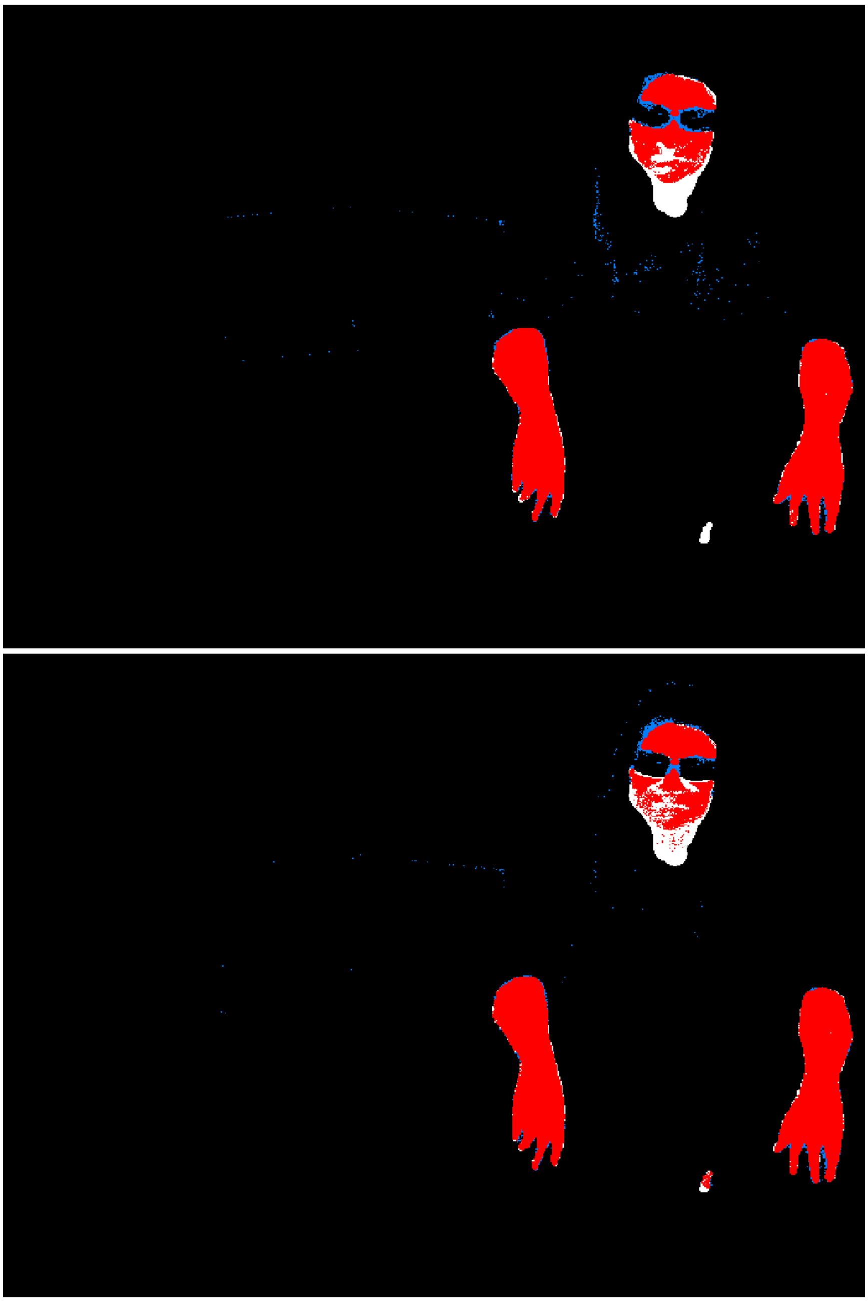

Figure 2.5: B2 active learning segmentation results for cartesian (top) and hyperspherical (bottom) coordinates.

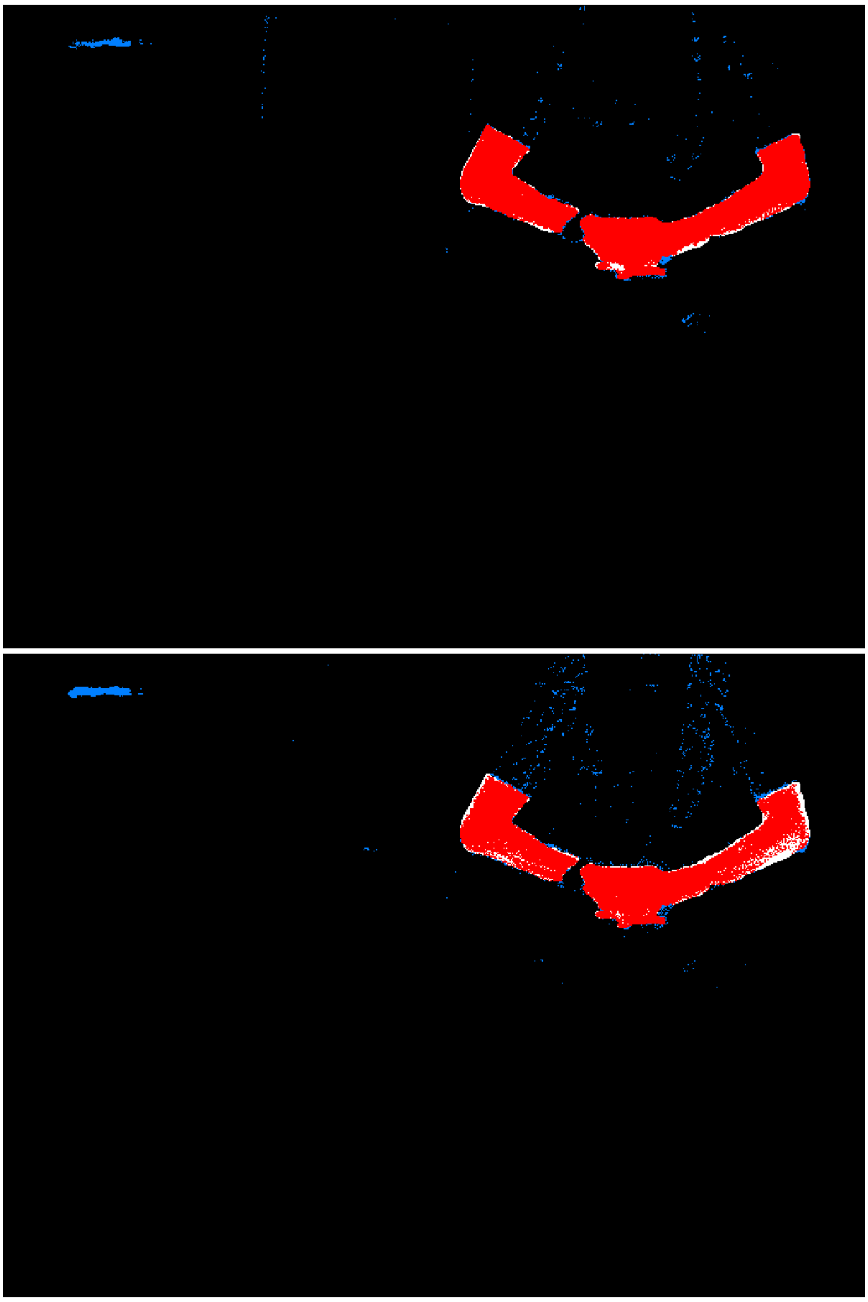

Figure 2.6: A3 active learning segmentation results for cartesian (top) and hyperspherical (bottom) coordinates.

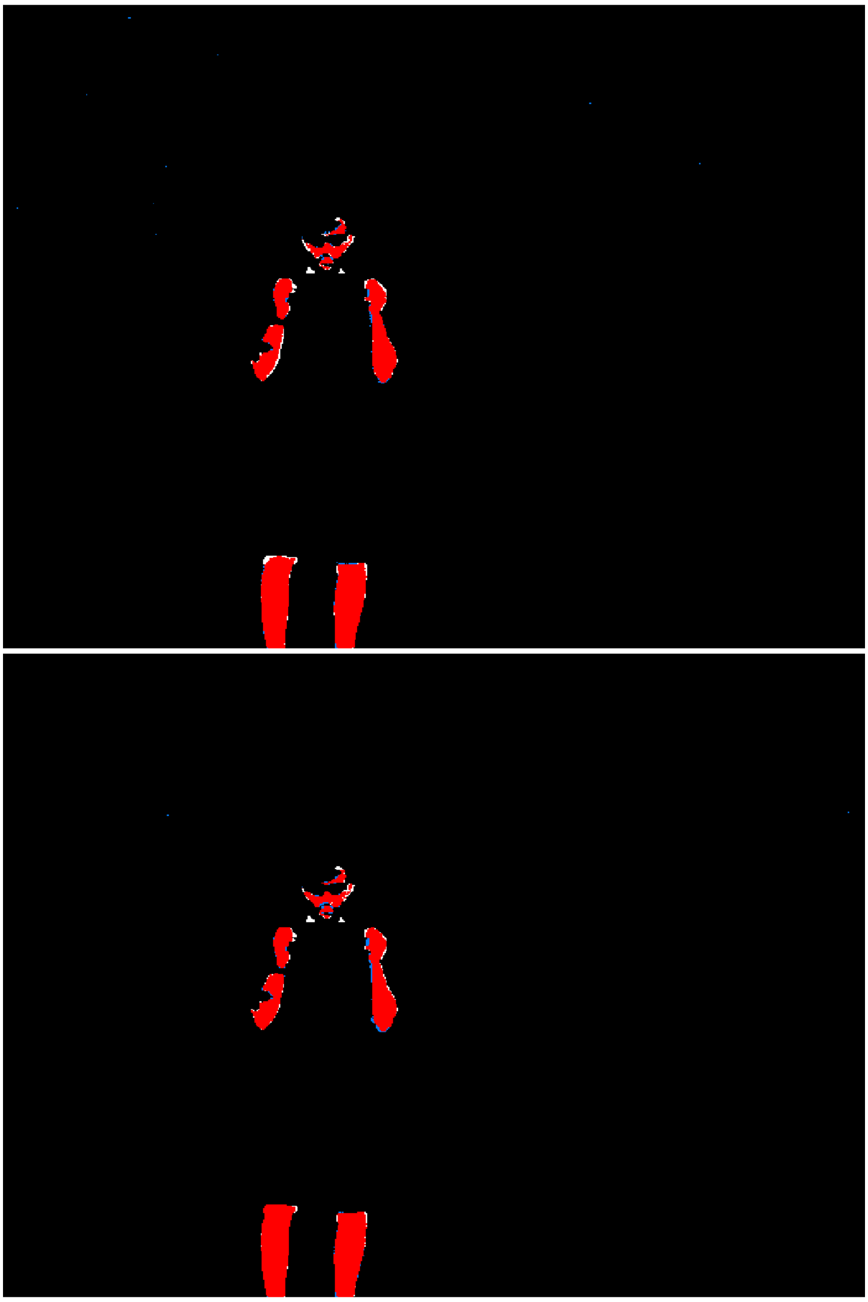

Figure 2.7: C4 active learning segmentation results for cartesian (top) and hyperspherical (bottom) coordinates.

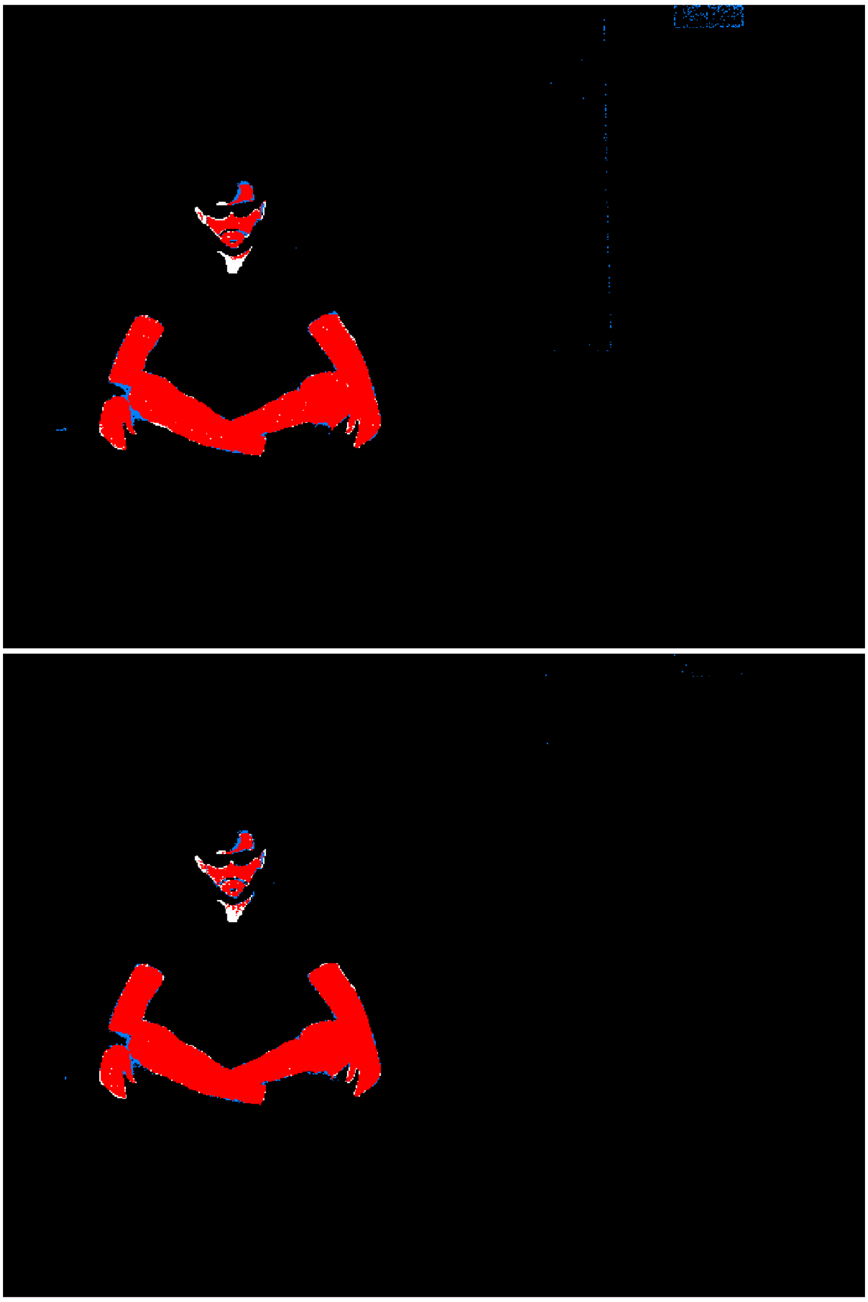

Figure 2.8: C5 active learning segmentation results for cartesian (top) and hyperspherical (bottom) coordinates.

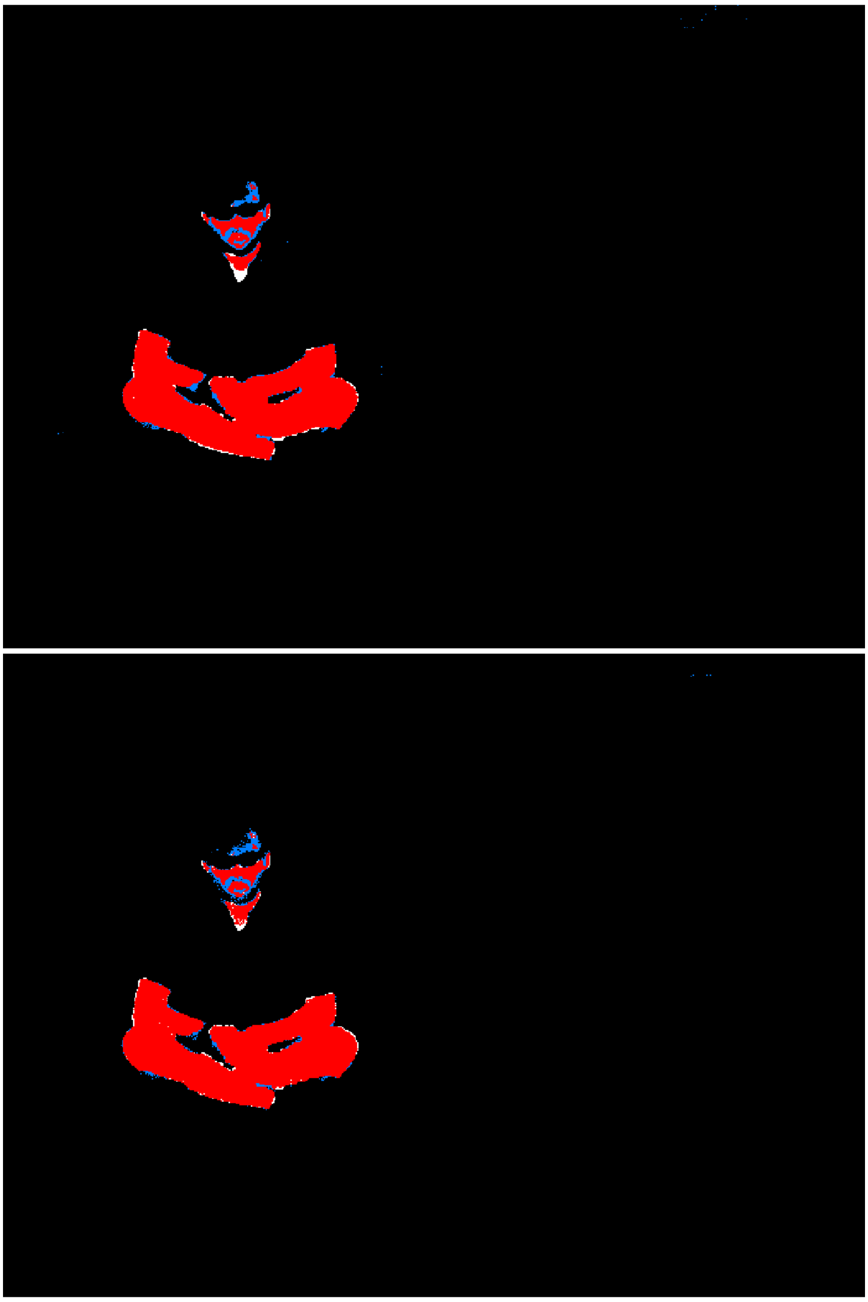

Figure 2.9: C5b active learning segmentation results for cartesian (top) and hyperspherical (bottom) coordinates.

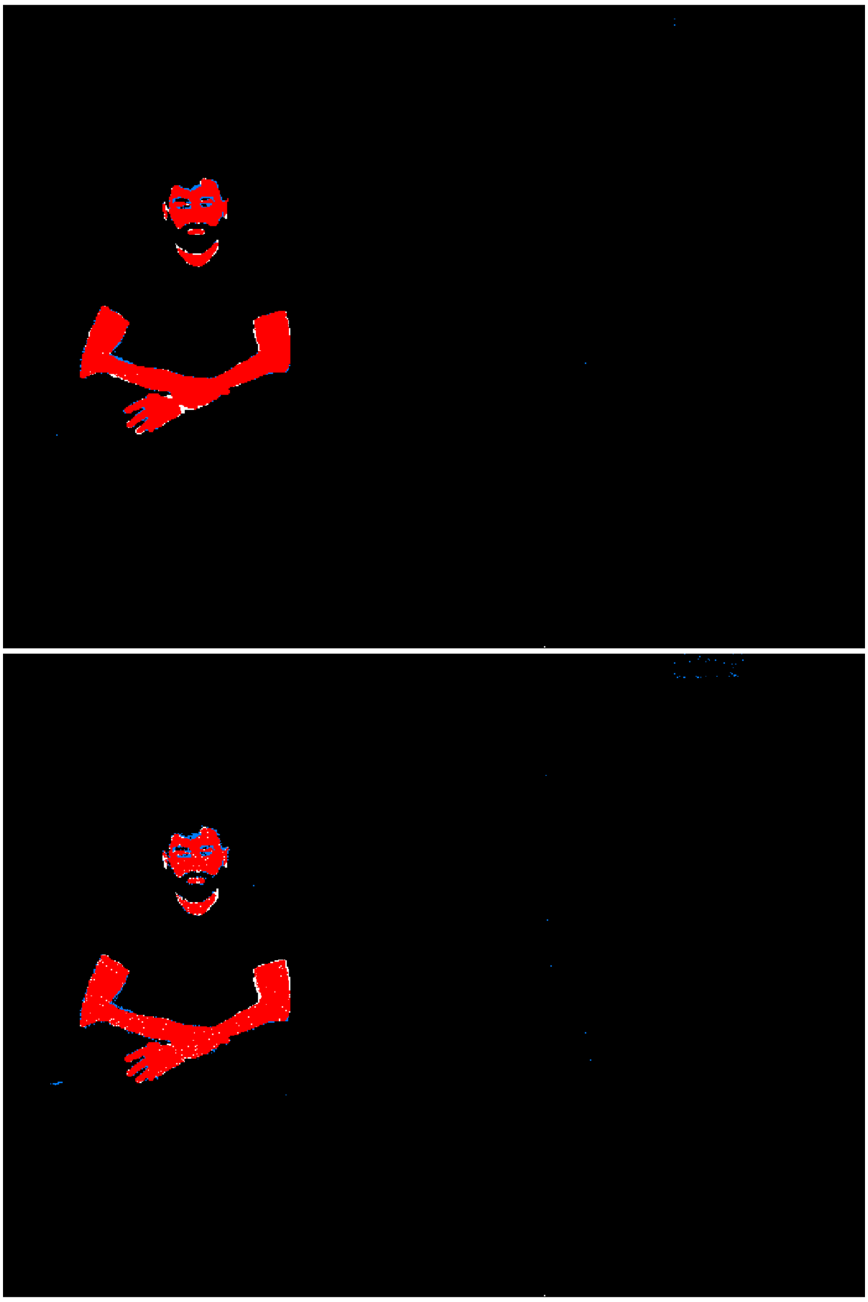

Figure 2.10: A5 active learning segmentation results for cartesian (top) and hyperspherical (bottom) coordinates.

Chapter 3

Skin Endmember Induction and Spectral Unmixing

The typical unsupervised hyperspectral scenario involves two steps: a) Inducing a set of endmembers **E** from the hyperspectral image and b) estimating the fractional abundances α. This subject is introduced in section 3.1 and a brief state of the art is provided. The details on the unmixing process as a whole and the calculation of the abundances are presented in section 3.2. The set of algorithms used to induce endmembers is detailed in section 3.3. The conclusions are reported in section 3.6.

3.1 Introduction

The analysis of hyperspectral images usually involves several computational techniques. There is a non trivial set of steps that cover the work-flow from the obtention of the raw data to the moment were the information is ready to be worked upon. These steps usually involve sensor calibration, radiometric correction, data geo-registration and atmospheric correction for remote-sensing images and preparing the correspondent metadata for each image.

There are some instances where the first step of processing the prepared data is Dimensionality Reduction (DR). DR tries to find a low-dimensional representation of the hyperspectral data. This representation should reduce the computational load requirements for the data. Moreover, the DR process should enhance the result of the following processing steps. It is desirable that at least it does not impair them. Signal Subspace Identification (SSI) [3] is the standard way of reducing dimensionality. It can be performed reducing the spatial or spectral resolution.

The first approach -Space transformation SSI- can be attained by such well known algorithms like Principal Component Analysis (PCA), Maximum Noise Fractions (MNF) or Singular Value Decomposition (SVD). This second order statistic methods may not be suitable, as hyperspectral images contain many subtel materials with sizes smaller than a pixel [7]. Other hyperspectral DR methods involve Independent Component Analysis (ICA) [137], Progressive dimensionality reduction by transform (PRDT) [20] and manifold and tensor based techniques [5, 6]. A limitation of these linear techniques is that they can not discover the nonlinear structure of the input data. Therefore, some recent approaches address the problem of modeling the nonlinear data structure of the underlying data manifold. Some of these methods use isometric feature mapping (ISOMAP) [129], locally linear embedding (LLE)[116, 33], or novel distance measures to deal with nonlinearity[100]. Other techniques make use of *a-priori* information about the data. Recent semi-supervised methods include using Local Scaling Cut Criterion (LCU) [148] or dual-geometric subspace projection (DGSP) [142].

The second family of DR algorithms are Band Selection SSI. Recent developments focus on signal quality, like the minimum noise band selection (MNBS) method proposed in [125]. A volume-gradient-based band selection (VGBS) method is proposed in [37]. Algorithms based on sparsity measures are also proposed, using linear regression model with L1 regularization (LASSO model) [124] or evolutionary strategies to select band matched to Multitask Sparsity Pursuit (MTSP) to evaluate selection performance [143].

The task after preprocessing the data is to induce the underlying endmembers of the data, either selecting some image pixel spectra as the best approximation to the endmembers or computing estimations of the endmembers on the basis of transformations of the image data. Geometric methods search for the vertexes of a convex polytope that covers the image data. One of the most popular methods is N-FINDR, proposed in [140]. Other more recent algorithm is orthogonal subspace projection (OSP)-based automatic target generation process (ATGP) [97]. Both of them have been tested in this work, and they are detailed in section 3.3. Other approach is to use lattice computing [110, 41, 108, 42], where a connection between linear mixing model algebraic properties and lattice independence is established. Some of these algorithms have been used for the experiments presented in this work. They are detailed in section 3.3. There are methods that do not use the strict theoretical definition of endmembers. The most popular among these heuristic methods is is the Pixel Purity Index (PPI) algorithm introduced in [71]. The Fast Iterative PPI algorithm (FIPPI) improves PPI in several aspects, it is described in section 3.3.

Recent endmember extraction techniques focus on one of these concepts: Introducing genetic algorithms [149, 45], using Ant Colony Optimization (ACO) methods[146, 145] or improving existing simplex-based methods [17, 138]. The integration of spatial and spectral information in unmixing has also seen considerable attention, and a comprehensive survey on the subject has been published in

[123]. Hybrid approaches where endmember induction and spectral unmixing are performed simultaneously are also being developed [31, 144, 32].

Once the endmembers are selected the next step is Spectral Unmixing. The idea behind spectral unmixing is that a single pixel is a mixture of one or more endmember spectra corresponding to the aggregation of materials in the scene due to reduced sensor spatial resolution. The section 3.2 details the linear mixing unmixing (LSU) that has been used in this work. It is the most used and well researched approach. There is a recent trend on using Nonnegative Matrix Factorization (NMF) [78] to solve the linear unmixing problem. However, there are other unmixing approaches. Non-linear spectral unmixing (NLSU) has attracted increasing attention in recent years. An algorithm based upon a combination of a data description in terms of (approximate) geodesic distances was proposed in [55] and later expanded to obey obeys the positivity and sum-to-one constraints [56]. Other approach interprets a single pixel as both a mixture of endmember spectra and nonlinear variations in reflectance, and a joint mixture resulting from the linearity and nonlinearity in hyperspectral data; transforming the unmixing problem into a Constrained Nonlinear Least Squares Algorithm (CNLS) [99].

Semi-supervised unmixing has also received significant attention lately. The idea behind this approach is that observed image signatures can be expressed in the form of linear combinations of known spectral collections or libraries. This combinatorial problem can be efficiently faced with sparse regression (SR) techniques [70]. The classic SR techniques like like Orthogonal Matching Pursuit (OMP), Basis Pursuit (BP) or iterative spectral mixture analysis (ISMA) are thoroughly explored in [70]. More recent techniques include sparse NMF [82], constrained $lp - l_2$ optimization [23] or a hierarchical Bayesian approach [130]. Although reformulating the unmixing problem as a SR case was motivated by semi-supervised learning, in [85] we attempt to transfer the approach to unsupervised unmixing.

3.2 Spectral unmixing

Recent developments in spectral unmixing of hyperspectral images represent a step forward in the progress of compound analysis of remote sensing images. The experimental setting on this chapter aims to apply those techniques to the skin detection problem. We intuitively propose that one or more endmembers should distinctively allow us to differentiate between skin and non skin regions. The goal is to extract the endmembers of the images and identify those pertaining to skin regions. The advantage of succeeding in this endeavor is threefold: We asses the suitability of the state of the art methods for endmember extraction, we develop unmixing pipelines able to successfully select interesting endmembers and finally we identify endmembers that can be used to detect skin in different conditions.

3.2.1 Linear unmixing model

We translate this intuitive idea into the Linear Mixing Model (LMM) [75]. It states that, given a hyperspectral image $\mathbf{H}$, whose pixels are vectors in L-dimensional space, it spectral signature is characterized by a set of endmembers, $\mathbf{E} = \left\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_q\right\}$. The spatial-spectral characterization is a tuple $(\mathbf{E}, \alpha)$, where , α is an $q \times 1$ vector of fractional abundances resulting from the unmixing process. For each pixel, the linear model is written as

$$\mathbf{x} = \mathbf{E}\alpha + \mathbf{n} \tag{3.1}$$

where $\mathbf{x}$ is a is a $L \times 1$ column vector of measured reflectance values and n represents the noise affecting each band.

Under a linear mixture model framework, as seen in 3.1, the q dimensional abundance of a pixel $\mathbf{x}$ can be estimated by solving a least squares problem with no constrains:

$$\min_{x} \|\mathbf{E}\alpha - \mathbf{x}\|^2, \tag{3.2}$$

which has a can be analytically approximated by solving the equation system

$$\alpha = \mathbf{E}^{\dagger}\mathbf{x}, \tag{3.3}$$

where $\mathbf{E}^{\dagger}$ indicates the pseudo-inverse of the endmember matrix $\mathbf{E}$, which can be calculated as $\mathbf{E}^{\dagger} = (\mathbf{E}^T\mathbf{E})^{-1}\mathbf{E}^T$. Using unconstrained abundance estimation has the advantages of implementation simplicity and fast execution time. However, this model carries the disadvantage that lacks physical meaning. It allows the occurrence of negative abundances, and it makes no sense to have a negative quantity of a given element in a pixel. It can also happen that the sum of abundances in a given pixel is not unitary. One would think that the sum of the abundances on a pixel would be 1. Two constrained are usually imposed in order to impose the aforementioned physical soundness. These are the abundance nonnegativity constraint (ANC) and abundance sum-to-one constraint (ASC), respectively defined as

$$\alpha_i \geq 0,\ \forall \mathrm{i} = 1, \ldots, \mathrm{q} \tag{3.4}$$

$$\sum_{i=1}^{q} \alpha_i = 1. \tag{3.5}$$

A least squares problem constrained by both ANC and ASC is called Fully Constrained Least Squares (FCLS) abundance estimation method. Given the image endmembers, enforcing the ANC and ASC conditions on Spectral Unmixing requires solving the so-called FCLS problem. It can be a very computationally expensive process. The Nonnegativity Constrained Least Squares (NCLS) approach is a less constrained approach than the classic FCLS [52, 53] in that it only applies the ANC constraint. It can be argued that many factors can add noise to the data on the acquisition process. Consequently, although the ANC constrain wold still be necessary, imposing ASC loses its physical significance. The experiments presented in this chapter solve the NCLS problem via the classic method described in [77].

3.2.2 Sparse unmixing model

Having a large set of endmembers, unmixing is equivalent to finding an optimal subset of signatures that can best model each mixed pixel in the scene. This can be understood as a combinatorial problem where the presence of an endmember in a pixel is very small compared to the dimensionality of the data and size of the endmember candidate pool. Consequently, hyperspectral unmixing can be reformulated as a sparse approximation problem. Sparse unmixing of hyperspectral data is a very active research area[70].

The sparse signal approximation problem can be summarized as follows: Let have a data matrix $\mathbf{X}$. We define a matrix $\Phi \in \mathbb{R}^{q\times L}$ called the dictionary. The q columns of Φ are referred as atoms. In this work, a set of endmembers set of endmembers, $\mathbf{E} = \left\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_q\right\}$ will be the dictionary Φ. Therefore, each of the q induced endmembers corresponds to one atom of the dictionary. The problem is to find a mixing matrix $\mathbf{Y}$ so that

$$\mathbf{X} = \Phi\mathbf{Y} + \varepsilon, \tag{3.6}$$

where matrix $\mathbf{Y}$ optimizes certain sparsity measure. This matrix Y is in fact the collection of abundance images obtained by the unmixing process.

One of many methods to achieve this sparsification is to use Conjugate Gradient Pursuit [11]. The conjugate gradient method is a popular directional optimization method. This method calculates a similar decomposition as the QR factorization; and it's guaranteed to solve quadratic problems in as many steps as the dimension of the problem. The conjugate gradient method, used as a directional pursuit method, is explained in algorithm 1. For clarity, we denote $\mathbf{y}$ the set of elements that compose matrix $\mathbf{Y}$.

Algorithm 1 Pseudo-code specification of the Conjugate Gradient Pursuit algorithm.

```
1. $\mathbf{r}^0 = X$ is the initial residual error. $\Gamma^0 = \emptyset$ is an index set. $\mathbf{y}^0_{\Gamma^0} = 0$ is the initial set of output sparse vectors. $b_0 = 1$ is a term needed to calculate new conjugate gradients.

2. For $i = 1,2,3,...$ util stopping criterion is met:

    (a) Calculate gradient $\mathbf{g}$ for $\mathbf{y}$ restricted to $\Gamma^i$:

            $\mathbf{g}_{\Gamma^i} = \Phi^T_{\Gamma^i}\left(X - \Phi_{\Gamma^i}\mathbf{y}^{i-1}_{\Gamma^i}\right).$

    (b) Select the best element index:

            $\gamma^i = \arg_j \max|\mathbf{g}_{\Gamma^i}|.$

    (c) Update the index set:
            $\Gamma^i = \Gamma^{i-1} \cup \gamma^i.$

    (d) Calculate the gram matrix $\mathbf{G}_{\Gamma^i}$ :

            $\mathbf{G}_{\Gamma^i} = \Phi^T_{\Gamma^i}\Phi_{\Gamma^i}.$

    (e) We denote $\mathbf{D}_{\Gamma^i}$ the matrix containing all conjugate update directions from iteration $i-1$, with an additional row all zeros. We calculate the update direction $\mathbf{d}_{\Gamma^i}$:

            $\mathbf{b} = \left(\mathbf{D}^T_{\Gamma^i}\mathbf{G}_{\Gamma^i}\mathbf{D}_{\Gamma^i}\right)^{-1}\left(\mathbf{G}^T_{\Gamma^i}\mathbf{D}_{\Gamma^i}\mathbf{g}_{\Gamma^i}\right),$

            $\mathbf{d}_{\Gamma^i} = b_0\mathbf{g}_{\Gamma^i} + \mathbf{D}_{\Gamma^i}\mathbf{b}.$

    (f) Calculate new set of vectors $\mathbf{y}^i_{\Gamma^i}$:

            $\mathbf{c}^i = \Phi_{\Gamma^i}\mathbf{d}_{\Gamma^i},$

            $a^i = \frac{\left\langle \mathbf{r}^i, \mathbf{c}^i \right\rangle}{||\mathbf{c}^i||_2^2},$

            $\mathbf{y}^i_{\Gamma^i} = \mathbf{y}^{i-1}_{\Gamma^i} + a^i\mathbf{d}_{\Gamma^i}.$

    (g) Calculate new residual error $\mathbf{r}^i$:

            $\mathbf{r}^i = \mathbf{r}^{i-1} - a^i\mathbf{c}^i.$

3. Output $\mathbf{r}$ and $\mathbf{y}$.
```

3.3 Endmember Induction Algorithms

Several endmember induction algorithms (EIAs) have been used. Some of them well known and widely used in the literature, like N-FINDR endmembers induction algorithm [140] and Fast Iterative Pixel Purity Index (FIPPI) endmembers induction algorithm [19]. ATGP endmembers induction algorithm, presented in [97], was also used. Two methods based on Lattice Computing, developed by the GIC research group, have been also applied -Incremental lattice Source Induction Algorithm (ILSIA) endmembers induction algorithm [44] and Endmember Induction Heuristic Algorithm (EIHA) endmembers induction algorithm [42]. WM endmembers induction algorithm has also been tested [114], and a sparsified WM (sWM) algorithm is proposed.

3.3.1 N-FINDR

N-FINDR is a geometric algorithm exploiting the fact that equation 3.1 defines a convex set. The data is considered to form a simplex where each vertex represents the spectral signature of a pure endmember. These M vertexes or endmembers are extracted in the following manner:

1. The data is projected down to an $M-1$ dimensional subspace using PCA.

2. An initial randomly selected set of $M+1$ pixels is chosen from the data. The formula for the volume of simplexes from all combinations of M pixels from this set is as follows:

$$V(\mathbf{E}) = \frac{|\det(\mathbf{E})|}{(M-1)!}, \tag{3.7}$$

   where the augmented matrix of endmembers is:

$$\mathbf{E} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_M \end{bmatrix}. \tag{3.8}$$

   $\mathbf{e}_i$ is the column vector containing the bands of endmember i.

3. N-FINDR searches (non-exhaustively) for the largest simplex that can be constructed within the data, by “inflating” the simplex inside the data. This means getting the maximum determinant value.

3.3.2 ATGP

The ATGP algorithm generates target pixels via orthogonal subspace projections (OSP). The process of obtaining M endmembers is as follows:

1. The pixel with maximum energy is selected as the initial endmember $\mathbf{e}_1$.
2. Until a set of target pixels $\mathbf{e}_1, \ldots, \mathbf{e}_M$ is extracted, we repeat the following process to iteratively extract each $\mathbf{e}_i$:
   (a) Calculate the OSP $P_U^{\perp} = \mathbf{I} - \mathbf{U}\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ with $\mathbf{U} = [\mathbf{e}_1\ \mathbf{e}_2 \ldots \mathbf{e}_{i-1}]$ to every pixel of the data.
   (b) $\mathbf{e}_i$ will be the target signature that has maximum orthogonal projection in $\langle \mathbf{e}_1\ \mathbf{e}_2 \ldots \mathbf{e}_{i-1} \rangle^{\perp}$

3.3.3 FIPPI

Standard PPI suffers from computational complexity problems. The algorithm proposed in [19] overcomes this weakness, and more importantly, is fully unsupervised.These M endmembers are extracted as follows:

1. The data is projected down to an M dimensional subspace using PCA.
2. Let $\left\{\mathbf{skewer}_{\mathrm{j}}^{(0)}\right\}_{j=1}^{M}$ be the set containing those target pixels generated by ATGP, as described in the preceding subsection.
3. At iteration k we calculate $\mathbf{skewer}_{\mathrm{j}}^{(\mathrm{k})}$, as follows:
   (a) All pixels are projected into $\mathbf{skewer}_{\mathrm{j}}^{(\mathrm{k})}$, to find those in extreme positions, to form an extrema set $\mathrm{S}_{\mathrm{extr}}\left(\mathbf{skewer}_{\mathrm{j}}^{(\mathrm{k})}\right)$.
   (b) Find pixels that produce largest $N_{PPI}\left(\mathbf{x}_j^{(k)}\right)$, defined by

$$\begin{array}{c} N_{PPI}(\mathbf{x}) = \sum_j I_{S_{extr}\left(\mathbf{skewer}_j^{(k)}\right)} \\ I_S = \left\{ \begin{array}{ll} 1, & if\, \mathbf{x} \in S \\ 0, & f\, \mathbf{x} \notin S \end{array} \right. , \end{array} \tag{3.9}$$

   and let them be denoted $\left\{\mathbf{x}_j^{(k)}\right\}$.
   (c) Having$\left\{\mathbf{skewer}_{\mathrm{j}}^{(\mathrm{k+1})}\right\} = \left\{\mathbf{skewer}_{\mathrm{j}}^{(\mathrm{k})}\right\} \cup \left\{\mathbf{x}_j^{(k)}\right\}_{N_{PPI}\left(\mathbf{x}_j^{(k)}\right)>0}$, if $\left\{\mathbf{skewer}_{\mathrm{j}}^{(\mathrm{k})}\right\} = \left\{\mathbf{skewer}_{\mathrm{j}}^{(\mathrm{k+1})}\right\}$ then we stop adding endmembers to the skewer set.

4. Pixels with $N_{PPI}\left(\mathbf{x}_j^{(k+1)}\right) > 0$ are the desired endmembers.

3.3.4 EIHA

EIHA [42] is a heuristic algorithm . It has a gain parameter α which has impact on the number of endmembers found. Low values imply large number of endmembers. The method is detailed in algorithm 2 -notice that $\overrightarrow{\sigma}$ is the standard variation of a pixel $\mathbf{x}_i$, which corresponds to the additive noise of the spectra.

Algorithm 2 Endmember Induction Heuristic Algorithm (EIHA)

1. Center the data $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{N}\}$.
2. Initialize the set of vertices $\mathbf{E} = \{\mathbf{e}_1\}$ with a randomly picked sample.
3. Construct $\mathbf{M}_{EE}$ and $\mathbf{W}_{EE}$.
4. For each pixel $\mathbf{x}_i$
    (a) compute the noise corrections sign vectors$\mathbf{x}_i^+ = \left(\mathbf{x}_i^c + \alpha\overrightarrow{\sigma} > \mathbf{0}\right)$ and $\mathbf{x}_i^- = \left(\mathbf{x}_i^c - \alpha\overrightarrow{\sigma} > \mathbf{0}\right)$
    (b) compute $y^+ = M_{EE} \boxed{\triangle} \mathbf{x}_i^+$
    (c) compute $y^- = W_{EE} \boxed{\triangledown} \mathbf{x}_i^-$
    (d) if $y^+ \notin \mathbf{X}$ or $y^- \notin \mathbf{X}$ then $\mathbf{x}_i^c$ is a new vertex to be added to $\mathbf{E}$, execute once 3 with the new $\mathbf{E}$ and resume the exploration of the data sample.
    (e) if $y^+ \in \mathbf{X}$ and $\mathbf{x}_i^c > \mathbf{e}_{y^+}$ the pixel spectral signature is more extreme than the stored vertex, then substitute $\mathbf{e}_{y^+}$ with $\mathbf{x}_i^c$.
    (f) if $y^- \in \mathbf{X}$ and $\mathbf{x}_i^c < \mathbf{e}_{y^-}$ the new data point is more extreme than the stored vertex, then substitute $\mathbf{e}_{y^-}$ with $\mathbf{x}_i^c$.
5. The final set of endmembers is the set of original data vectors $\mathbf{x}_i$ corresponding to the sign vectors selected as members of $\mathbf{E}$.

3.3.5 ILSIA

The objective of the algorithm is to extract a set of SLI vectors from the input dataset. If SLI was the only criteria to be tested to include input vectors in the set of lattice sources, then a large number of lattice sources would be detected. That being the case, there would be little significance of the abundance coefficients because many of them will be closely placed in the input vector space. To avoid that the algorithm applies the results on Chebyshev-best approximation [44] discarding input vectors that can be well approximated by a fixed point of the LAAM constructed from the current set of lattice sources. The method is detailed in algorithm 3:

Algorithm 3 Incremental Lattice Source Induction Algorithm (ILSIA)

```
1. Initialize the set of lattice sources $\mathbf{E} = \{\mathbf{e}_1\}$ with a randomly picked pixel-vector in the input hyperspectral image $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{N}\}$.
2. Construct the LAAM based on the strong lattice independent (SLI) vectors: $\mathbf{W}_{EE}$.
3. For each data vector $\mathbf{x}_\mathrm{j}$;j=1,...,N
   (a) if $\mathbf{x}_\mathrm{j} = \mathbf{W}_{EE} \boxvee \mathbf{x}_\mathrm{j}$ then $\mathbf{x}_j$ is lattice dependent on the set of lattice sources $X$, skip further processing.
   (b) if $\varsigma\left(\mathbf{W}_{EE} \boxvee \left(\mu + \mathbf{e}^{\#}\right), \mathbf{x}_j\right) < \theta$, where $\mathbf{e}^{\#} = \mathbf{W}^{*}_{EE} \boxwedge \mathbf{x}_j$ and $\mu = \frac{1}{2}\left(\left(\mathbf{W}_{EE} \boxvee \mathbf{e}^{\#}\right) \boxvee \mathbf{x}_j\right)$, then skip further processing.
   (c) test max/min dominance to ensure SLI, consider the enlarged set of lattice sources $\mathbf{E}' = \mathbf{E} \cup \left\{\mathbf{x}_j\right\}$
       i. $\mu_1 = \mu_2 = 0$
       ii. for $i = 1, \ldots, K+1$
       iii. $\mathbf{s}_1 = \mathbf{s}_2 = \mathbf{0}$
            A. for $j = 1, \ldots, K+1$ and $j \neq i$
               $\mathbf{d} = x_i - \mathbf{e}_j$; $m_1 = \max(\mathbf{d})$; $m_2 = \min(\mathbf{d})$.
               $\mathbf{s}_1 = \mathbf{s}_1 + (\mathbf{d} == m_1)$, $\mathbf{s}_2 = \mathbf{s}_2 + (\mathbf{d} == m_2)$.
            B. $\mu_1 = \mu_1 + (\max(\mathbf{s}_1) == K)$ or $\mu_2 = \mu_2 + (\max(\mathbf{s}_2) == K)$.
       iv. If $\mu_1 = K+1$ or $\mu_1 = K+1$ then $\mathbf{E}' = \mathbf{E} \cup \left\{\mathbf{x}_j\right\}$ is SLI, go to 2 with the enlarged set of lattice sources and resume exploration from $j+1$.
4. The final set of lattice endmembers is $\mathbf{E}$.
```

3.3.6 WM

The WM algorithm was proposed in [108, 113]. Given an hyperspectral image $\mathbf{H}$, it is reshaped to form a matrix $\mathbf{X}$ of dimension $N \times L$, where N is the number of image pixels, and L is the number of spectral bands. The algorithm starts by computing the minimal hyperbox covering the data, $\mathscr{B}(\mathbf{v}, \mathbf{u})$, where $\mathbf{v}$ and $\mathbf{u}$ are the *minimal* and *maximal corners,* respectively, whose components are computed as follows:

$$v_k = \min_{\xi} x_k^{\xi} \text{ and } u_k = \max_{\xi} x_k^{\xi};\ k = 1,\dots,L;\ \xi = 1,\dots,N. \qquad (3.10)$$

Next, the WM algorithm computes the dual erosive and dilative Lattice Auto-Associative Memories (LAAMs), $\mathbf{W}_{XX}$ and $\mathbf{M}_{XX}$, as described in appendix C.

The columns of $\mathbf{W}_{XX}$ and $\mathbf{M}_{XX}$ are scaled by $\mathbf{v}$ and $\mathbf{u}$, forming the additive scaled sets $W = \left\{\mathbf{w}^k\right\}_{k=1}^{L}$ and $M = \left\{\mathbf{m}^k\right\}_{k=1}^{L}$:

$$\mathbf{w}^k = u_k + \mathbf{W}^k;\ \mathbf{m}^k = v_k + \mathbf{M}^k,\ \forall k = 1,\dots,L, \qquad (3.11)$$

where $\mathbf{W}^k$ and $\mathbf{M}^k$ denote the k-th column of $\mathbf{W}_{XX}$ and $\mathbf{M}_{XX}$, respectively. Finally, the set $\mathbf{E} = W \cup M \cup \{\mathbf{v},\mathbf{u}\}$ contains the vertices of the convex polytope covering all the image pixel spectra represented as points in the high dimensional space.

The algorithm is simple and fast but the number of induced endmembers, the amount of column vectors in V, can be too large for practical purposes. Furthermore, some of the endmembers induced that way can show high correlation even if they are affine independent. To obtain a meaningful set of endmembers, we search for an optimal subset of V in the sense of minimizing the unmixing residual error and the number of endmembers.

Algorithm 4 WM endmember induction algorithm

```
1. $L$ is the number of the spectral bands and $N$ is the number of data samples.
2. Compute $\mathbf{v} = [v_1,\dots,v_L]$ and $\mathbf{u} = [u_1,\dots,u_L]$,
       $v_k = \min_{\xi} x_k^{\xi}; u_k = \max_{\xi} x_k^{\xi}$
   for all $k = 1,\dots,L$ and $\xi = 1,\dots,N$,
3. Compute the LAAMs
       $\mathbf{W}_{XX} = \bigwedge_{\xi=1}^{N} \left[\mathbf{x}^{\xi} \times \left(-\mathbf{x}^{\xi}\right)'\right]; \mathbf{M}_{XX} = \bigvee_{\xi=1}^{N} \left[\mathbf{x}^{\xi} \times \left(-\mathbf{x}^{\xi}\right)'\right]$
   where $\times$ is any of the $\boxed{\vee}$ or $\boxed{\wedge}$ operators.
4. Build $W = \left\{\mathbf{w}^1,\dots,\mathbf{w}^L\right\}$ and $M = \left\{\mathbf{m}^1,\dots,\mathbf{m}^L\right\}$ such that
       $\mathbf{w}^k = u_k + \mathbf{W}^k; \mathbf{m}^k = v_k + \mathbf{M}^k; k = 1,\dots,L.$
5. Return the set $\mathbf{E} = W \cup M \cup \{\mathbf{v},\mathbf{u}\}$.
```

3.3.7 sWM

The main problem that WM faces is that it proposes a large set of candidates. Many of those candidates are highly correlated with each other. If the number of bands is L, then the set of endmember candidates $\mathbf{E}$ will have $2L+2$ signatures. The algorithm proposed here, called Sparse WM (sWM for short), aims to reduce the number of endmembers. It would reduce the complexity of the problem so that a small set of induced endmembers can be considered a dictionary for unmixing the data following a sparse strategy seen in section 3.2.2.

The endmember selection procedure follows a clustering rationale. The idea is that highly correlated endmembers will form a cluster. Loosely correlated endmembers will correspond to different elements present in the scene. First, the proposed method finds the number of underlying endmember clusters. Secondly, it clusters the data into that number of groups. Thirdly, it calculates the endmember closest to the centroid of said clusters. These endmembers will conform the final set of endmembers $\tilde{\mathbf{E}}$.

The initial set of endmember candidates is $|\mathbf{E}|$, where $\mathbf{E}$ is obtained via WM as described in algorithm 4. The number of cluster is selected via clustering the data fitted in a Gaussian Mixture Distribution (GMD). Let's suppose that the set $\mathbf{E}$ is a mixture of Gaussians. Then, the maximum likelihood estimates of the parameters of the Gaussian Mixture Model (GMM) can be calculated using an Expectation Maximization (EM) algorithm. This fitting is used to cluster the endmembers. The accuracy of this partitioning of the data is evaluated calculating the sum of the silhouette of all the endmember candidates. The silhouette value for each endmember is a measure of how similar that endmember is to endmembers in its own cluster, when compared to endmembers in other clusters. This process is repeated fitting the GMM with $k = 2, 3, \dots, 20$ components. The resulting clustering with a given k that results in the biggest sum of silhouettes is considered the best fit. Consequently, the underlying number of clusters is deduced to be that k value.

The final clustering process is conducted via k-means. The well known algorithm is used with the previously calculated number of clusters k, using correlation distance and replicating the clustering multiple times in order to retain the best fitting partition. After that, the endmembers closest to the centroids form the final endmember set $\tilde{\mathbf{E}}$. This endmembers will be later used as the dictionary to solve the unmixing problem reformulated as a sparse regression problem.

3.4 Experimental design

The experiments have been designed towards assessing the quality of unmixing that different algorithms provide. The underlying logic is that, after extracting endmembers and calculating abundances, the reconstructed hyperspectral image $\hat{\mathbf{X}} = \mathbf{E}\alpha$ should be as equal to the original image $\mathbf{X}$ as possible. This criterion has been measured using three error calculations: Mean Squared Error (MSE), Mean Average Error (MAE) and Mean Angular Distance (MAD). These measurements are detailed in appendix B. The experimental design, for each image $\mathbf{X}$ and each EIA f can be summarized as follows:

1. Calculate the set of induced endmembers $\mathbf{E} = f(\mathbf{X})$.
2. Calculate the abundances α such that $\mathbf{X} \simeq \mathbf{E}\alpha$ by solving the NCLS problem of the form $\min_\alpha \|\mathbf{E}\alpha - \mathbf{X}\|^2$ where $\alpha = 0$.
3. Reconstruct the image $\hat{\mathbf{X}} = \mathbf{E}\alpha$.
4. Calculate the reconstruction errors MSE, MAE and MAD.

Additionally, it is interesting to see if the different methods can propose a viable skin endmember -i.e. a pure endmember that best represents the human skin pixels. The assessment of this condition has been evaluated with the following procedure:

1. For each image $\mathbf{X}$, calculate the average skin pixel.
2. For every EIA, from all the proposed endmembers, calculate the correlation distance to that average skin pixel.
3. The best skin endmember will be that which minimizes the correlation distance to the average skin pixel.

3.4.1 Parameter selection

The main problem that hyperspectral unmixing faces, from an experimental point of view, is choosing the number of endmembers present on the data. Regarding this issue, the algorithms that have been used on these experiments can be divided in three groups:

- Algorithms requiring an explicit number of endmembers:

  NFINDR, FIPPI and ATGP require a number of endmembers to be specified. Instead of setting an arbitrary number, all three methods are preceded by an analytical algorithm that chooses the number of endmembers. The Harsanyi–Farrand–Chang (HFC) [18, 49] was chosen for that task. This method, also known as the Neyman–Pearson detection theory-based eigenthreshold method, was first proposed in [49]. It can be summarized as follows: Let the eigenvalues generated by the sample correlation matrix and the sample covariance matrix be denoted by correlation eigenvalues and covariance eigenvalues, respectively. The component dimensionality is equal to the total number of eigenvalues. Consequently, each eigenvalue corresponds to a component dimension and gives an idea of the significance of that particular component in terms of energy or variance. A null hypothesis is proposed, which indicates the case of the zero difference: For components without signal source, the corresponding correlation eigenvalue and covariance eigenvalue in these components should reflect only noise, in which case, correlation eigenvalue and covariance eigenvalue are equal. The alternative hypothesis indicates the case that the difference is greater than zero. When the Neyman–Pearson test is applied to a pair of correlation eigenvalue and its corresponding covariance eigenvalue, a failure of the test implies that there is a signal source in this particular component. The number of times the test fails indicates how many signal sources are present in the image. The method is detailed in algorithm 5.

- Algorithms requiring a threshold value parameter:

  ILSIA requires a Chebyshev-best approximation tolerance threshold. The bigger this number, the smaller the set of induced endmembers is. Similarly, EIHA requires a perturbation tolerance threshold. Equally, higher values reduce the number of induced endmembers. The relation between these values and the number of endmembers depends on the nature of the data. There is not an analytical way of establishing a threshold value that would drop a desired amount of endmembers. One approach could be to calculate HFC and then run the endmember induction algorithm, varying the threshold, until the number of endmembers indicated by HFC is obtained. Evidently, this would be very time consuming. Therefore, we have set these threshold values manually. The threshold for ILSIA in radiance response images was let to the default 0. In the case of EIHA, the threshold was set to 125. The reflectance images were unmixed using a threshold of 12 for ILSIA and 2750 for EIHA. The goal was to achieve a reasonable execution time and consequently a not-too-big set of endmembers.

- Algorithms without parameters:

  It is immediately deduced from the description of WM algorithm in section 3.3 (see algorithm 4) that for a hyperspectral image with L bands the number of induced endmembers will be $2L+2$. The images that were tested have 128 bands, thus WM will propose 258 endmember candidates. Regarding sWM, the quantity of endmembers will always be smaller than that of WM.

Algorithm 5 Harsanyi–Farrand–Chang (HFC) eigenthresholding method for the estimation of the number of endmembers in hyperspectral imagery.

1. Lets have hyperspectral image $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{N}\} \subset \mathbb{R}^L$.

2. Calculate the sample correlation matrix $\mathbf{R}_{L\times L}$ and the sample covariance matrix $\mathbf{K}_{L\times L}$.

3. Calculate the set of eigenvalues for both matrices, sorting them in descending order. Lets denote the sets Λ_R and Λ_K.

4. Calculate the difference and variance matrices $\mathbf{D} = \Lambda_R - \Lambda_K$ and $\Sigma = \sqrt{\frac{2\left(\Lambda_R^2+\Lambda_K^2\right)}{N}}$.

5. Calculate the L normal inverse cumulative distribution functions τ_i, $i = 1, \ldots, L$ using zero mean and variances $\Sigma = \{\sigma_1, \ldots, \sigma_L\}$ at the corresponding probabilities with desired probability $p = 10^{-5}$. The normal inverse function at zero mean is defined as

   $$\tau_i = F^{-1}\left(p|0, \sigma_i\right) = \left\{\tau_i : F\left(p|0, \sigma_i\right) = p\right\},$$

   whose result is the solution of

   $$p = \frac{1}{\sigma_i\sqrt{2\pi}} \int_{-\infty}^{\tau_i} e^{\frac{t^2}{2\sigma_i^2}}\, dt.$$

6. The number of endmembers, M, is given by the number of times the the correlation eigenvalue is greater than its corresponding covariance eigenvalue -implying that there is an endmember contributing to the correlation eigenvalue in addition to noise- with a tolerance threshold τ. Therefore, for the set $\mathbf{X}$:

   $$M = \sum_{i=1}^{L} \left(d_i > \tau_i\right), \, d_i \in \mathbf{D}$$

That number is calculated by the algorithm and is manually bound to lay between 2 and 20.

Both FIPPI and NFINDR algorithms have a loop that can run indefinitely. In order to avoid excessive execution time, it was established a maximum iteration cap of $100 \cdot M$ -a hundred times the number of endmembers to be induced that is calculated using HFC. Consequently, it is possible to end the execution of either FIPPI or NFINDR having extracted less endmembers that the amount suggested by HFC. The used implementation of FIPPI does not have a stopping rule depending on M, so it is also possible that the final set of endmembers is bigger than M.

Using correlation distance in sWM endmember selection process is risky. If a endmember candidate would have a standard deviation close to zero, the endmember should not be considered in the clustering process. One of the candidates that WM selects is the *minimal corner* $\mathbf{v} = [v_1, \ldots, v_L]$, where $v_k = \min_\xi x_k^\xi$. It is sensible to consider that each wavelength can measure zero radiance in a given pixel. If that would be the case, then $\mathbf{v} = 0$ and therefore $\text{std}(\mathbf{v}) = 0$. As a measure of precaution, it is reasonable to let $\mathbf{v}$ out of the k-means calculations. The number of underlying clusters was selected following the procedure described in section 3.3.7. The results of calculating the sum of silhouettes is illustrated in figures 3.1 and 3.2.

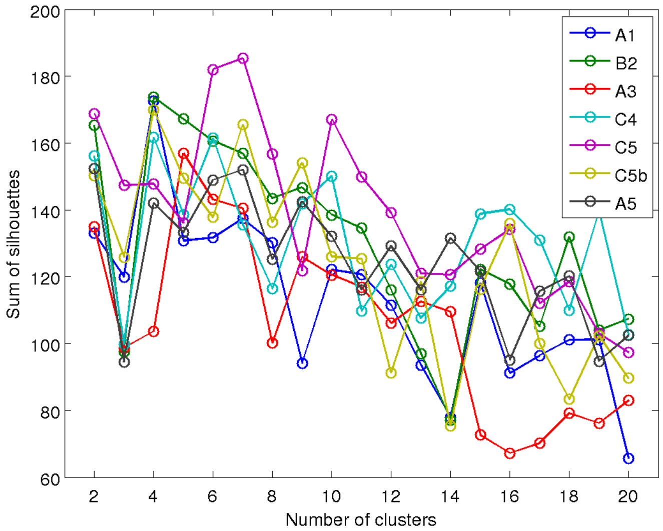

Figure 3.1: Sum of silhouettes calculated for every radiance image, with different cluster sizes. The maximum of each image is selected as the number o0f endmembers in sWM algorithm.

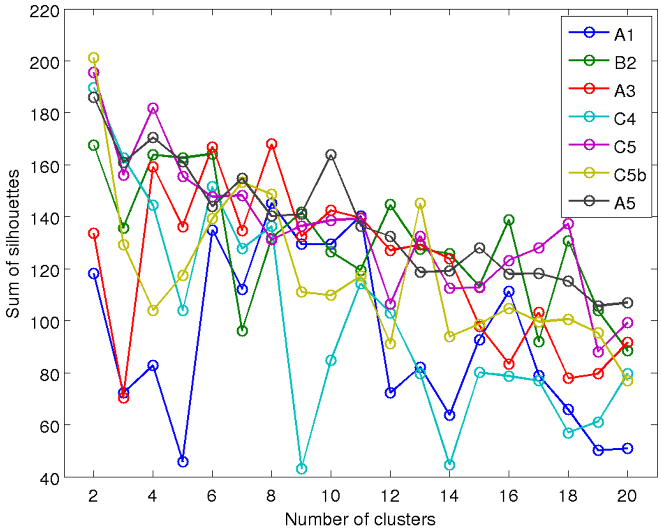

Figure 3.2: Sum of silhouettes calculated for every reflectance image, with different cluster sizes. The maximum of each image is selected as the number of endmembers in sWM algorithm.

3.5 Experimental results

The unmixing results were evaluated taking into consideration the computational cost, the ability to induce a small set of endmembers, the reconstruction error and the capabilities of extracting skin endmembers.

WM is the fastest algorithm, foolowed by sWM and EIHA, as can be seen in table 3.1. However, both WM and sWM take constant time, meaning that it does not depend on algorithm parameters. EIHA on the other hand can be more costly if the threshold parameter is lower. ATGP, FIPPI and ILSIA are notably slower. It is inportant to notice that the higher the tolerance parameter ILISA has, the fewer endmembers it proposes, taking more time to execute. Regarding NFINDR, it is known to be a very slow algorithm that needs a maximum number of iteration to be set if the computational cost is to be compared to that of other algorithms.

The parameter estimation is the main throwback of EIHA and ILSIA. The same value for all seven radiance and reflectance images was used on this experiments, as stated in section 3.4.1. Table 3.2 shows the disparate number of endmember induced from different images. The method sWM is capable of selecting a reduced set of endmembers from the standard WM prodedure, form 258 to between 2 and 8. Finally, table 3.3 shows that NFINDR and ILSIA are the algorithms that take the most time per induced endmember, while WM is the fastest one.

Radiance images							
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM
A1	2392.40	385.19	597.39	59.12	963.10	31.88	79.74
B2	2967.55	278.91	739.40	46.95	986.38	31.61	71.90
A3	2079.12	220.52	426.79	45.99	835.26	29.55	71.50
C4	738.38	168.82	145.20	85.83	569.18	29.63	72.22
C5	2748.32	302.95	691.80	48.11	1131.19	31.98	74.06
C5b	5405.33	302.28	958.15	61.32	1160.06	31.82	73.20
A5	16543.12	374.57	1275.34	48.13	352.76	31.83	70.78
mean	4696.32	290.46	690.58	56.49	856.85	31.19	73.34
Reflectance images							
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM
A1	4350.35	476.46	1052.09	5.55	632.63	32.97	73.04
B2	950.48	179.43	254.30	71.83	605.65	28.99	73.70
A3	1073.90	207.75	343.91	320.30	598.18	28.87	73.66
C4	65.95	40.15	28.09	37.51	621.69	28.77	74.87
C5	1315.04	171.38	240.16	100.79	601.21	28.89	74.59
C5b	593.47	146.98	197.17	82.88	597.59	28.91	74.37
A5	1070.99	206.06	367.22	133.62	606.10	28.88	73.29
mean	1345.74	204.03	354.71	107.49	609.01	29.47	73.93

Table 3.1: Unmixing hyperspectral images: Execution times.

Radiance images							
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM
A1	39	39	38	198	18	258	4
B2	38	38	87	131	16	258	4
A3	30	30	65	170	15	258	5
C4	22	22	5	332	10	258	4
C5	41	41	32	106	17	258	7
C5b	41	41	96	101	18	258	4
A5	51	51	95	100	6	258	2
Reflectance images							
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM
A1	43	43	128	2	6	258	8
B2	25	25	48	198	10	258	2
A3	28	28	63	46	6	258	8
C4	6	6	12	73	12	258	2
C5	23	23	47	356	5	258	2
C5b	20	20	45	283	4	258	2
A5	28	28	60	449	9	258	2

Table 3.2: Unmixing hyperspectral images: Number of induced endmembers.

Radiance images							
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM
A1	61.344	9.877	15.721	0.299	53.506	**0.124**	19.935
B2	78.093	7.340	8.499	0.358	61.649	**0.123**	17.976
A3	69.304	7.351	6.566	0.271	55.684	**0.115**	14.300
C4	33.563	7.674	29.041	0.259	56.918	**0.115**	18.056
C5	67.032	7.389	21.619	0.454	66.541	**0.124**	10.580
C5b	131.837	7.373	9.981	0.607	64.448	**0.123**	18.300
A5	324.375	7.345	13.425	0.481	58.793	**0.123**	35.390
mean	109.364	7.764	14.979	0.390	59.648	**0.121**	19.220
Reflectance images							
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM
A1	101.171	11.080	8.219	2.775	105.439	**0.128**	9.130
B2	38.019	7.177	5.298	0.363	60.565	**0.112**	36.849
A3	38.353	7.420	5.459	6.963	99.696	**0.112**	9.207
C4	10.991	6.691	2.341	0.514	51.807	**0.112**	37.436
C5	57.175	7.451	5.110	0.283	120.243	**0.112**	37.294
C5b	29.674	7.349	4.382	0.293	149.396	**0.112**	37.184
A5	38.250	7.359	6.120	0.298	67.344	**0.112**	36.646
mean	44.805	7.790	5.276	1.641	93.499	**0.114**	29.107

Table 3.3: Unmixing hyperspectral images: Execution time per induced endmember

Tables 3.4, 3.5 and 3.6 show the reconstruction error results. Regarding radiance images, it is clear that EIHA and ATGP are the methods that allow the better reconstruction. Regarding reflectance images, the results are more similar for all the algorithms. However, it is important to note that this error measures -explained in appendix B- reconstruct the image using the Linear Mixing Model, therefore inside the scope of linear algebra. This is detrimental for WM and sWM, which use exclusively lattice algebra. The proper reconstruction method would involve using $\boxed{\vee}$ and $\boxed{\wedge}$ operators over the $\mathbf{W}_{XX}$ and $\mathbf{M}_{XX}$ auto-associative memories respectively. Nevertheless, the reconstruction process was limited to linear algebra for the sake of comparability.

The capability of extracting skin endmember was similar for all methods. The results for radiance images are shown in figures 3.3 to 3.9. The correlation distances, seen in table 3.7, show similar results in all experiments. It is interesting to note that despite the small number of induced endmembers, sWM still has comparable results to those of other methods. The graphics pertaining to reflectance images -figures 3.10 to 3.16- offer less visual help understanding the results. The irregularity shown in these plots can be consequence of many factors, the main hypothesis being that the signal recieved in white standard area used to calculate

Radiance images								
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM	*mean*
A1	30.97	29.17	37.80	**18.32**	38.82	217.67	200.87	81.95
B2	79.91	62.32	84.88	**50.29**	268.55	722.62	1001.02	324.23
A3	106.25	81.12	152.54	**43.50**	595.83	2534.26	3525.47	1005.57
C4	521.13	217.25	24403.30	**89.08**	6656.58	9036.79	76799.92	16817.72
C5	148.00	53.54	260.50	**41.68**	143.29	357.49	964.14	281.23
C5b	73.22	54.10	70.17	**43.56**	221.72	413.65	550.63	203.86
A5	73.17	51.96	57.75	**41.91**	258.04	415.29	1109.52	286.81
mean	147.52	78.49	3580.99	**46.91**	1168.97	1956.82	12021.65	
Reflectance images								
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM	*mean*
A1	24.45	28.80	**19.52**	47.61	49.94	43.86	43.60	36.82
B2	5.32	5.72	4.57	**2.18**	5.94	5.77	7.59	5.30
A3	45.60	47.54	39.87	**44.96**	66.45	67.80	69.68	54.56
C4	27.20	20.94	17.42	**4.93**	14.95	22.84	56.74	23.57
C5	26.41	29.24	25.54	**10.12**	35.53	29.53	34.12	27.21
C5b	20.80	22.29	18.85	**7.68**	24.14	21.06	24.13	19.85
A5	25.11	26.80	26.56	**10.75**	34.43	33.52	39.06	28.03
mean	24.98	25.90	21.76	**18.32**	33.05	32.05	39.27	

Table 3.4: Mean squared reconstruction error (MSE) of unmixed hyperspectral images.

Radiance images								
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM	*mean*
A1	4.13	3.92	4.12	**3.27**	4.08	10.47	10.24	5.75
B2	6.45	5.84	6.37	**5.29**	9.42	17.20	24.21	10.68
A3	6.92	6.31	8.14	**5.04**	10.61	37.49	42.61	16.73
C4	14.84	10.15	97.80	**6.98**	33.85	63.78	201.38	61.26
C5	7.08	5.30	9.21	**4.84**	7.31	10.98	21.97	9.53
C5b	5.83	5.34	5.78	**4.93**	8.65	12.56	15.95	8.43
A5	5.76	5.23	5.33	**4.85**	9.26	12.28	23.49	9.46
mean	7.29	6.01	19.54	**5.03**	11.88	23.54	48.55	
Reflectance images								
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM	*mean*
A1	**2.62**	2.84	2.36	3.31	3.65	3.63	3.37	3.11
B2	1.25	1.31	1.20	**0.62**	1.06	0.86	1.17	1.07
A3	2.87	2.99	**2.71**	2.83	3.15	3.90	3.91	3.19
C4	2.59	2.30	2.08	**1.12**	2.09	3.15	4.64	2.57
C5	2.88	3.01	2.94	**1.39**	2.12	1.77	2.11	2.32
C5b	2.57	2.63	2.53	**1.16**	1.96	1.54	1.80	2.03
A5	2.42	2.56	2.89	**1.41**	2.15	1.88	2.27	2.23
mean	2.46	2.52	2.39	**1.69**	2.31	2.39	2.75	

Table 3.5: Mean absolute reconstruction error (MAE) of unmixed hyperspectral images.

Radiance images								
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM	*mean*
A1	0.0443	0.0436	0.0382	**0.0331**	0.0420	0.1643	0.1569	0.0746
B2	0.0123	0.0077	0.0058	**0.0051**	0.0259	0.0309	0.0430	0.0187
A3	0.0118	0.0104	0.0114	**0.0069**	0.0204	0.1544	0.2176	0.0618
C4	0.0016	0.0008	0.0525	**0.0004**	0.0065	0.0189	0.1720	0.0361
C5	0.0211	0.0037	0.0078	**0.0030**	0.0154	0.0148	0.0323	0.0140
C5b	0.0050	0.0036	0.0037	**0.0029**	0.0217	0.0158	0.0253	0.0112
A5	0.0057	0.0041	0.0036	**0.0035**	0.0210	0.0193	0.0436	0.0144
mean	0.0146	0.0106	0.0176	**0.0079**	0.0218	0.0598	0.0987	
Reflectance images								
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM	*mean*
A1	**0.1763**	0.2076	0.1477	0.3086	0.3716	0.2705	0.2761	0.2512
B2	0.1934	0.2100	0.1838	0.0577	0.1260	**0.0523**	0.0938	0.1310
A3	0.2401	0.2536	0.2302	**0.2112**	0.2650	0.2348	0.2619	0.2424
C4	0.1142	0.0889	0.0685	**0.0196**	0.0668	0.0856	0.2126	0.0937
C5	0.3450	0.3660	0.3553	0.1111	0.1860	**0.1000**	0.1301	0.2277
C5b	0.3547	0.3635	0.3466	0.1089	0.2138	**0.1024**	0.1304	0.2315
A5	0.2539	0.2810	0.3332	0.1148	0.1906	**0.1021**	0.1373	0.2019
mean	0.2396	0.2529	0.2379	**0.1331**	0.2028	0.1354	0.1775	

Table 3.6: Mean angular distance (MAD) reconstruction error of unmixed hyperspectral images.

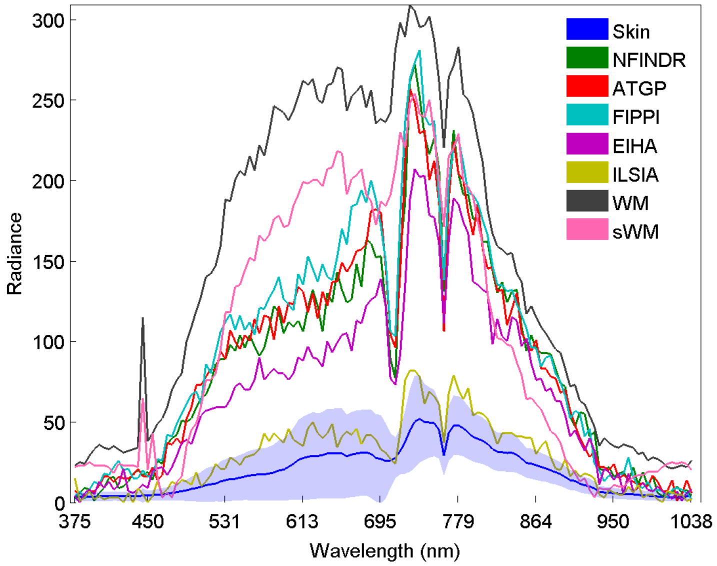

Figure 3.3: Average skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the endmembers closest to the mean for each of the EIAs, for radiance image A1.

the reflectance is quite noisy. Numerical results from 3.7 are more enlightening. The results are worse than those obtained over radiance images. The best algorithm seems also to be EIHA. Agian, despite inducing in some cases only 2 endmembers, the results of sWM are comparable to those of the other algorithms.

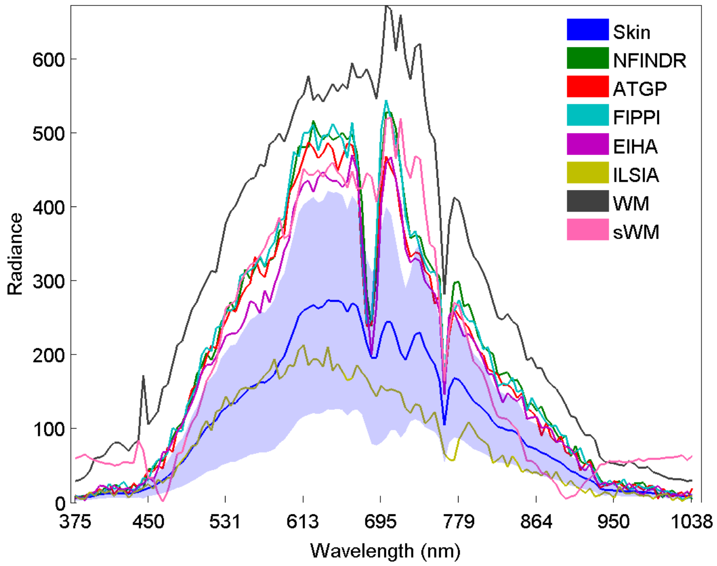

Figure 3.4: Average skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the endmembers closest to the mean for each of the EIAs, for radiance image B2.

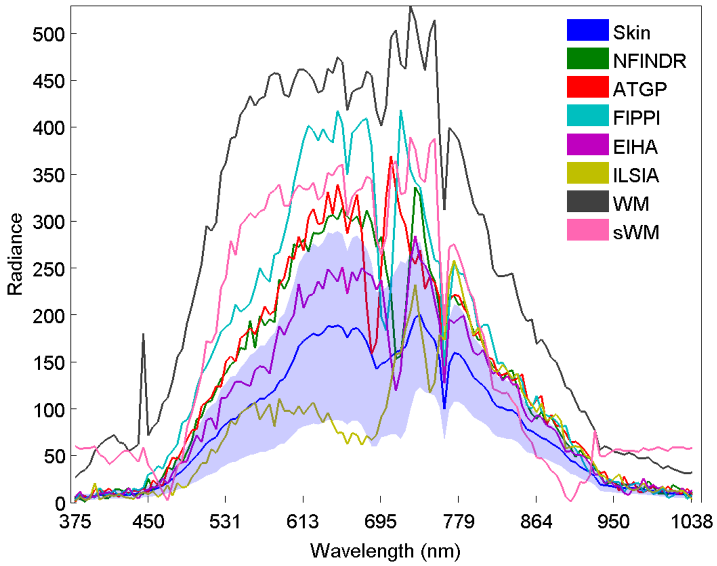

Figure 3.5: Average skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the endmembers closest to the mean for each of the EIAs, for radiance image A3.

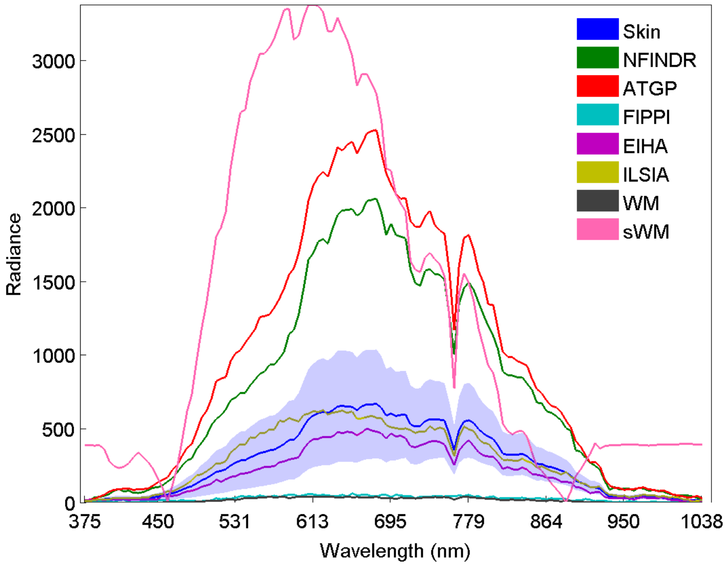

Figure 3.6: Average skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the endmembers closest to the mean for each of the EIAs, for radiance image C4.

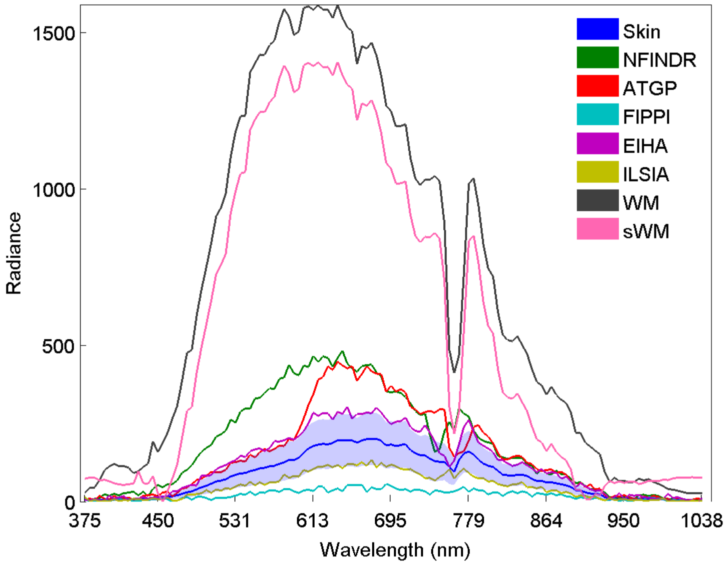

Figure 3.7: Average skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the endmembers closest to the mean for each of the EIAs, for radiance image C5.

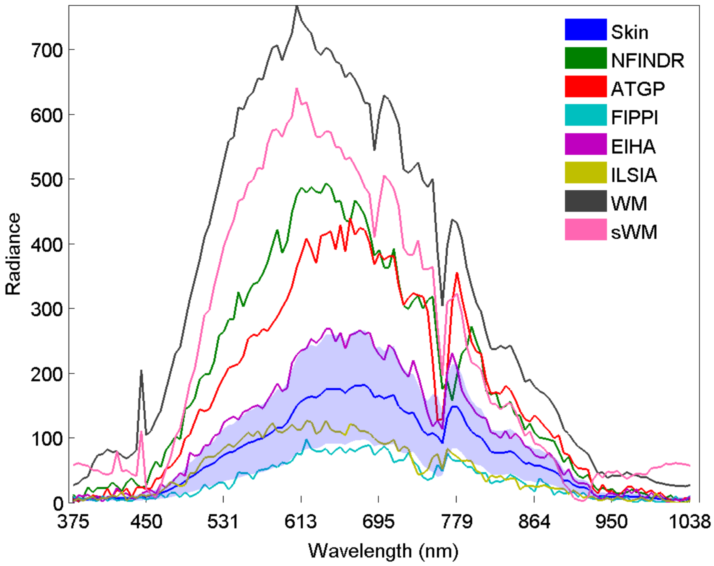

Figure 3.8: Average skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the endmembers closest to the mean for each of the EIAs, for radiance image C5b.

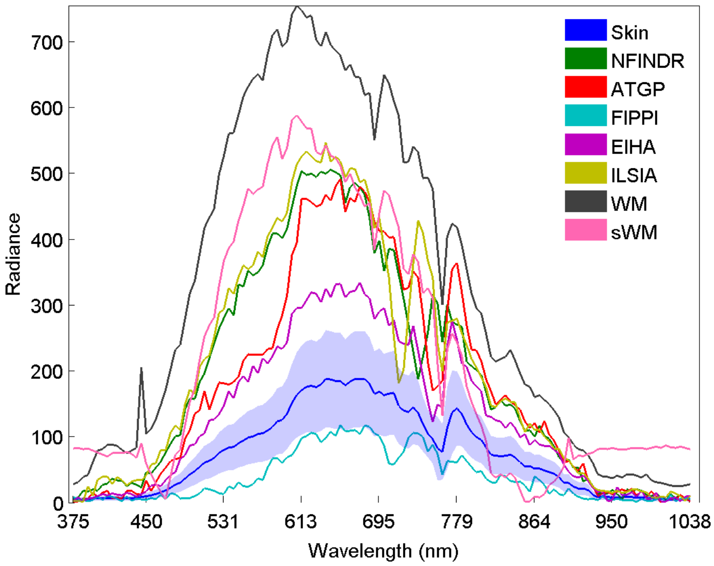

Figure 3.9: Average skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the endmembers closest to the mean for each of the EIAs, for radiance image A5.

Radiance images							
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM
A1	0.029	0.041	0.043	**0.023**	0.038	0.127	0.128
B2	**0.013**	0.015	0.019	**0.013**	0.057	0.025	0.044
A3	0.021	0.031	0.037	**0.015**	0.234	0.053	0.086
C4	0.004	0.006	0.078	**0.001**	0.033	0.053	0.231
C5	0.062	0.027	0.069	**0.006**	0.013	0.095	0.112
C5b	0.061	0.012	0.020	**0.006**	0.098	0.098	0.109
A5	0.044	0.005	0.058	**0.006**	0.055	0.077	0.137
mean	0.033	0.020	0.046	**0.010**	0.075	0.076	0.121
Reflectance images							
	NFINDR	ATGP	FIPPI	EIHA	ILSIA	WM	sWM
A1	0.483	0.538	0.409	**0.332**	0.755	0.722	0.722
B2	**0.499**	0.616	0.541	0.541	0.497	0.667	0.698
A3	0.707	0.686	0.689	**0.464**	0.800	0.745	0.789
C4	0.520	0.484	0.422	**0.190**	0.335	0.253	0.389
C5	0.598	0.681	0.635	**0.536**	0.668	0.713	0.875
C5b	0.610	0.677	0.585	**0.506**	0.540	0.728	0.865
A5	0.681	0.709	0.615	**0.530**	0.705	0.751	0.950
mean	0.585	0.627	0.557	**0.443**	0.614	0.654	0.755

Table 3.7: Correlation distance to the mean skin pixel of the closest induced endmember.

3.6 Conclusion

The unmixing of hyperspectral images presents many challenges. Perfectly preprocessed and broadly used hyperspectral data is usually the ideal scenario. In this work the data was manually collected and labeled, adding complexity to the unmixing process. Noise and the conditions of image retrieval make difficult to obtain clean unmixing results. The conducted experiments show differences in performance in different images. Nevertheless, this set of experiments allows us to test the capabilities of unmixing algorithms under these circumstances. The results vary greatly depending on the number of extracted endmembers. The results were first evaluated using three measurements of reconstruction error. Overall, Lattice Computing methods show equal or better unmixing capabilities. EIHA is the algorithm that results in the smallest reconstruction error. The next unit of measurement was the ability to detect endmembers representing human skin. All methods showed good results, EIHA being again the most accurate. The proposed endmember selection step added to WM, called sWM, showed comparable results, taking into account the small number of endmembers. Summarizing, this experiments demonstrate that, in a situation where human detection is needed and visual

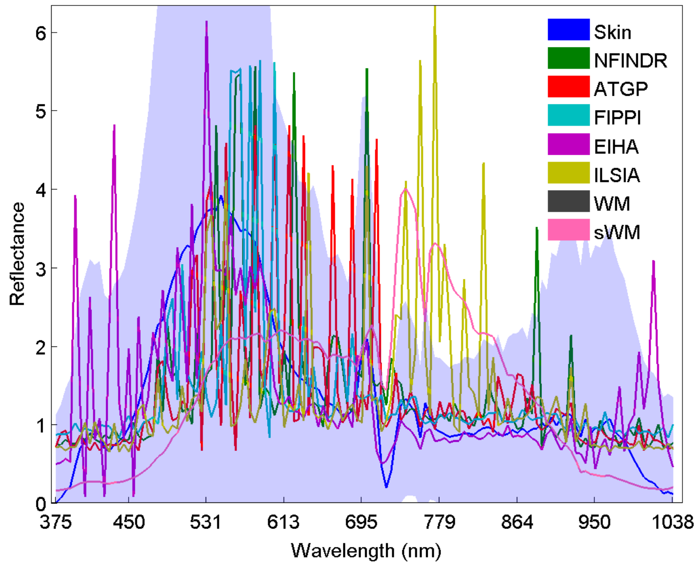

Figure 3.10: Average scaled and centered skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the scaled and centered endmembers closest to the mean for each of the EIAs, for reflectance image A1.

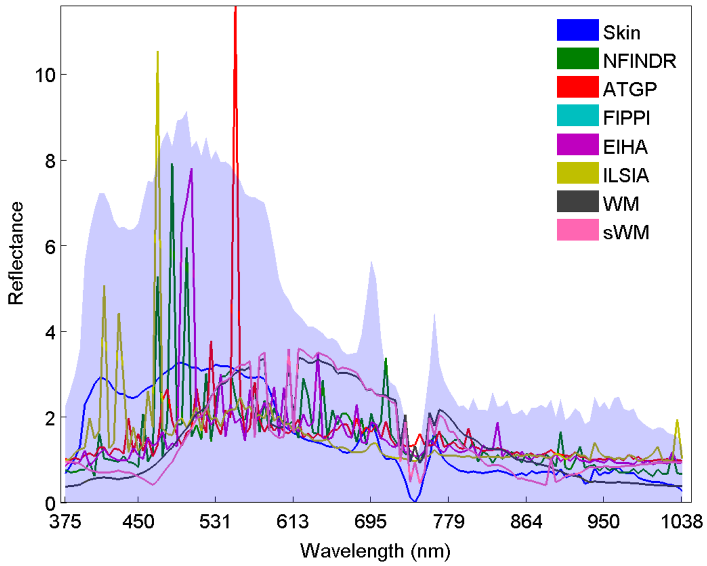

Figure 3.11: Average scaled and centered skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the scaled and centered endmembers closest to the mean for each of the EIAs, for reflectance image B2.

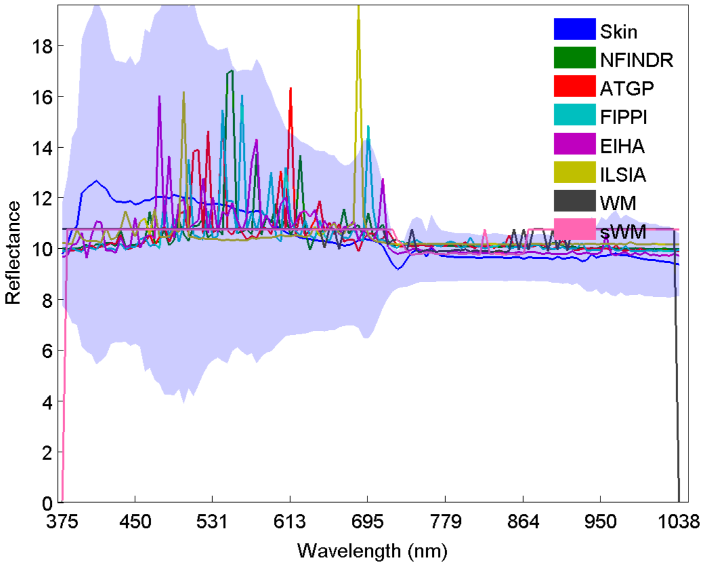

Figure 3.12: Average scaled and centered skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the scaled and centered endmembers closest to the mean for each of the EIAs, for reflectance image A3.

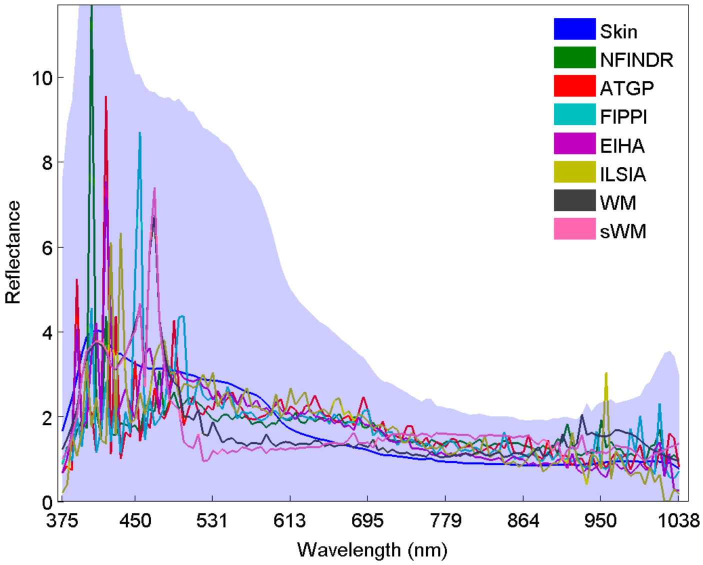

Figure 3.13: Average scaled and centered skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the scaled and centered endmembers closest to the mean for each of the EIAs, for reflectance image C4.

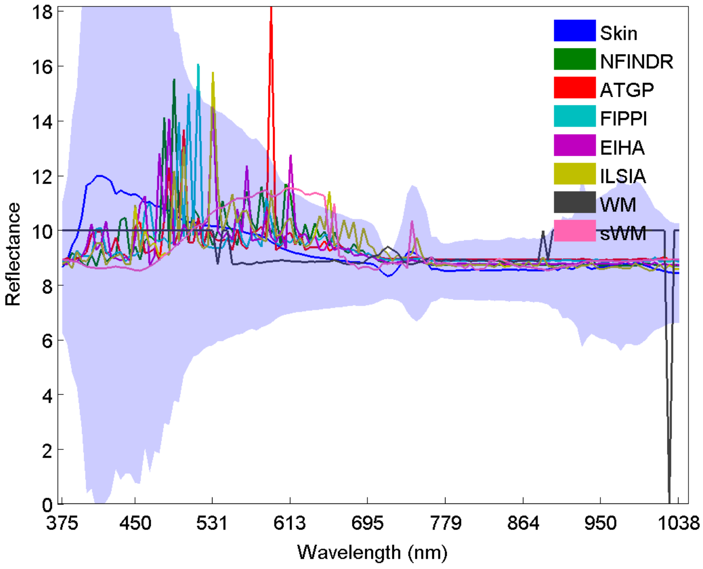

Figure 3.14: Average scaled and centered skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the scaled and centered endmembers closest to the mean for each of the EIAs, for reflectance image C5.

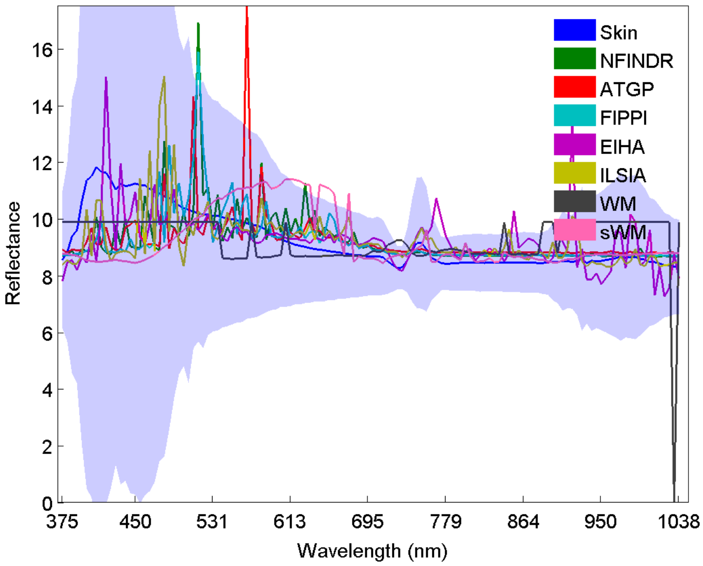

Figure 3.15: Average scaled and centered skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the scaled and centered endmembers closest to the mean for each of the EIAs, for reflectance image C5b.

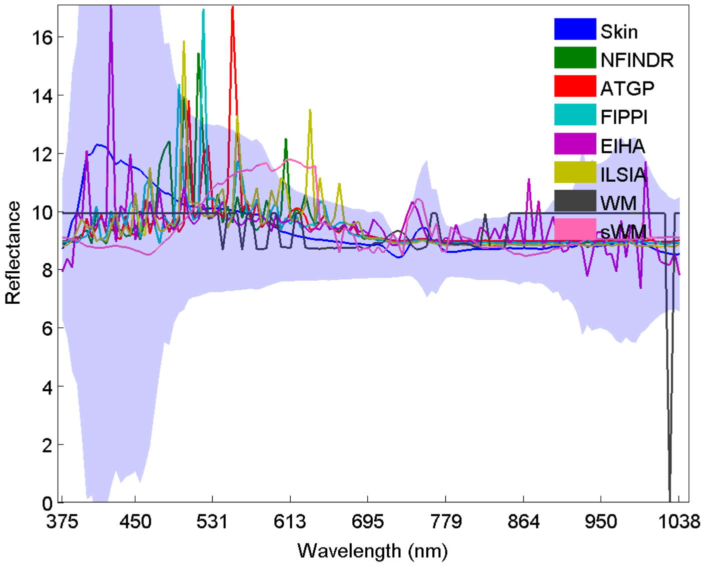

Figure 3.16: Average scaled and centered skin pixel (shadowed areas cover values under standard deviation) drawn in in blue, and the scaled and centered endmembers closest to the mean for each of the EIAs, for reflectance image A5.

or thermal information alone are not sufficient, it is possible to unmix hyperspectral data and extract significant endmembers belonging to skin regions.

Chapter 4

Face Recognition in Unbalanced Databases

Most researches that aim towards more capable face recognition focus on developing algorithms that achieve increasingly better results. The limitation of this scope is that the experiments are usually conducted over well balances *neat* databases. The face images are all equally sized, taken under the same circumstances, all subjects have the same number of photographs taken, with similar poses, etcetera. Real world applications call for algorithms that can function under *ugly* circumstances, i.e when dealing with unbalanced databases. Section 4.1 gives an introduction to Chapter contents. Feature extraction and classification algorithms are presented in Sections 4.2 and 4.3, respectively. The experimental design is detailed in Section 4.4. Computational experiment results are reported in Section 4.5. Finally, a concise discussion is presented in section 4.6.

4.1 Introduction

In statistical learning approaches, each face image is viewed as a point (vector) in a d-dimensional space. The face images often belong to a low dimension manifold. The high dimensionality of the data imposes the need for feature extraction processes previous to face classification. Therefore, the goal is to choose and apply the right statistical tool for the extraction and analysis of the manifold where the face images lie in this high dimensional space. These tools must define the embedded face space in the image space and extract the basis functions from the face space. Ideally, patterns belonging to different classes (identities) will occupy disjoint and compact regions in the feature space, which will be easy to discriminate by means of statistical or bio-inspired classifier systems. In the best case a linear discriminant would be enough to obtain good classification performance results. The earliest approach applied Principal Component Analysis (PCA) for feature extraction [133], other approaches use the variations of the Linear Discriminant Analysis (LDA) [150, 101, 141, 103, 15], or the Locality Preserving Projections (LPP) [50]. Other successful statistic tools include Bayesian networks [90], bi-dimensional regression [74], generative models [54], and ensemble based and other boosting methods [81]. In this Chapter we propose Lattice Independent Component Analysis (LICA) [43], using a Endmember Induction Algorithm (EIA) [134] based on Lattice Computing [38] to perform feature extraction and dimension reduction. This is a new approach to face recognition, although Lattice Computing approaches have been previously applied to fMRI imaging [40, 44], mobile robot localization [135] and hyperspectral image analysis [43, 109].

The classification system development process involves training a classifier from a data sample and testing the trained system on independent samples to guess the correct class. Translated into the face recognition paradigm, it means to train the system on a set of identified faces and then try to assign each new unknown face image to the correct identity. Extreme Learning Machine (ELM) constitute an innovative category of neural-network based classification and regression techniques [63]. Different kinds of ELM variations have been recently used in fields as diverse as sales forecasting [126], antiviral therapy [98], metal temperature prediction [131] or arrhythmia classification [76]. ELMs have been also applied in biometrics, specifically for on-line face detection [94] and fingerprint classification [87]. We provide a formal short review in Appendix 4.

Unbalanced class distribution of the data set [67], i.e. quite different *a priori* probabilities of the class intances, leads to big performance problems in most conventional classification building methods, because they tend to be biased to the most frequent class. However, most face recognition algorithms and classifiers are tested over well balanced databases like ORL, Yalefaces or Multi-PIE. Under such ideal circumstances, most classifiers and feature extraction methods mentioned before work successfully [22]. It is reasonable to think that the environments or devices that require face recognition will not always provide such well balanced databases. Therefore, it is relevant to address the face recognition task in these unfavorable conditions. We have used Color FERET database [96, 95] to create 4 unbalanced experimental databases. We have tested LICA and other well known algorithms for feature extraction using ELM as the classifier construction method for fair comparison of the feature extraction, i.e. avoiding bias due to classifier construction method. Additionally, the performance of ELM has been compared with other classifiers. The aim of these experiments was to test the proficiency of both LICA and ELM in the recognition of faces of a complex and unbalanced database. Experimental results indicate that, among the tested methods, LICA is the most effective feature extraction algorithm for face recognition under high subject-per-class variability. Experimental results also reveal that ELM is the classifier less sensitive to high class-variation induced noise.

4.2 Feature extraction algorithms

Feature extraction is the process of mapping the original data into a more effective feature space. The extracted features must preserve the best class separability possible in addition to dimension reduction. That is, if we have some data X, we find coefficients Y such that

$$X = A \cdot Y, \tag{4.1}$$

$$Y = A^{-1} \cdot X, \tag{4.2}$$

where A is the matrix of basis vectors for the feature extraction transformation. The data in X is therefore projected by its inverse A^{-1} into coefficients Y living in a more convenient feature space. We have tested some of the most widely used feature extraction algorithms: Principal Component Analysis (PCA) [133], Independent Component Analysis (ICA) [9, 93, 92, 28, 91, 88, 47] and Linear Discriminant Analysis (LDA) [8] along with Lattice Independent Component Analysis (LICA) [43]. Both PCA and LDA both try to find orthogonal projection directions with greatest variance of the prejection coefficients. While PCA is an unsupervised approach LDA is a supervised algorithm, using class label information. ICA sources need not be orthogonal, because it maximizes the source statistical independence. Finally, LICA is a Lattice Computing approach based on lattice independence. These algorithms are explained in more detail below.

4.2.1 Principal Component Analysis (PCA)

The PCA finds othogonal projection axes of the data in the order of decreasing projection variance. These directions are called principal components. Therefore, A^{-1} is formed by the principal components of the covariance matrix of X.

Let be a data-set composed of N images of n pixels, denoted by $X = \left\{\mathbf{x}_j; j = 1, \ldots, N\right\} \in \mathbb{R}^{n \times N}$, where each $\mathbf{x}_j$ is an image column vector. We center the data by subtracting the mean column. We want to find the eigenvectors $\mathbf{a}$ solving the eigen-problem:

$$\lambda \mathbf{a} = X\mathbf{a} \tag{4.3}$$

The Singular Value Decomposition of X given by $X = U \cdot S \cdot V^T$ where matrix U is the matrix of the eigenvectors of XX^T, S is the diagonal matrix of the eigenvalues. The data matrix X can be projected into a reduced spaced of dimensionality m by computing $Y = U_m^T X$, where U_m denotes the matrix composed of the first m columns of U.

4.2.2 Linear Discriminant Analysis (LDA)

PCA is unsupervised because it doesn't use the class information of data sample points. Linear Discriminant Analysis (LDA) searches for optimal class discrimination projections given data-set

$$X = \left\{ \mathbf{x}_j^k ; j = 1, \ldots, N; k = \{1, \ldots, C\} \right\} \in \mathbb{R}^{n \times N} \tag{4.4}$$

where data data samples are partitioned into C classes, $\mathbf{x}$ are n-dimensional vectors. Each class has m_k samples. Assume that the mean has been extracted from the samples, as in PCA. The objective function for the LDA can be defined [15] as

$$\mathbf{a}_{opt} = \arg\max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_t \mathbf{a}}, \tag{4.5}$$

$$S_b = \sum_{k=1}^{c} m_k \mu^k (\mu^k)^T \tag{4.6}$$

$$= \sum_{k=1}^{c} \left(\frac{1}{m_k} (\sum_{i=1}^{m_k} \mathbf{x}_i^k) \right) \left(\frac{1}{N_k} (\sum_{i=1}^{m_k} \mathbf{x}_i^k) \right),^T \tag{4.7}$$

$$S_t = \sum_{i=1}^{m} \mathbf{x}_i (\mathbf{x}_i)^T, \tag{4.8}$$

where μ is the total sample mean vector, μ^k is the mean vector of the k-th class and $\mathbf{x}_i^k$ is the i-th sample in k-th class. The total scatter matrix S_t and between-class scatter matrix S_b can be expressed in matrix form, if the sample vectors of each class are grouped together:

$$S_b = X W_{NxN} X^T, \tag{4.9}$$

$$S_t = XX^T, \tag{4.10}$$

where W_{NxN} is a diagonal matrix defined as

$$W_{NxN} = \begin{bmatrix} W^1 & 0 & \dots & 0 \\ 0 & W^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W^c \end{bmatrix} \tag{4.11}$$

and W^k is a $m_k \times m_k$ matrix

$$W^k = \begin{bmatrix} \frac{1}{m_k} & \frac{1}{m_k} & \dots & \frac{1}{m_k} \\ \frac{1}{m_k} & \frac{1}{m_k} & \dots & \frac{1}{m_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m_k} & \frac{1}{m_k} & \dots & \frac{1}{m_k} \end{bmatrix} \tag{4.12}$$

Finally, we can state LDA as the following eigenproblem:

$$S_b\mathbf{a} = \lambda S_t\mathbf{a}, \tag{4.13}$$

which is equivalent to

$$XW_{NxN}X^T(XX^T)^{-1}\mathbf{a} = \lambda\mathbf{a}. \tag{4.14}$$

The solution of this eigenproblem provides the eigenvectors needed to project the data in an analogous manner of PCA. When there are many variables, for instance if samples are images and observations are pixels, some previous dimensionality reduction must be performed.

4.2.3 Independent Component Analysis (ICA)

ICA is a generative model which aims to describe how the data is generated by mixing non-Gaussian, mutually statistically independent latent variables with and unknown mixing matrix [68]. Let us denote $\mathbf{x}$ the n-dimensional observed data vector and B the $n \times M$ mixing matrix. The mixing model is formulated for ICA as follows:

$$\mathbf{x} = B\mathbf{s}, \tag{4.15}$$

$$\mathbf{s} = V\mathbf{x}, \tag{4.16}$$

where $V = B^{-1}$ and $\mathbf{s}$ are the independent sources. If we consider the whole sample, the equation is rewritten as

$$S = VX \tag{4.17}$$

where $X = \{\mathbf{x}_j; j = 1, \ldots, N\} \in \mathbb{R}^{n \times N}$, each $\mathbf{x}_j$ being a face image column vector.

It has been shown that the mixing model is completely identifiable, up to a permutation and scale of the sources, if the sources are statistically independent and at least $M-1$ of them are non-Gaussian. In the case of M gaussian variables, the matrix B is not identifiable. It is also required that the number of sources is smaller than or equal to the number of available observations, i.e. $M \leq n$. The mixing and unmixing matrices can be estimated following three approaches: maximizing the nongaussianity, minimizing the mutual information and maximizing the likelihood. Quantitative measures of random variable nongaussianity are kurtosis, negentropy or approximations of negentropy. If the component are constrained to be uncorrelated, ICA estimation by minimization of mutual information is equivalent to maximizing the sum of nongaussianities. The constraint of uncorrelatedness simplifies the computations considerably. In the maximum likelihood estimation approach, the log-likelihood it's usually used, which is equivalent to entropy maximization, or “infomax”.

There are two possible ways of performing face recognition with ICA. We can treat the images as random variables and pixels as observation. This approach maximizes the independence of pixels It has been argued that it will produce better object recognition, since it implements recognition by parts [79]. Other approach is to treat pixels as variables and images and observations. Treating the face recognition problem from a wholistic approach, it has been demonstrated that it performs better [30]. In this work we chose the second option. We have used the DTU:ICA toolbox developed by the Technical University of Denmark [29].

Mean-field ICA

This method estimates sources from the mean of their posterior distribution and the mixing matrix (and noise level) is estimated by maximum a posteriori (MAP) [69]. The latter requires the computation of a good approximation to the correlations between sources. For this purpose, [69] propose three increasingly advanced mean-field methods: the variational (also known as naive mean field) approach, linear response corrections, and an adaptive version of the Thouless, Anderson and Palmer (TAP) mean-field approach [93, 92].

We have empirically searched for the best of those approaches on our problem. The followed criteria was recognition accuracy, constrained to a feasible execution time. The selected method uses a constant prior mixing matrix and noise covariance as well as a non-analytic power law source prior. The Mean-field method used was linear response correction.

ICA Infomax

The “infomax” framework original purpose was to maximize the output entropy of a neural network with non-linear outputs [9]. It is closely connected to the maximum likelihood estimation. For a data matrix $X = \left\{\mathbf{x}_j; j = 1, \ldots, N\right\} \in \mathbb{R}^{n\times N}$, the log-likelihood function has the form [68]

$$L = \sum_{i=1}^{t}\sum_{j=1}^{n} \log f_j(\mathbf{v}_j\mathbf{x}(i)) + t \cdot \log|\det V| \tag{4.18}$$

where $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \in \mathbb{R}^{t\times n}$ is the inverse of the source mixing matrix B. In our case, the function used is

$$L = t \cdot \log|\det V| - \sum_{i=1}^{t}\sum_{j=1}^{n} \log f_j(\mathbf{v}_j\mathbf{x}(i)) + N \cdot n \cdot \log(\pi) \tag{4.19}$$

where $f(x) = cosh(x)$.

ICA with Molgedey and Schuster decorrelation algorithm

ICA with the Molgedey and Schuster decorrelation algorithm (ICA-MS) uses the decorrelation algorithm presented in [88] to uncorrelate a some superimposed sources X and X_{ts}, where ts stands for time-shifted. The problem was reduced to solve the eigenproblem of correlation matrices $X_{ts}X^T$ and XX^T. The solution is found by solving the eigenvalue problem of the quotient matrix $Q = X_{ts}X^T(XX^T)^{-1}$ [48]. The delay time is estimated using autocorrelation differences.

4.2.4 Lattice Independent Component Analysis (LICA)

Lattice Independent Component Analysis is based on the Lattice Independence discovered when dealing with noise robustness in Morphological Associative Memories [112], later renamed Lattice Associative Memories introduced in Apendix C. Works on finding lattice independent sources (aka endmembers) for linear unmixing started on hyperspectral image processing [43, 115]. Since then, it has been also proposed for functional MRI analysis [40, 44] or mobile robot location [135] among others.

Under the Linear Mixing Model (LMM) the design matrix is composed of endmembers which define a convex region covering the measured data. The linear coefficients are known as fractional abundance coefficients that give the contribution of each endmember to the observed data:

$$\mathbf{y} = \sum_{i=1}^{M} a_i\mathbf{s}_i + \mathbf{w} = \mathbf{S}\mathbf{a} + \mathbf{w}, \tag{4.20}$$

where $\mathbf{y}$ is the d-dimension measured vector, $\mathbf{S}$ is the $d \times M$ matrix whose columns are the d-dimension endmembers $\mathbf{s}_i, i = 1,..,M$, $\mathbf{a}$ is the M-dimension abundance vector, and $\mathbf{w}$ is the d-dimension additive observation noise vector. Under this generative model, two constraints on the abundance coefficients hold. First, to be physically meaningful, all abundance coefficients must be non-negative $a_i \geq 0, i = 1,..,M$, because the negative contribution is not possible in the physical sense. Second, to account for the entire composition, they must be fully additive $\sum_{i=1}^{M} a_i = 1$. As a side effect, there is a saturation condition $a_i \leq 1, i = 1,..,M$, because no isolate endmember can account for more than the observed material. From a geometrical point of view, these restrictions mean that we expect the endmembers in $\mathbf{S}$ to be an Affine Independent set of points, and that the convex region defined by them covers *all* the data points.

The *Lattice Independent Component Analysis* (LICA) approach assumes the LMM as expressed in equation 4.20. Moreover, the equivalence between Affine Independence and Strong Lattice Independence [109] is used to induce from the data the endmembers that compose the matrix $\mathbf{S}$. Briefly, LICA consists of two steps:

1. Use an Endmember Induction Algorithm (EIA) to induce from the data a set of Strongly Lattice Independent vectors. In our works we use the algorithm described in [43, 40]. These vectors are taken as a set of affine independent vectors that forms the matrix $\mathbf{S}$ of equation 4.20.

2. Apply the Least Squares or Full Constrained Least Squares estimation to obtain the abundance vector of the LMM.

The advantages of this approach are (1) that we are not imposing statistical assumptions to find the sources, (2) that the algorithm is one-pass and very fast because it only uses lattice operators and addition, (3) that it is unsupervised and incremental, and (4) that it can be tuned to detect the number of endmembers by adjusting a noise-filtering related parameter. When $M \ll d$ the computation of the abundance coefficients can be interpreted as a dimension reduction transformation, or a feature extraction process.

Our input is a matrix of face images in the form of column vectors. In the linear mixing model (LMM), we represent the a face image as a linear combination of endmember faces. The weight of each endmember face (abundance) is proportional to its fractional contribution to the construction of the observed face image. In other words, the induced SLI vectors (endmembers) are selected face images which define the convex polytope covering the data. A face image is defined as a $A_{a \times b}$ matrix composed by $a \cdot b = N$ pixels. Images are stored like row-vectors. Therefore, column-wise the data-set is denoted by $Y = \left\{ \mathbf{y}_j; j = 1, \ldots, N \right\} \in \mathbb{R}^{n \times N}$

Algorithm 6 LICA feature extraction for face recognition. $E^{\#}$ denotes the pseudo-inverse of the matrix E.

```
1. Build a training face image matrix $X_{TR} = \left\{\mathbf{x}_j; j = 1, \dots, m\right\} \in \mathbb{R}^{N \times m}$. The testing image matrix is denoted $X_{TE} = \left\{\mathbf{x}_j; j = 1, \dots, m/3\right\} \in \mathbb{R}^{N \times m/3}$.
2. Obtain a set of $k$ endmembers using an EIA over $X_{TR}$: $E = \left\{\mathbf{e}_j; j = 1, \dots, k\right\}$ from $X_{TR}$. Varying EIA parameters will give different $E$ matrices. The algorithm has been tested with $\alpha$ values dependant on database size.
3. Unmix train and test data: $Y_{TR} = E^{\#} X_{TR}^T$ and $Y_{TE} = E^{\#} X_{TE}^T$.
```

, where each $\mathbf{y}_j$ is a pixel vector. Firstly, the set of SLI $X = \{\mathbf{x}_1\} \in \mathbb{R}^{n \times K}$ is initialized with the maximum norm pixel (vector) in the input data-set Y. We chose to use the maximum norm vector as it showed experimentally to be the most successful approach. The method is summarized in algorithm 6.

The algorithm for endmember induction, the EIA, used is the one in [43] which has tolerance parameter α controlling the amount of endmembers detected. In the ensuing experiments we have varied this parameter in order to obtain varying numbers of endmembers on the same data. Further explanations on Lattice Computing theory are available in appendix C.

4.3 Classification

One of the goals of this work is to compare the performance of Extreme Learning Machines (ELM) with other classifiers. Details on ELMs are reported in Appendix D. In the experiments in this Chapter basic ELM [64] and feature mapping or regularized ELM (ELM-FM) [63] were used. We have chosen two competing state of the art classification algorithms. One is an ensemble classifier based on decision trees - Random Forest [14]. The other is a Support Vector Machine variant introduced in [120] called $\nu-$SMV. We have used the implementations of Random Forest and $\nu-$SMV provided in Weka [46, 21]. In the following subsections, we describe the classifiers in more detail. Additionally, we have also compared ELM with Feed-forward Neural Networks (FFNNs) trained with two standard learning algorithms as provided in Matlab.

4.3.1 Random Forest

A random forest is a classifier consisting of a collection of tree-structured classifiers $h(x, \Theta_k), k = 1, \dots$ where the Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x

[14]. Random Forest select inputs randomly. This randomness is chosen so that the correlation between two different members of the forest is minimized. A Random Tree is formed by selecting at random, at each node, a small group of input variables to split on. In our case, this number was set to $log_2 a + 1$, where a is the number of attributes. The tree grows using CART methodology to maximum size. Trees are not pruned.

4.3.2 Support Vector Machines

Support Vector Machines (SVMs) are linear or non-linear (with a kernel trick) non-probabilistic binary classifiers [27]. The class of SVM that we have used was introduced in [120]. When it is a regression method we call it SVR, when it is a classifier it's called SVC. The main idea behind SVMs is to build a hyperplane that best separates members of different classes. Let be $(x_1, y_1) \ldots (x_l, y_l)$, our two-class labeled data set. It is said to be linearly separable if there exists a vector w and scalar b so that for all the elements of the training set

$$y_i(w \cdot x_i + b) \geq 1. \tag{4.21}$$

In the $\nu - \text{SVM}$ classification algorithm [120, 119], the optimization problem presented is to minimize

$$\tau(w, \xi, \rho) = \frac{1}{2} \|w\|^2 + \nu\rho \sum_{i=1}^{l} \xi_i \tag{4.22}$$

where $\|w\|^2$ is a term that characterizes the model complexity, the ξ are some variables and ν and ρ are two constants. This function is subject to the constraints

$$y_i\left((x_i w) + b\right) \geq \rho - \xi_i \tag{4.23}$$

$$\xi_i \geq 0 \,,\, \rho \geq 0. \tag{4.24}$$

The decision function, defining α_i that are $0 \leq \alpha_i \leq \frac{1}{l}$, and using a kernel k, takes the form

$$f(x) = \text{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i k(x, x_i) + b\right). \tag{4.25}$$

SVMs are binary classifiers, so we use one-against-one approach for multi-class classification. Details on the computation of b and ρ and justification of the preference of $\nu - \text{SVM}$ over classic SVM are thoughtfully explained on [120].

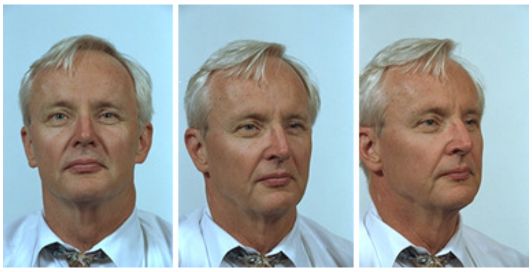

Figure 4.1: Example of the rotation that we allowed. Images from Color FERET database [96].

4.4 Experimental design

We have performed two separate but related experiments. The goal was to obtain answers to two questions about ELMs:

1. Used as a preprocessing step for ELMs, is LICA a better than or comparable to other state-of-the-art feature extraction algorithms when dealing with big, unbalanced face databases? and
2. Can ELMs outperform state-of-the art classifiers in such experimental environment?

We based our experimental designs on the Color FERET database [96, 95]. Color FERET contains 10344 face images, varying in scale, rotation and lighting. There are also occlusions caused by glasses or hair. Some of the images are grayscale, but the vast majority are RGB. We chose frontal and mildly rotated images - with a rotation of 15 ,22.5 and 45 degrees. Representative face image samples can be seen in figure 4.1. This left us with 5175 facial photo candidates to build our experimental databases. Classes correspond to subject identities. These databases have a highly unbalanced class size distribution, as is illustrated in figure where we plot a histogram of the number of samples per class in the first selected database. Following the detection process described below, we made three additional face image subset selections, resulting in four experimental databases of 5169, 3249, 832 and 347 images respectively. Table 4.1 shows a summary of each database's main features.

The faces were not suitable for recognition, because of the noise produced by different backgrounds and the differences in scale. Therefore, we used the detection algorithm developed in [136, 80] and available in Scilab SIVP. The algorithm usually detects several faces in a photography of a single subject. We added a face selection process based firstly on candidate's size. A second step checked if in the middle row's average color composition the red channel was predominant. This

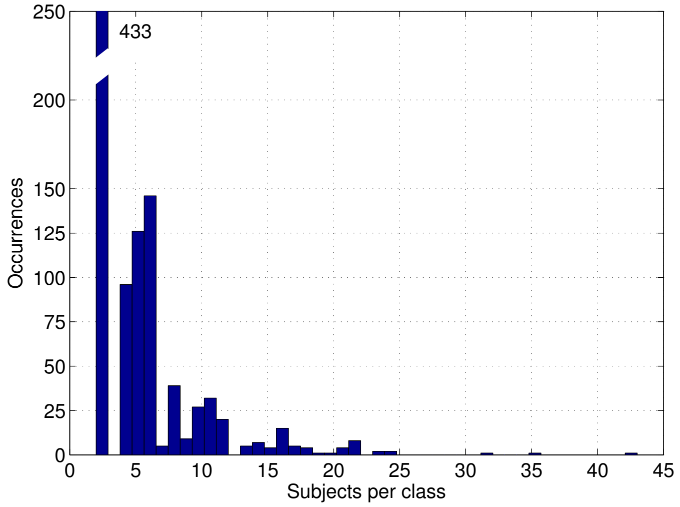

Figure 4.2: Histogram showing the class distribution of the DB 1 database.

	DB 1	DB 2	DB 3	DB 4
Number of samples	5169	3249	832	347
Number of classes	994	635	265	79
Mean (samples per class)	4.3924	3.1396	5.2835	5.2002
Standard deviation (samples per class)	5.8560	3.4498	4.9904	4.5012
Median (samples per class)	2	2	4	4
Mode (samples per class)	2	2	2	2

Table 4.1: Summary of the 4 databases used in our experiments.

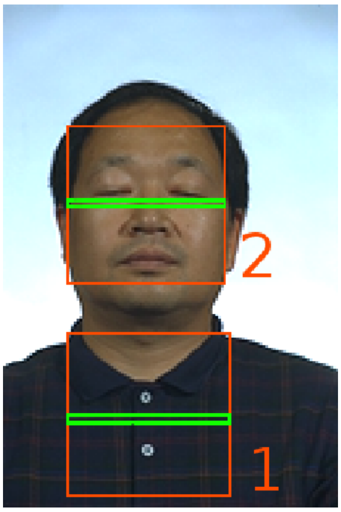

Figure 4.3: Detection example. Orange squares show the first and second candidates. First candidate's middle row's RGB values are R=41.95 G=41.97 B=46.60. Second candidate's are R=133.03 G=106.84 U=79.49.

method works well under average lighting conditions and regardless of skin color. We did not modify the face area selected by the algorithm. We allowed a partial occlusion of the faces, up to a 20% of the face area. There were 18 detection failures. We also removed 6 detected faces because the provided ground-truth deviated from reality. Overall, this method achieved a success rate of 99.65%. The process is illustrated in figure 4.3. The next step was to scale images to 100x100 pixels using bicubic resampling. Then we needed to do a conversion from RGB to grayscale prior to feature extraction. We used a $Gr = 0.85 \cdot R + 0.10 \cdot G + 0.05 \cdot B$ conversion method which is reported to be the optimal grayscale conversion formula for face recognition [24].

Feature extraction was performed using the algorithms mentioned on section 4.2. PCA has no parameter whatsoever. LDA usually needs a previous dimension reduction phase. We performed Singular Value Decomposition (SVD) over the data retaining the maximum amount of eigenvectors. Both ICA Infomax and ICA-MS also require a the same preprocess. Mean-field ICA has several parameters, like prior mixing matrix, noise covariance, etc. We found that constant mixing matrix and noise covariance, as well as power law tail source prior. This method showed empirically the best results in a reasonable time.

Classifiers were also empirically tuned. The parameter of the ELMs was the number of hidden nodes, in addition to the ridge parameter λ in the case of ELM-FM. Random Forest only required to fix the number of trees. In the case of SVMs, we chose $\nu -$SVM because it showed better recognition rate that C-SVM. The

ν parameter was also set empirically. Both the $\nu-$SVM kernel function and the ELM activation function were sigmoidal. We also tested two FFNNs with Backpropagation algorithm. One uses Resilient Backpropagation Algorithm (RPROP) [104] and the other Scaled Conjugate Gradient Algorithm (SCG) [89]. The five classifiers were tested with the four experimental databases described above, tuning their parameters to obtain the best accuracy possible. We performed 2-fold cross-validation. The recognition results are obtained based on 20 repetitions. In other words, in each of the 20 trials we randomly choose the 50% of the members of each class, having both testing and training set a similar size (not equal, because some classes contain an odd number of images).

4.5 Experimental results

Experiments were run on a Intel i5 2400 processor and 8 GB of RAM memory. Random Forest is resource greedy, and it's performance is limited by the amount of trees that computer's memory allows to grow. Other classification and feature extraction processes do not pose any computational resource-related problem. The following two subsections describe the results obtained, each corresponding to one of the two questions raised earlier in the section 4.4.

4.5.1 Results of LICA using Extreme Learning Machines

The computational experiments covered systematic dimensionality reduction up to 86, 107, 32 and 21 dimensions for databases DB 1, DB 2, DB 3 and DB 4, respectively. Working with dimensions above those limits did not show any increase in the accuracy of the algorithms. For ICA and PCA selecting the target dimension reduction was immediately accomplished selecting the desired sources and eigenvectors, respectively. For LICA that exploration implies varying the value of the α parameter and observing the number of endmembers detected. All feature extraction methods were evaluated in a wrapper scheme using an ELM for classification. The average number of hidden nodes was 1290 for DB 1, 870 for DB 2, 275 for DB 3 and 142 for DB 4. Extended information about the parameters of the classifiers are provided in table 4.2.

Figure 4.5 shows the recognition rate for the smallest database DB 4. The database has high average number of images per class (5.2002) with a standard deviation of 4.5012. Most classes have 2 samples. The results show that LDA and PCA converge quickly to their maximum hit-rate. This small database with high class size variability seems to be unsuitable for some ICA methods, such as the Mean field ICA and the M&S ICA. Although showing worst results than LDA in 0 to 5 dimension space, LICA based classification obtains the best recognition

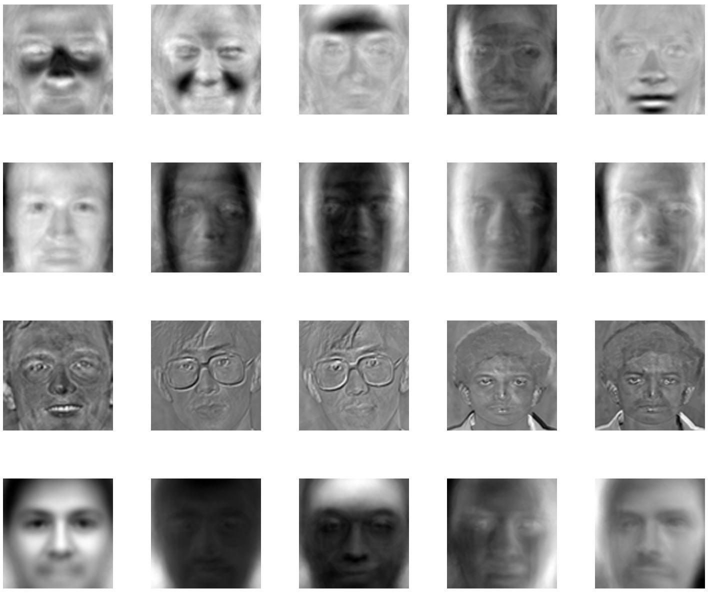

Figure 4.4: An instance of the first 5 independent components (ICA Infomax and ICA MS), endmembers (LICA) and eigenvectors (PCA)

Classifier	*Parameter*	*DB1*	*DB2*	*DB3*	*DB4*
ELM	hidden nodes	1290	870	275	142
ELM-FM	λ	2800	7900	4700	8400
$\nu - \text{SVM}$	ν	0.08	0.085	0.115	0.08
	cost, kernel degree, ε	1, 3, 0.001			
Random Forest	number of trees	60	60	86	208
	nF, max. tree depth	14, unlimited			
FFNN RPROP	mG, δi, i_0, epochs, Mf, lr, Δd ,Δmax, g	$5 \cdot 10^{-6}$, 1.005, 0.015, 100, 5, 0.01, 0.5, 50, 0			
	min. performance gradient	$1 \cdot 10^{-8}$	$1 \cdot 10^{-7}$	$1 \cdot 10^{-8}$	$1 \cdot 10^{-8}$
FFNN SCG	change in weight for second derivative approximation	0.5	0.05	0.5	0.5
	regulation the indefiniteness of the Hessian	0.005	0.0005	0.005	0.005
	epochs, Mf, g	100, 5, 0			

Table 4.2: Summary of the main parameters of the classifiers in our experiments. ε is the tolerance of the termination criterion, nF is the number of randomly chosen attributes, mG is the minimum performance gradient, Δi is the increment to weight change, i_0 is the initial weight change, Mf is the maximum validation fails, lr is the learning rate, Δd is the decrement to weight change, Δmax is the maximum weight change and g is the performance goal.

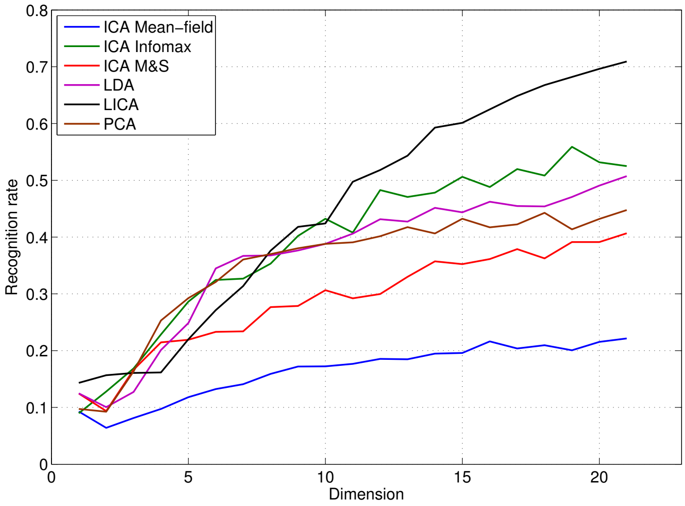

Figure 4.5: ELM recognition rate on DB 4 (347 subjects).

rate for dimensions above 5. Notice that LDA is a supervised dimension reduction algorithm, so that the remaining algorithms have a strong handicap against LDA. Figure 4.6 provides the recognition results for the next bigger database DB 3. LICA is also the best feature extraction algorithm in this case, improving PCA and LDA. The ICA algorithms perform badly in this database. The change from DB 3 to DB 2, as shown in table 4.1, lies in the addition of much more classes with few samples. This makes the DB 2 database even more unbalanced and complex than DB 3 and DB 4. The performance of all feature extraction algorithms drops heavily. Nevertheless, LICA continues to offer the best results, followed by LDA, as seen in figure 4.7. The change from DB 2 to DB 1 is different. DB 1 has many more subjects with more than two samples, thus rising both the sample-per-class mean and standard deviation. The most sensitive algorithm to the cited change is LICA. While the other methods see a 10-20% drop in their hit-rate at most, LICA drops about a 40%. The most efficient algorithms when testing DB 1 are LICA and LDA, as shown on figure 4.8. We must remind the reader that LDA is a supervised feature extraction method, while LICA is unsupervised. The main conclusion of this collection of computational experiments is that LICA-ELM outperforms the remaining feature extraction algorithms.

4.5.2 Results of ELM compared to other classifiers

In order to evaluate the resilience of ELMs to unbalanced datasets such as those in face recognition problems, we extracted the LICA features from all the databases

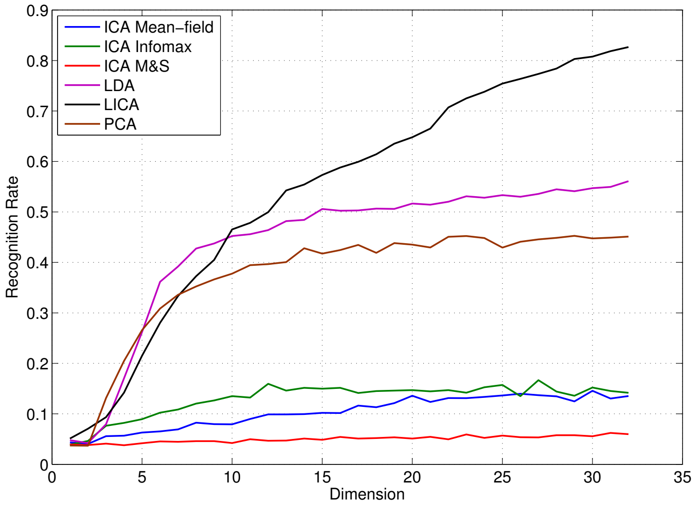

Figure 4.6: ELM recognition rate on DB 3 (832 subjects).

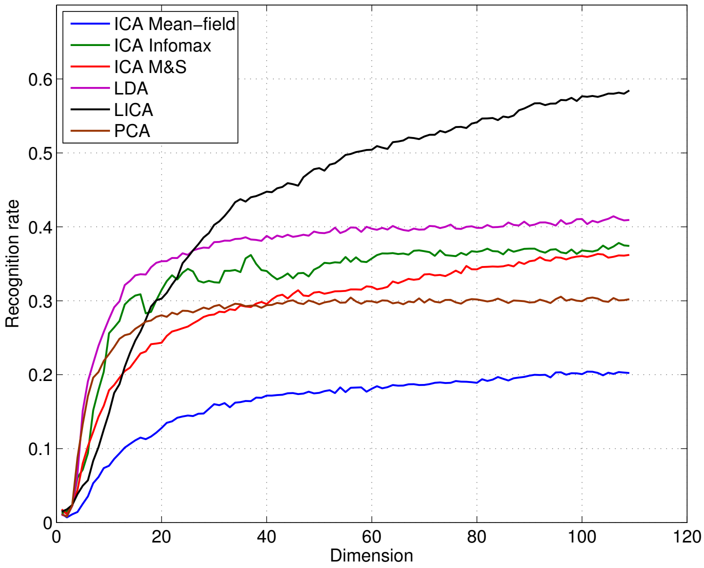

Figure 4.7: ELM recognition rate on DB 2 (3249 subjects).

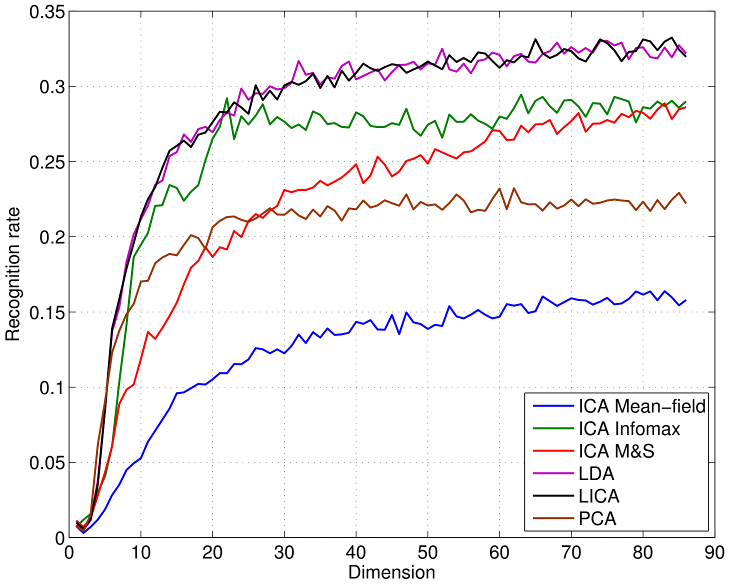

Figure 4.8: ELM recognition rate on DB 1 (5169 subjects).

and tested the five classifiers described in section 4.4. Other algorithms Naive-Bayes, Multinomial Naive-Bayes, Radial Basis Function Networks or Multilayer Perceptrons were discarded after pilot experiments on the DB1 database that resulted in very low recognition (below 1%). The recognition results are summarized in Table 4.3. We report the mean and standard deviation test accuracy over the databases, for all LICA feature dimensions.

The figure 4.9 plots the obtained results. The FFNNs, Random Forest and $\nu - \text{SVM}$ obtain systematically decrease their accuracy results as the size of the database increases. When testing the two small databases, $\nu - \text{SVM}$ improves Random Forest. The FFNN SCG algorithm reports better results that FFNN RPROP.

	DB 4	DB 3	DB 2	DB 1
ELM [64]	0.7093 (0.0385)	0.8782 (0.0199)	0.5834 (0.0126)	0.4735 (0.0061)
ELM-FM [63]	0.9035 (0.0237)	0.8721 (0.0153)	0.5834 (0.0143)	0.4830 (0.0056)
Random Forest [14]	0.7719 (0.0100)	0.7506 (0.0489)	0.3457 (0.0135)	0.2431 (0.0126)
$\nu - \text{SVM}$ [120]	0.8713 (0.0012)	0.8509 (0.0334)	0.3572 (0.0148)	0.2111 (0.0094)
FFNN RPROP [104]	0.8494 (0.0217)	0.7800 (0.0201)	0.1448 (0.0084)	0.3719 (0.0228)
FFNN SCG [89]	0.8692 (0.0198)	0.8166 (0.0244)	0.1205 (0.0024)	0.2110 (0.0338)

Table 4.3: Testing accuracy average (variance) for 4 Color FERET database subsets on features computed by the LICA feature extraction algorithm.

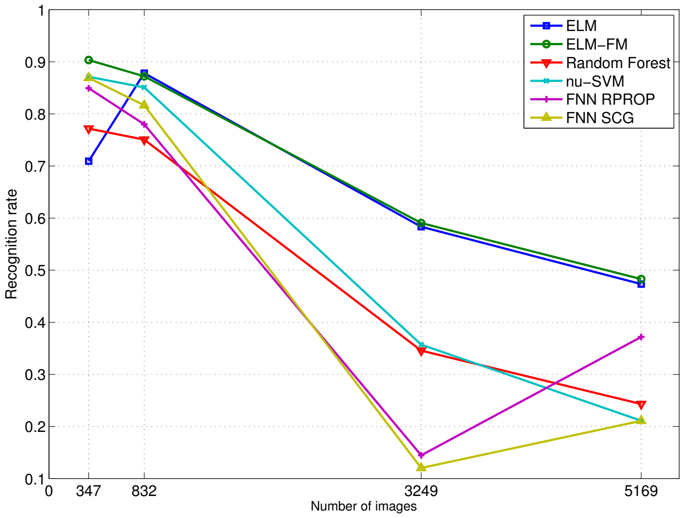

Figure 4.9: Recognition rate on the 4 databases using ELM, Randon Forest, $\nu-$SVM, FFNN BPROP and FFNN SCG on features extracted with LICA.

It is interesting that ELM obtains the worst accuracy result in the DB 4 case but the best one in the remaining databases. ELM-FM algorithm, adding a regularization method, overcomes this disadvantage. ELM-FM obtains the best results in the small database and similar results than those of ELM in the other databases. ELMs systematically are more robust against introducing more classes and samples while maintaining the samples per class ratio. The experiments with DB 2 and DB 1 represent a big rise on complexity and database size. ELM is the algorithm that best deals under these circumstances. Specially in the DB 1 scenario, where it doubles the other algorithm's recognition rate. It's also noticeable that standard FFNNs perform poorly in those big complex databases. Particularly, FFNN SCG seems unable to train properly DB 2 and DB 1.Besides, we can assert that ELM's total time of training and testing was several magnitudes smaller.

4.6 Discussion

We have applied LICA and five well known feature extraction procedures to recognize faces on four subsets of a well known face database. We have also tried ELM and two widely used classifiers. The databases on which the experiments have been performed were unbalanced, large and complex. We draw the following conclusions from the obtained results:

- LICA is a better feature extraction algorithm for face recognition under the mentioned circumstances. It shows a better recognition rate in conjunction with ELM classifier than the rest of methods. LICA also is less likely to drop its effectiveness when we use smaller databases with high subject to class ratio variability. ICA methods depend highly on the number of samples of the database. LDA's results are more consistent, specially when dealing with the biggest database and high subject per class standard deviation. Overall, Lattice Computing-based LICA algorithm its approach to feature extraction is effective, being more competitive with large unbalanced databases, such as those common in face recognition applications.

- The joint use of LICA and ELM has retrieved the best recognition results. We can suggest that Lattice-based Endmember Induction Algorithms could be best fitted to work with ELMs than other statistical tools (PCA, LDA) or independent component extraction algorithms (ICA Infomax, ICA M&S, Mean-field ICA).

- It is stated in [67] that Naive-Bayes is more robust to higher levels of class noise then Random Forest and C-SVM. However, we have found that when dealing with large unbalanced face databases, Naive-Bayes is far outperformed by ELM, $\nu-$SVM and Random Forest. The same applies to Multinomial Naive-Bayes, Radial Basis Function Networks or Multilayer Perceptrons. Results were so bad that they do not deserve publication here. There is no implementation bias as far as we applied the standard implementation found in Weka.

- Of all tested classifiers, ELM and ELM-FM are the most robust methods for large databases with high class-variation induced noise. It shows similar results than Random Forest or $\nu-$SVM when the databases are small. When the size is increased, ELM show an improvement of 124% and 95% over the results of $\nu-$SVM and Random Forest respectively. Furthermore, FFNNs with standard learning algorithms show worse performance than the rest of the classifiers. It is noteworthy that the regularization step added by ELM-FM to the basic ELM greatly increases the recognition accuracy in the smallest database.

The composition of LICA feature extraction and ELM classification show promising results in the domain of face recognition. More experiments over highly unbalanced databases could be performed on future works. It would also be valuable to test the various ELM algorithms apart from basic-ELM available in the literature [72, 62]. We think that it would be interesting to explore further the interplay between Lattice Computing-based feature extraction methods and Extreme Learning Machines.

Chapter 5

Feature Fusion Improving Face Recognition

Information fusion is a research area that has received a lot of attention lately. In the case of face recogntion, it is interesting to be able to combine methods that extract features differently, in order to better characerize the underlying nature of face images. This chapter proposes a fusion scheme of linear and lattice computing based features, introduced in Chapter 3, to improve face identification. The subject is introduced in Section 5.1. Section 5.3 proposes the feature fusion methodology. The experimental designed is explained in section 5.4. The results are shown and discussed in Section 5.5. Finally, some conclusions are exposed in Section 5.6.

5.1 Introduction

In machine learning approaches either statistical or biologically inspired, each image is represented as a vector identifying a point in a high dimensional space. The strong regularities in the face images induces to think that they are located in a low dimension manifold embedded in the high dimensional face image space. Therefore, a lot of effort has been addressed to define feature extraction processes which uncover this low dimensionality face space through linear and/or non-linear subspace projections. The goal is that the projected face images belonging to different classes occupy disjoint and compact regions in the feature space which can be easily separated by linear or non-linear discriminant functions. These projections are often defined by a collection of basis functions, so that face images are expressed as a linear combination of them. The basis functions can be found through linear or non-linear process. Linear approaches which have been applied to this problem in the literature are Principal Component Analysis (PCA) [133], Linear Discriminant Analysis (LDA) [151], or the Locality Preserving Projections (LPP) [50]. Chap-

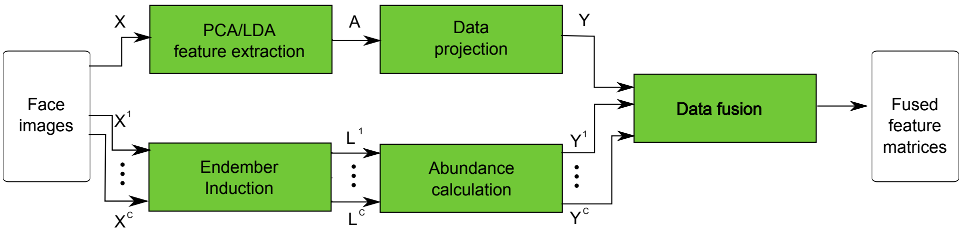

Figure 5.1: Flow diagram of the feature extraction and fusion process. We perform a linear feature extraction process (either PCA or LDA) over the whole input data. Concurrently, we extract class conditional endmembers and abundances. The last step performs feature fusion merging selected features computed in one or other process.

ter 3 shown that some lattice computing [38] approaches are successful instances of non-linear induction processes of basis functions from data [43, 40, 39]. The Lattice Independent Component Analysis (LICA) [39, 40] looks for lattice independent vectors in the data which are used to perform a linear unmixing of the data obtaining the data features as the fractional abundance coefficients of the data samples. Applying LICA to face recognition problems, we have found [83, 84] that LICA features provide classification accuracy comparable to Independent Component Analysis (ICA), LDA and PCA. The contribution of this Chapter is the fusion of LICA and linear features to obtain improved results in face identification experiments. LICA features and linear features are computed in independent processes. Linear features are appended to the LICA features to form the complete feature vector. When there is some ranking on the basis functions, such as the eigenvalues associated to the eigenfaces, the LICA features correspond to the high rank linear features, i.e. highest eigenvalues. Besides, we implement a class conditional LICA performing independent searches for lattice independent vectors in each class restricted dataset, obtaining class specific LICA features. We test our algorithm with four face databases, different test and training set sizes and different number of features. The fusion approach shows better overall results than traditional feature extraction methods.

5.2 Lattice-based feature extraction

5.3 Feature fusion

The whole process of feature fusion is illustrated by the flow diagram in figure 5.1. A face image $I_{a\times b}$ is a matrix composed of $a\cdot b = N$ pixels. To build the dataset we transform the images into 1D vectors $\mathbf{x}\in\mathbb{R}^N$ which compose row-wise the dataset matrix X:

$$X = \{\mathbf{x}_i^c; i = 1, \ldots, n; c \in \{1, 2, \ldots, C\}\} \in \mathbb{R}^{n \times N}, \tag{5.1}$$

where n is the number of face images, $\mathbf{x}_i^c$ is the i-th face vector which belongs to class c, and C the number of face classes (i.e. subjects). Let X^c be the submatrix corresponding to the set of face image samples belonging to class c:

$$X^c = \left\{\mathbf{x}_j^c \in X; j = 1, \ldots, M\right\} \in \mathbb{R}^{M \times N}, \tag{5.2}$$

where M is the number of face image samples belonging to that class. For each class c we compute a set of class conditional endmembers L^c applying an endmember induction algorithm (EIA), as described on section 5.2, to the class restricted dataset X^c. Its noise-related α parameter controls to what extent an endmember candidate can be regarded as different to an already chosen endmember. The number of induced endmembers will depend on the dataset and can be different for each subset within the same data base. They are used to calculate the abundance matrix of each class restricted dataset by straightforward unconstrained least squares (# denotes the matrix pseudo-inverse)

$$A^c = (L^c)^{\#} X^{cT}, \tag{5.3}$$

where $A^c = \{\mathbf{a}_i^c; i = 1, \ldots, M\} \in \mathbb{R}^{M_c \times M}$ are the class restricted abundance coefficients, and M_c the number of endmembers found for this class. On the other hand, the whole data set X is used to compute a mixing matrix W applying a linear feature extraction algorithms, such as PCA [133], 2DPCA [73], 2D2PCA [147], kernel PCA [118] or LDA [51]. The data features obtained by linear projection are given by

$$Y = WX^T \tag{5.4}$$

where $Y = \{\mathbf{y}_i^c; i = 1, \ldots, n; c \in \{1, 2, \ldots, C\}\} \in \mathbb{R}^{d \times n}$ is the feature matrix, when each face image is transformed into a feature vector $\mathbf{y}_i^c$ of dimensionality d. The final feature fusion step involves substituting the first linear features from Y with the corresponding abundances in A^c. Formally, the new i-th feature vector $\mathbf{z}_i^c \in \mathbb{R}^d$ of a face of class c is defined as

$$\mathbf{z}_i^c = \mathbf{a}_{j(i)}^c \| \left[y_{i,M_c+1}^c, \ldots, y_{i,d}^c\right], \tag{5.5}$$

where the operator $\|$ denotes the concatenation of vectors, appending the second vector to the end of the first one, the index $j(i)$ is the index in the set of class restricted data corresponding to the i-th vector in the dataset, and $y_{i,k}$ denotes the k-th component of the i-th feature vector. The final feature matrix collecting all the

Table 5.1: Summary characteristics of the used databases. *Variations in AT&T data base are less pronounced.

Name	Number of images	Number of subjects	Variations
AT&T Database of Faces[1]	400	40	Pose, expression, light*
MUCT Face Database [86]	3755	276	Pose, expression, light
PICS (Stirling) [2]	312	36	Pose, expression
Yale Face Database [10]	165	15	Expression, light, glasses

transformations of the face images is

$$Z = \{\mathbf{z}_i^c; i = 1, \ldots, n; c \in \{1, 2, \ldots, C\}\} \in \mathbb{R}^{n \times d}$$

5.4 Experimental design

We have performed our experiments on four public face databases, whose features are described in table 5.1. Images form Yalefaces database have a big white background area. We did a face detection pre-process to Yalefaces in order to remove the unnecessary background. The rest of the databases were left unchanged. Some details regarding the realization of the feature extraction processes:

- When dealing with small databases, LDA may encounter the so-called *small sample size problem* [36]. This problem arises when there are less samples than features. We have addressed this problem reducing the dimensionality of the data prior to the LDA algorithm. We have tested PCA, 2DPCA and 2D2PCA methods for this purpose.

- The EIA that we used for endmember extraction has an α parameter, as stated in section 5.3. We set this parameter manually to 3.9 for Yale databases, 3.5 for AT&T and MUCT databases and 10 for PICS database, obtaining sensible sets of endmembers.

- Values of features obtained with different extraction algorithms belong to different scales. In fact some of them differ in several orders of magnitude. To address this problem, we have performed the z-score normalization of the final feature vectors, subtracting the mean of the training data and dividing it by the standard deviation. The training data mean and standard deviation values are the ones used to normalize the testing data to avoid circularity effects.

The feature fusion approach has been tested building and validating Extreme Learning Machine (ELM) [64, 65] classifier. ELM is a fast, simple and effective single-layer feed-forward neural network learning scheme. Hidden node bias and input-to-hidden-unit weights are randomly chosen. Given a unit activation function, number of hidden nodes and targets, the output weights are given by a least-squares solution. ELMs have been successfully used for face recognition [84] and show promising performance compared to other classifiers [62, 63]. Further insight into ELMs is offered in appendix D.

We evaluated the feature fusion approach in a cross validation scheme where we have defined several random partitions of the data into train and test sets, specifically partitions where the training data correspond to 30%, 40%, 50%, 60% and 70% of the data have been applied. Training and testing samples were selected randomly without replacement. Each partition size was repeated 20 times. This variation of training data size is relevant to ascertain that classification accuracy does not depend only on the appropriate size of the training set. We performed more experiments using half of the samples for training and the other half for testing. We tested the performance of our approach across different feature space dimensionality. We tested 1 to 96 features in steps of 5.

5.5 Experimental results

Table 5.2 shows the average cross validation results for the ELM classifiers according to the random design described in the previous section, using 100 features. Bold values correspond to the winner in the comparison between LICA feature fusion and the conventional linear feature extraction. The results show that feature fusion enhances the recognition accuracy regardless of the train and test set sizes. Feature fusion obtained better results in 33 out of 35 cross validation experiments for the AT&T database. The accuracies shown in table 5.2 are pretty high, because AT&T is a simple and well balanced database, with very slight variations of pose, expression and lighting. The best standard method in this database is 2DPCA, with a 91.25% recognition rate when the testing samples are the 30%. The fusion with LICA features increases this accuracy to a 96.92%. The MUCT database is more difficult: The number of subjects *per* class is unbalanced and the images have notable pose, expression and illumination variations. The database is also the largest of the four. The recognition results are poor, as expected from the literature. Nevertheless, LICA feature fusion improves the results in each cross validation experiment. The best method in this database is 2DPCA+LDA, with

AT&T	PCA		2DPCA		2D2PCA		kernel PCA		PCA + LDA		2DPCA + LDA		2D2PCA + LDA	
test size	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused
30%	0.6842	**0.9583**	0.9125	**0.9692**	**0.5904**	0.5863	0.7396	**0.7929**	0.7229	**0.8879**	0.9071	**0.9300**	0.9083	**0.9254**
40%	0.6428	**0.9459**	0.9044	**0.9656**	0.5525	**0.5659**	0.7025	**0.7575**	0.6800	**0.8503**	0.8825	**0.9031**	0.8744	**0.9034**
50%	0.6013	**0.9185**	0.8875	**0.9492**	0.5187	**0.5225**	0.6765	**0.7338**	0.6273	**0.7910**	0.8515	**0.8592**	0.8505	**0.8557**
60%	0.5542	**0.8952**	0.8577	**0.9350**	**0.4806**	0.4706	0.6348	**0.6915**	0.5625	**0.7246**	0.8063	**0.8248**	0.7910	**0.8144**
70%	0.4998	**0.8389**	0.8046	**0.8927**	0.4109	**0.4177**	0.5832	**0.6332**	0.4632	**0.6279**	0.7095	**0.7446**	0.6961	**0.7257**
MUCT	PCA		2DPCA		2D2PCA		kernel PCA		PCA + LDA		2DPCA + LDA		2D2PCA + LDA	
test size	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused
30%	0.1171	**0.8643**	0.2369	**0.7574**	0.4375	**0.5183**	0.0554	**0.1819**	0.3069	**0.8999**	0.9214	**0.9606**	0.9192	**0.9601**
40%	0.1187	**0.8567**	0.2354	**0.7322**	0.4244	**0.4987**	0.0563	**0.1855**	0.3048	**0.8798**	0.8918	**0.9401**	0.8951	**0.9423**
50%	0.1185	**0.8402**	0.2176	**0.6880**	0.3798	**0.4669**	0.0545	**0.1770**	0.3041	**0.8427**	0.8467	**0.9140**	0.8444	**0.9157**
60%	0.1190	**0.8229**	0.2022	**0.6495**	0.3621	**0.4340**	0.0562	**0.1761**	0.3009	**0.8102**	0.8038	**0.8840**	0.8010	**0.8837**
70%	0.1230	**0.7711**	0.1719	**0.5595**	0.2945	**0.3656**	0.0540	**0.1592**	0.2815	**0.7349**	0.7072	**0.8147**	0.7063	**0.8173**
PICS	PCA		2DPCA		2D2PCA		kernel PCA		PCA + LDA		2DPCA + LDA		2D2PCA + LDA	
test size	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused
30%	0.1630	**0.8356**	0.6529	**0.7846**	0.3558	**0.3702**	0.5370	**0.5899**	0.1726	**0.6558**	**0.6514**	0.6447	**0.6875**	0.6822
40%	0.1672	**0.8018**	0.6109	**0.7471**	**0.3391**	0.3361	0.4854	**0.5507**	0.1613	**0.5931**	**0.5964**	0.5891	**0.6077**	0.5869
50%	0.1720	**0.7694**	0.5740	**0.6734**	0.3049	**0.3165**	0.4379	**0.4957**	0.2066	**0.5272**	**0.5049**	0.5040	**0.5280**	0.5263
60%	0.1831	**0.7529**	0.5506	**0.6843**	0.3009	**0.3220**	0.4417	**0.4826**	0.1937	**0.5303**	**0.5034**	0.4877	**0.5294**	0.4986
70%	0.2310	**0.7053**	0.4889	**0.6075**	0.2625	**0.2844**	0.3716	**0.4305**	0.1736	**0.4647**	0.4055	**0.4144**	**0.4293**	0.4269
Yalefaces	PCA		2DPCA		2D2PCA		kernel PCA		PCA + LDA		2DPCA + LDA		2D2PCA + LDA	
test size	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused	stand.	fused
30%	0.7989	**0.9744**	0.8111	**0.9267**	0.5700	**0.6667**	0.4889	**0.5122**	0.6744	**0.8833**	0.5933	**0.6756**	0.4611	**0.5600**
40%	0.8083	**0.9775**	0.7808	**0.9133**	0.5633	**0.6200**	0.4892	**0.5208**	0.6325	**0.8292**	0.5917	**0.6275**	0.4100	**0.5150**
50%	0.7250	**0.9511**	0.7256	**0.8889**	0.4811	**0.5444**	0.4611	**0.5228**	0.5456	**0.7039**	0.4844	**0.5339**	0.3239	**0.4283**
60%	0.6919	**0.9243**	0.7081	**0.8686**	0.4471	**0.5290**	0.4614	**0.5071**	0.4790	**0.6452**	0.4233	**0.4781**	0.2971	**0.3586**
70%	0.6650	**0.8929**	0.6842	**0.8279**	0.3854	**0.4792**	0.4313	**0.4933**	0.4200	**0.5300**	0.3625	**0.3950**	0.2571	**0.2987**

Table 5.2: Average recognition accuracy with ELM classifier using 100 features for a) AT&T, b) MUCT, c) PICS and d) Yalefaces databases. Bold numbers indicate the best method (standard vs fused with LICA). Asterisks indicate statistically significant differences using two-sample paired t-tests with a 0.05 significance level.

92.14% recognition accuracy. Feature fusion raises the accuracy to 96.06%. The PICS database lacks illumination variations, but includes frontal and profile photos as well as expression changes. The LICA feature fusion approach performs better than the standard approach in 25 out of 35 cross validation experiments. Notice that the results in the table for this database are very poor, maybe due to the curse of dimensionality. The Yalefaces results show that LICA feature fusion approach always introduces improvement. The best standard method is 2DPCA with 81.11% recognition accuracy. Enhancing 2DPCA with LICA features provides a 92.67% accuracy, but the fusion of LICA and PCA reaches a peak 97.44%. Overall, the experiments show that the fusion method achieves better recognition rates in 128 cases out of 140. Our fusion approach using LICA shows better performance than standard methods, regardless of the tests-training set sizes, extraction method used or the tested database.

In order to assess the statistical significance of the differences of recognition accuracy due to feature fusion, we perform paired t-tests between each linear feature extraction and its fusion with LICA based features using a 0.05 significance level. The LICA feature fusion approach wins in 126 cases and loses in only 6 cases. There are 8 draws. Most defeats occur in the PICS database with 2D2PCA or 2D2PCA+LDA methods.

We explore the performance of the methods along different feature dimensionality to assess how the pattern recognition algorithm improves or worsens when we consider more features. This allows us to obtain some useful information: How much time and memory space requires an algorithm to obtain acceptable results, the performance degradation induced by additional features due to the curse of dimensionality, and the consistency of the LICA feature fusion method improved performance. Figure 5.2, 5.3, 5.4 and 5.5 provide the plot of the classifier average accuracy for increasing feature vector dimensionality. Solid lines correspond to the fusion of LICA features and the linear features obtained with the legend named method. Dotted lines correspond to the features obtained from the bare linear methods. In most cases, the peak recognition accuracy is obtained with less than 50 features. In general, the effect of LICA feature fusion is twofold: (1) accuracy is increases, in some cases dramatically, and (2) the best performances are obtained with a lower number of features. For some linear feature extraction algorithms the Hughes effect, i.e. decrease of the algorithm's performance when adding new dimensions to the data, is very strong. This is most notable for 2D2PCA algorithm. The LICA feature fusion does not alleviate this effect for the worst cases.

In Table 5.2 we reported the worst results for the PICS database dropping the accuracy to near random choice results. The examination of the effect of the dimensionality increase for this database in figure 5.4 shows the general decrease of all methods for dimensionalities above 40, thus the results in Table 5.2 for 100

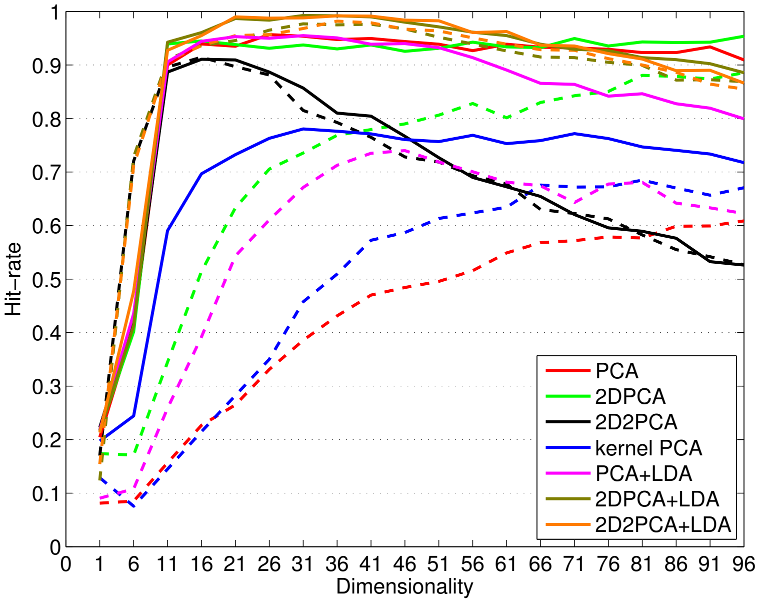

Figure 5.2: Recognition rate using ELM classifier for the AT&T database. Dotted lines correspond to standard feature extraction methods. Solid lines show the results of the proposed feature fusion approach.

features correspond to the worst scenario for this database. The most spectacular increase in accuracy is provided by the fusion of PCA and LICA features, however the 2DPCA+LDA and 2DPCA+LDA fused with LICA, provide the best recognition rates using few features. The nature of the kernel is always a relevant issue when using kernel PCA. We have tested different databases and used the same polynomial kernel. As we can see, results vary greatly.

5.6 Conclusion

This Chapter proposes the fusion of class conditional LICA features with linear features to obtain improved face identification results. Class conditional LICA computes the LICA features on class restricted data, obtaining a more accurate representation of the local data structure corresponding to each class. The fusion process aims to complement the descriptive power of LICA features with the conventional subspaces spanned by linear features. We have employed the same state-of-the-art classifier approach, the ELM, for the final classification avoiding any bias due to classification construction. The experimental results proved that our method enhances the performance of linear feature extraction algorithms specifically we have performed computational experiments involving PCA, 2D2PCA, 2D2PCA,

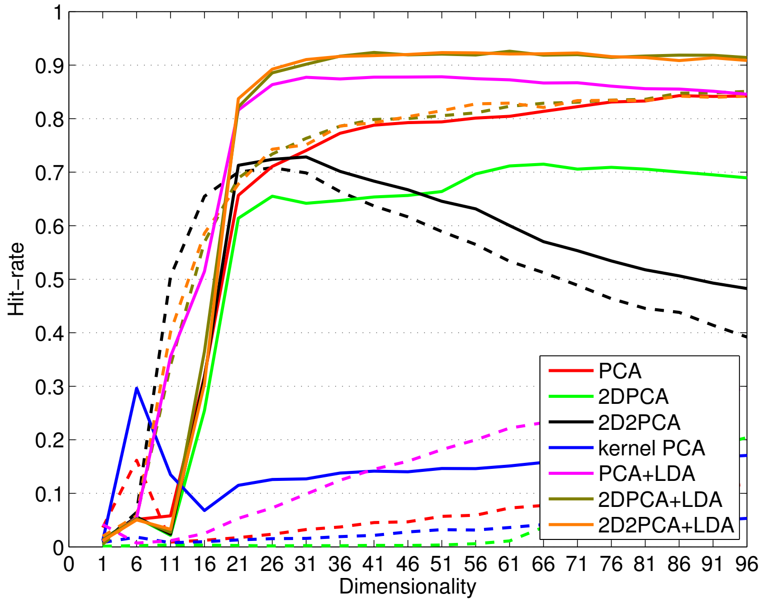

Figure 5.3: Recognition rate using ELM classifier for the MUCT database. Dotted lines correspond to standard feature extraction methods. Solid lines show the results of the proposed feature fusion approach.

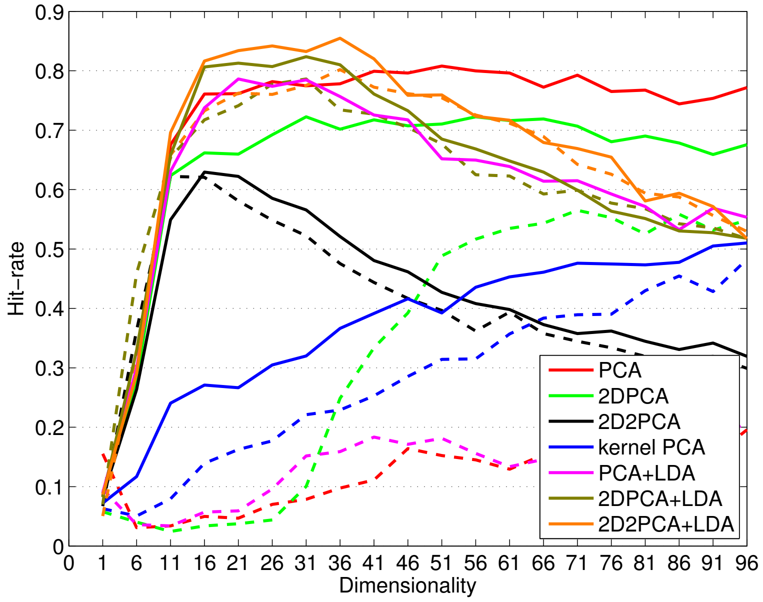

Figure 5.4: Recognition rate using ELM classifier for the PICS database. Dotted lines correspond to standard feature extraction methods. Solid lines show the results of the proposed feature fusion approach.

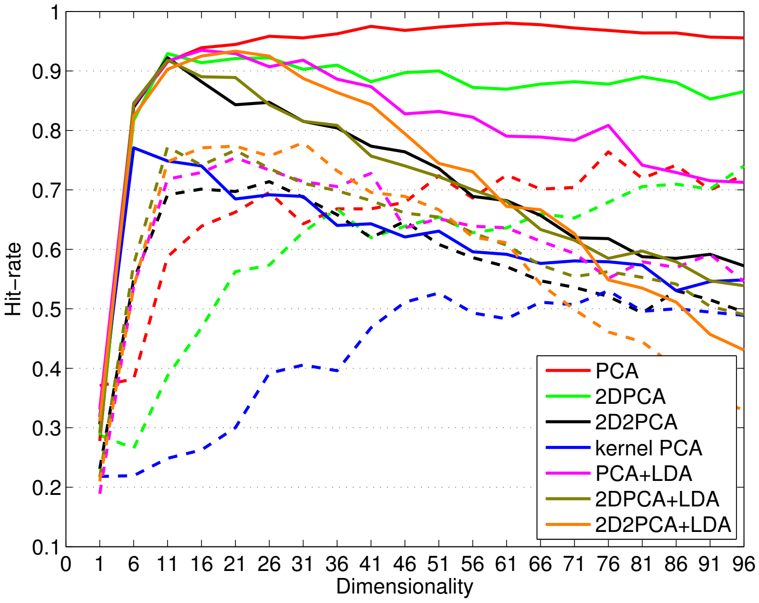

Figure 5.5: Recognition rate using ELM classifier for the Yalefaces database. Dotted lines correspond to standard feature extraction methods. Solid lines show the results of the proposed feature fusion approach.

kernel PCA and three variations of LDA. We performed the computational experiments on four face databases. The improved results are consistent under various settings: Different training and testing set sizes and different dimensionality. Overall, the LICA-fused 2D2PCA+LDA shows the best performance overall.

These experiments leave some open questions for further research. It would be interesting to study the influence of different Endmember Induction Algorithms and their parameters. We could also consider testing our approach with other state-of-the-art classification algorithms, such as classifier ensembles. It seems that fusing essentially different methods like LICA and PCA derivatives is a good approach towards better face recognition systems.

Appendix A

A Hyperspectral Image Database for Person Detection

Advances on computational methods have evolved around different applications and types of data. The experiments reported in the the chapters of this thesis devoted to skin detection in hyperspectral images have been performed over a very specific dataset. The generation and preprocess of said data is not a negligible part of the work done in the thesis.

The motivation for seeking to generate our own data comes from the lack of available hyperspectral image databases focused on person detection. The introduction of this work explained why is that skin detection is relevant, and the motives behind the use of imagery technology whose spectral resolution goes beyond classic RBG images. However, taking usable hyperspectral images depends not only on the availability of a hyperspectral camera and the knowledge of how to use it- desirable climatic conditions are necessary. We were fortunate to create an optimal framework for the capture and preprocess of the images that we needed: A proper research collaboration agreement with a research group that not only offered their full collaboration, but was geographically located in a convenient place. The data was collected in El Paso, Texas, United States of America by the author of this dissertation in collaboration with Miguel Velez-Reyes, Ph. D., Professor and Chair Electrical & Computer Engineering at the University of Texas at El Paso (UTEP) and his students Stephanie M. Sanchez and Mohammed Q. Alkhatib. The remaining of this appendix aims to explain the details concerning the dataset. The image capture technicalities are detailed in section A.1. Section A.2 explains normalization and smoothing processes.

A.1 Data collection

The hyperspectral data was obtained using a SOC710 camera. It has a spectral coverage of 400 to 1000 μm, a spectral resolution of 4.6875 μm and a dynamic range of 12 bits. It delivers 696 pixel per line images, and 128 bands. We set an integration time of 20 milliseconds and an electronic gain of 2 (twice the gain). Increasing the integration time compensates for poor lightning, but it also increases the capture time, which can produce smear. Increasing the integration time electronically can provide a good trade-off between data acquisition speed and getting a meaningful dynamic range. Obtaining one 696x520 image cube takes around 23.2 seconds. The images were collected in El Paso, Texas along the moths of April and May of 2014. The temperatures ranged between 21ºC and 31ªC, with completely clear skies. This condition avoided illumination variations due to passing clouds. Moreover, dry weather minimizes moisture and water, which can produce complex radiance responses. The collection time was around noon to minimize the influence of shadows. The only light source was the sun. This stable lighting conditions and the short image acquisition times allowed minimizing spectral variations due to illumination changes. However, in some cases wind moved background foliage branches, thus introducing some undesirable motion noise. Once established the ideal lighting conditions, we aimed to obtain images including the following elements:

- A person showing parts of exposed skin. The data is designed for experiments aiming to skin detection regardless of the body part, and especially not depending only on the presence of a face, which is usually the main person detection landmark. It is also interesting not to include the whole body in a position which could be seen as the ideal person detection setup.

- For the same reasons, we also included the presence of hair, glasses or other elements that might partially occlude the skin sections.

- Different subjects with different skin colors, preferably dark skinned. It is most difficult to detect dark skin in RGB images, as color information can hardly be taken advantage of. Therefore, it was desirable to include diverse skin tones, emphasizing in dark colors.

- The presence of small shadows, both on skin and non-skin surfaces. The shadows should have the size that allows us to analyze their effect on skin detection without covering all the skin.

- Backgrounds formed by man made objects, vegetation or both. Man made objects have distinctly different spectral signatures from vegetation, so both kind of backgrounds were included. Figure A.1 illustrates the difference of skin, cloth and vegetation spectra taken from image C4.

- White standard surface present in all images, in order to calculate spectral radiance, by normalization to true white.

We took images of three subjects with diverse color skins. The subjects were located in places with both artificial structures like concrete or rocks and natural backgrounds like grass or flowers. The images are named using this subject-background data scheme, as shown in table A.1. Note that skin color are assessed via the Fitzpatrick scale, Type III being sun sensitive light intermediate caucasian skin, Type IV minimally sun sensitive mediterranian/hispanic darker skin, and Type V darker sun insensitive skin. Lighter Types I and II and very dark Type VI were not present.

The validation experiments require a reference ground truth. This is a binary image selecting which pixels in the hyperspectral image are skin and which pixels correspond to other elements. This segmentation was performed manually. Figures A.2, A.3, A.4 , A.5, A.6, A.7 and A.8 show the false RGB image alongside the manual segmentation of each hyperspectral cube.

A.2 Data cube preprocessing

This section describes the preprocessing steps. Firstly, it states how the reflectance values were calculated. Then, the coordinate system change from Cartesian to Hyperspeherical is explained. Finally, a noise analyisis of the images is presented along with the smoothing filter employed to reduce noise effect.

A.2.1 Reflectance normalization

The sensor of a hyperspectral camera provides a fine resolution spectral sampling of the total radiance reflected by the surface. The empirical line (EL) approach to reflectance normalization assumes that within each image, there are at least two target areas of low and high reflectance for the spectral bands recorded by the sensor. We can use a single white target object as reference assuming surfaces of zero reflectance will produce zero radiance. The bright reference target can be a Lambertian surface, i.e. a surface whose radiance is isotropic; and we can assume that the bright target will have unitary reflectance for all the wavelengths. The reflectance of a single pixel is then computed over the full wavelength range at the spectral resolution of the camera as:

$$R = \frac{L_T}{L_R}, \tag{A.1}$$

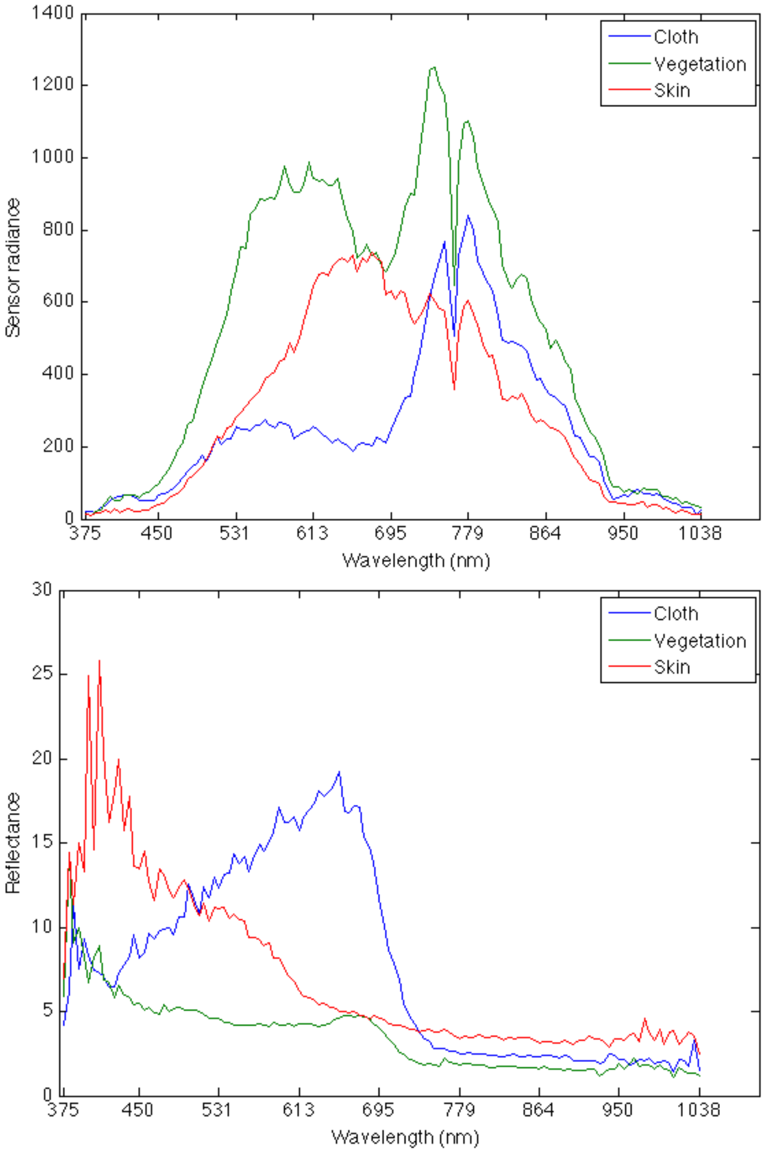

Figure A.1: Radiance and reflectance samples from image C4, corresponding to three pixels located in the pants of the subject, the left arm, and the bushes.

Name	Subject		Background		Notes
	Code	Skin Color	Code	Type	
A1	A	III	1	Vegetation, flowers, grass	Strong wind
B2	B	IV	2	Concrete	
A3	A	III	3	Grass	
C4	C	V	4	Trees, rocks	Light wind
C5	C	V	5	Concrete, metal	Light camera movement
C5b	C	V			
A5	A	III			Light camera movement

Table A.1: Hyperspectral image dataset summary table.

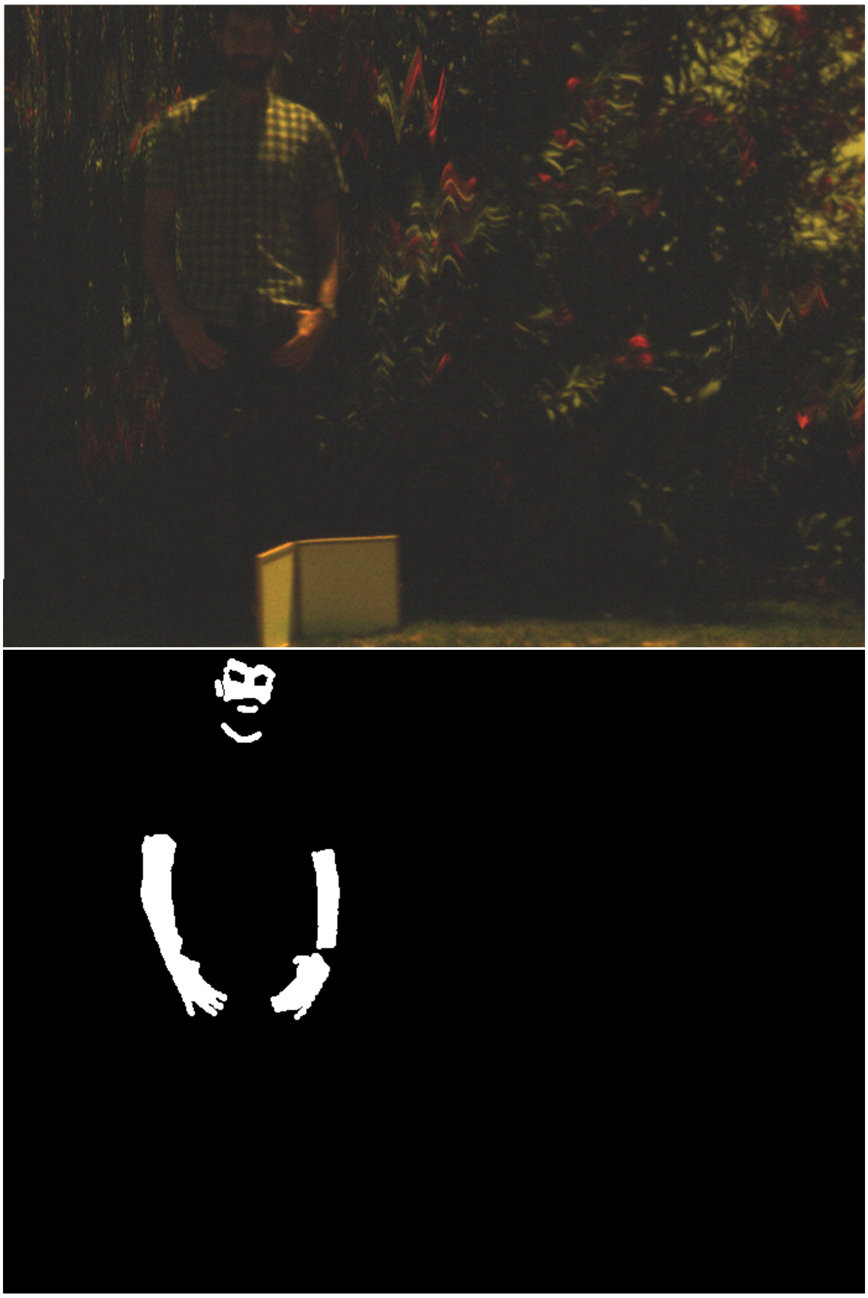

Figure A.2: False RGB composite and manual segmentation of image A1.

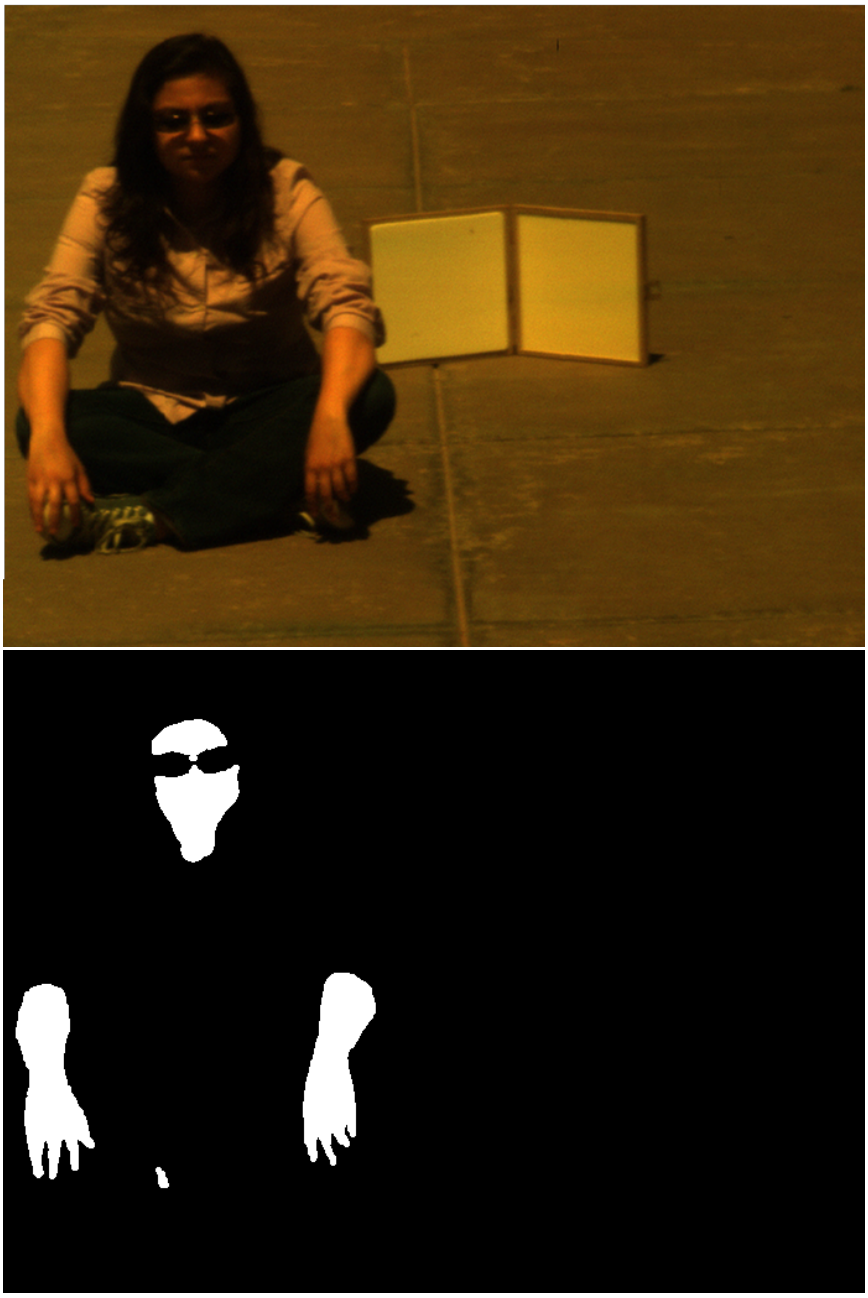

Figure A.3: False RGB composite and manual segmentation of image B2.

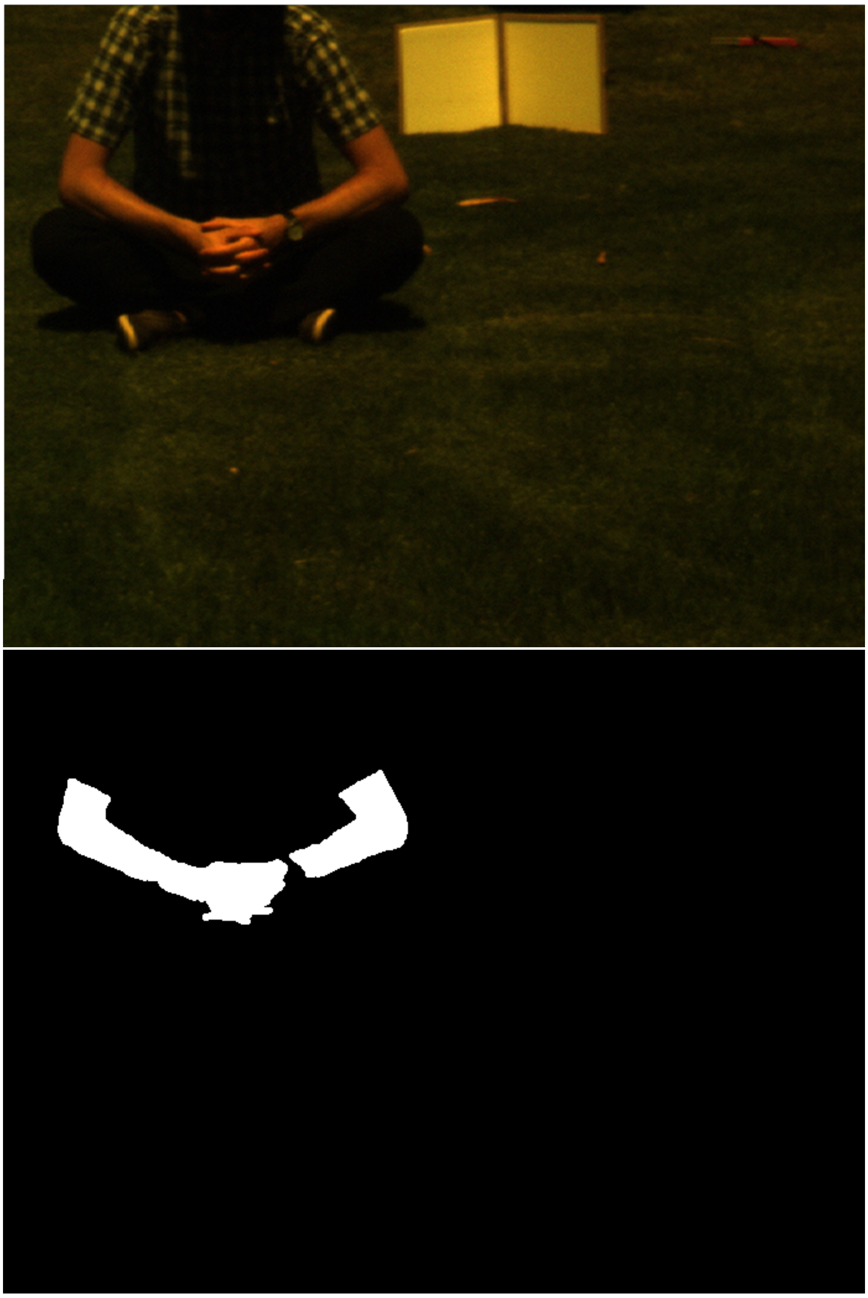

Figure A.4: False RGB composite and manual segmentation of image A3.

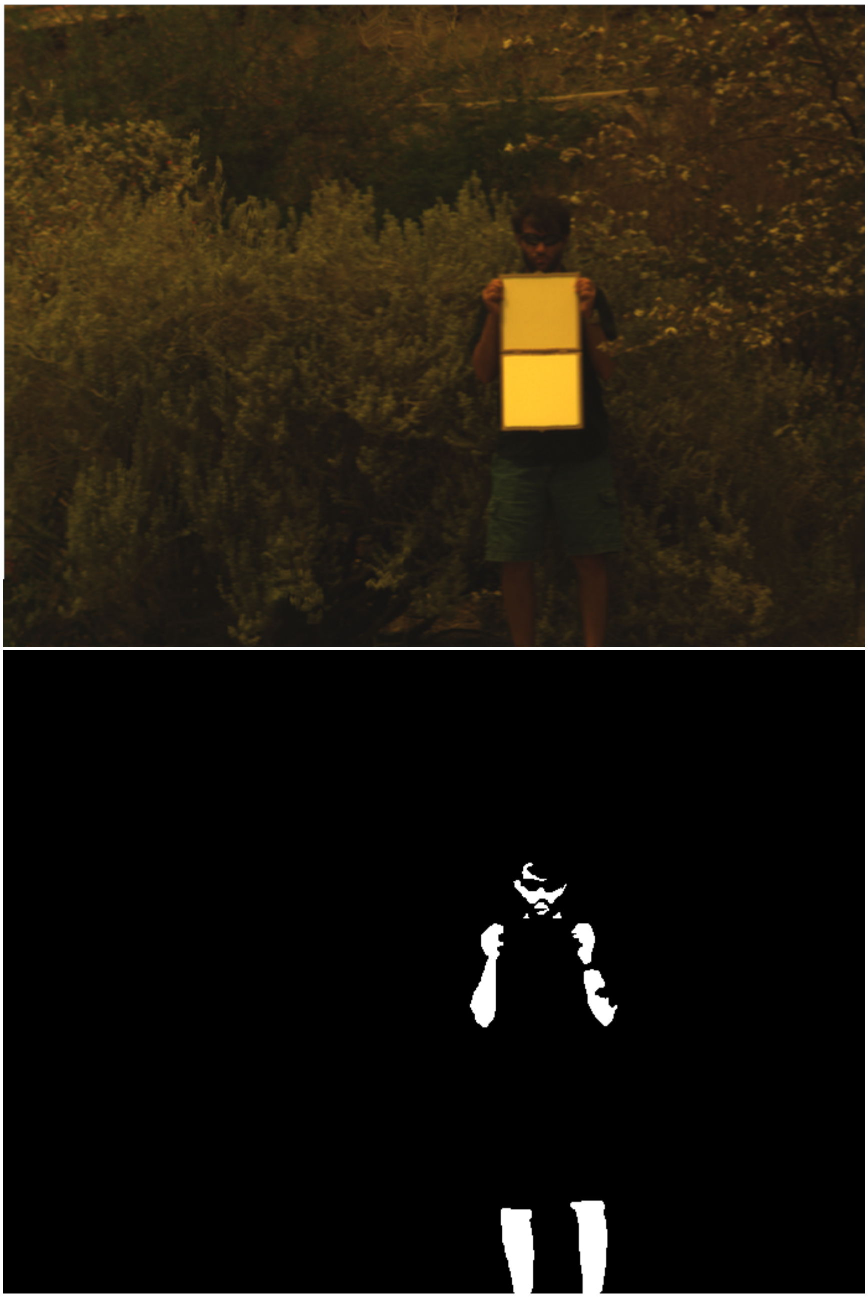

Figure A.5: False RGB composite and manual segmentation of image C4.

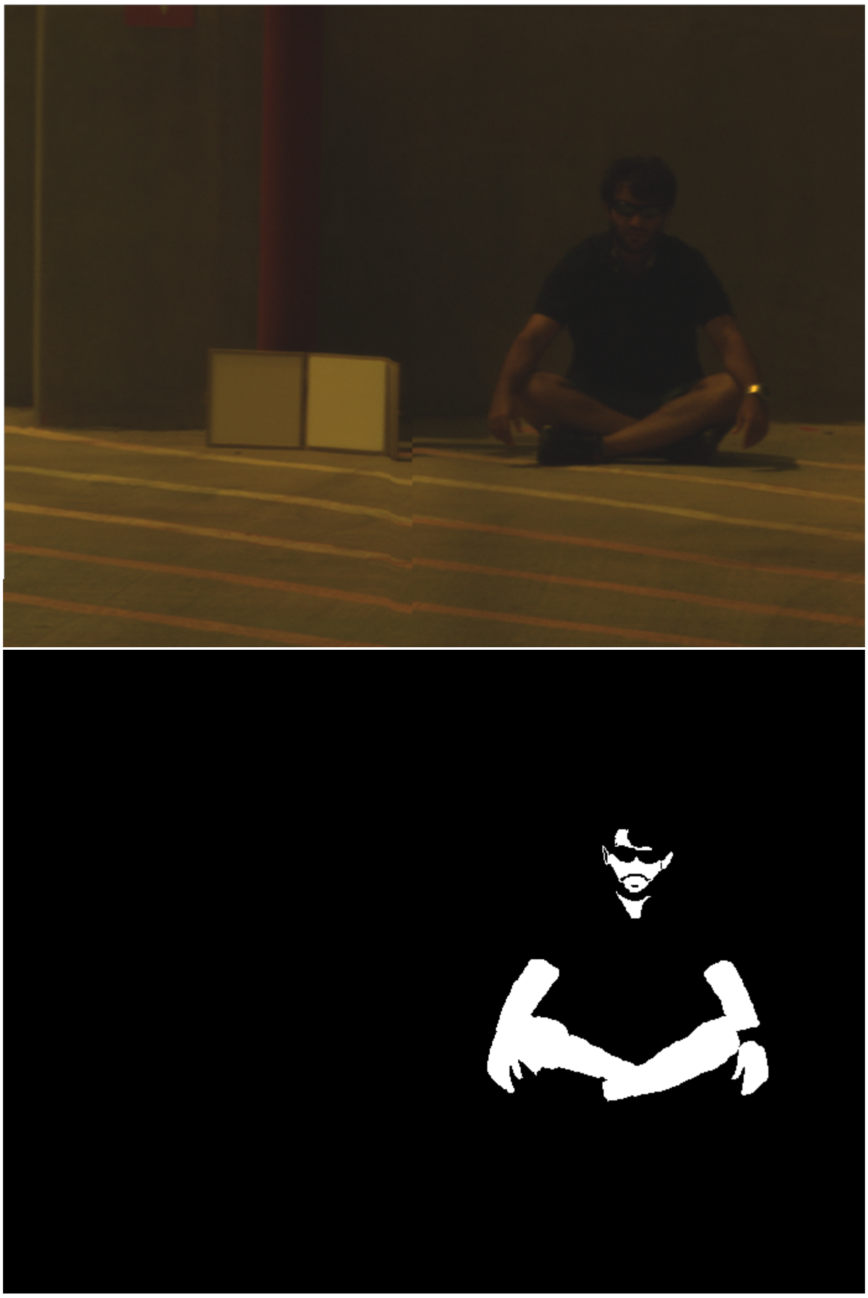

Figure A.6: False RGB composite and manual segmentation of image C5.

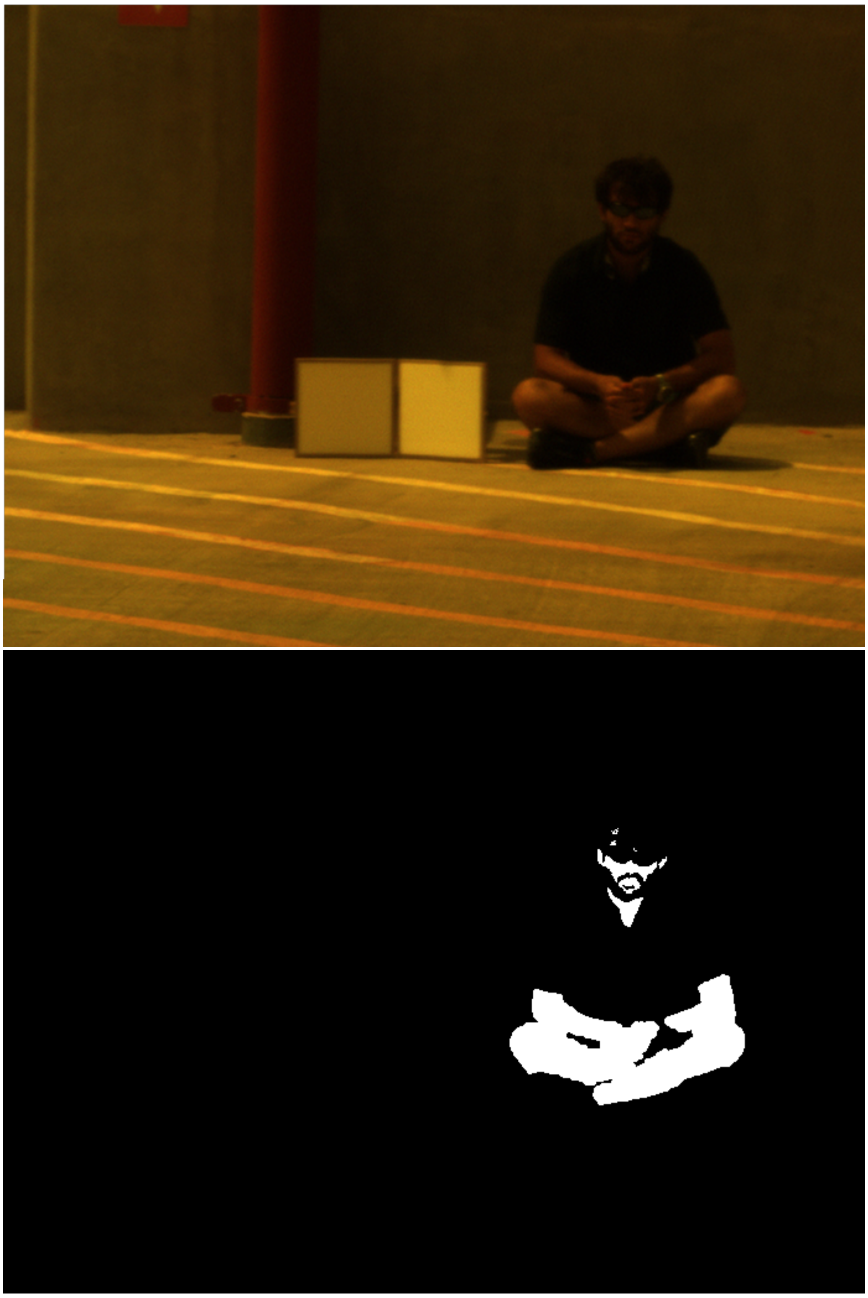

Figure A.7: False RGB composite and manual segmentation of image C5b.

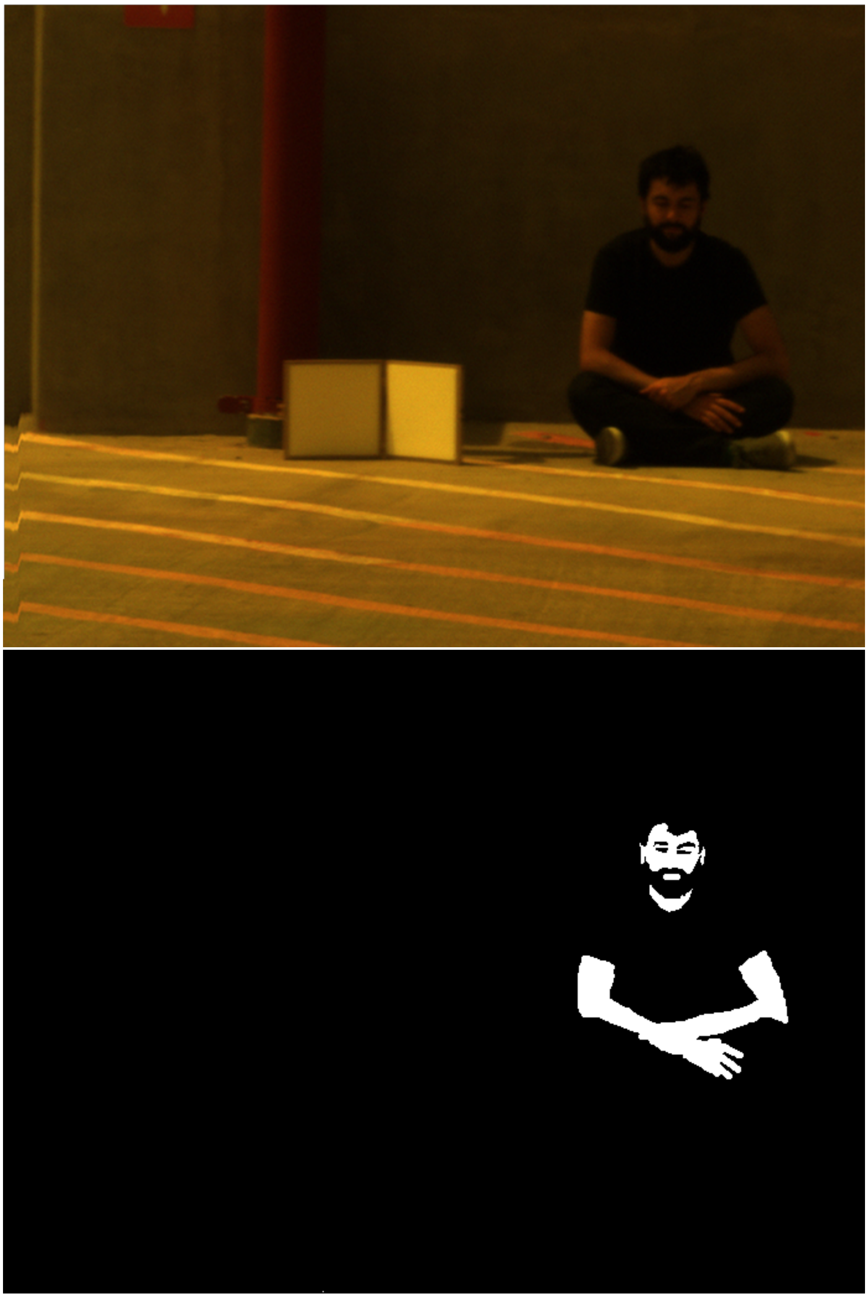

Figure A.8: False RGB composite and manual segmentation of image A5.

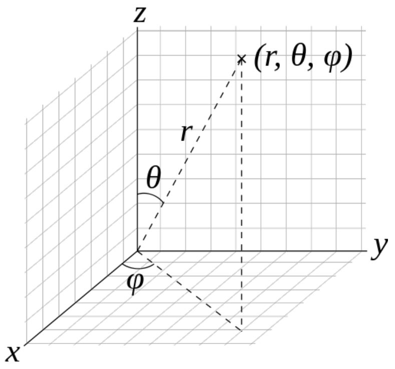

Figure A.9: Representation of a point in a 3-spherical coordinate system, given by radial distance r, azimuth angle φ and elevation angle θ.

where L_T is the radiance of the pixel and L_R is a reference radiance value. The images collected for this experiments were set up with a "white standard" surface as reference. We averaged the pixels pertaining to the surface to obtain a reference radiance value for each image. Then, the radiance of each pixel on each cube was calculated using equation A.1, hence each image has its own reference radiance value for normalization.

A.2.2 Hypershperical coordinates

We have explored other coordinate system besides the cartesian representation. The underlying idea it is to be able to remove the intensity component of the pixels, performing our experiments only on the chromatic features of the data. We can define a spherical coordinate system, akin to the euclidean system, in a three dimensional space. A point in a 3-spherical coordinate system is given by the radial distance r from the origin point, the azimuth and elevation angles, as seen in figure A.9.

This coordinate system can be expanded from a 3-dimensional space to a n-dimensional space. Any given point represented in an n-dimensional euclidean space by $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ may be represented in an n-dimensional spherical space by $\mathbf{x} = \{r, \phi_1, \phi_2, \ldots, \phi_{n-1}\}$, where r is the vector magnitude that gives the radial distance, and $\{\phi_1, \phi_2, \ldots, \phi_{n-1}\}$ are the angular parameters. This representation of a point image is named the hyperspherical coordinates of the point.

Given a hyperspectral image $X = \{(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{d})_\mathrm{i}\}_{i=1}^{n}$, where each pixel is a d-dimensional array, we can transform each pixel to its hyperspherical coordinates. The hyperspectral image will be denoted $X = \{(\mathrm{r}, \phi_1, \phi_2, \ldots, \phi_{\mathrm{d}-1})_\mathrm{i}\}_{i=1}^{n}$, where r is the intensity of each pixel and $\{\phi_1, \phi_2, \ldots, \phi_{d-1}\}$ its chromaticity representation. The transformation to hyperspectral coordinates is given by the following formula:

$$\begin{aligned} r &= \left(\textstyle\sum_{i=1}^{d} x_i^2\right)^{1/2} \\ \phi_i &= \cot^{-1} \frac{x_i}{\left(\sum_{j=i+1}^{d} x_j^2\right)^{1/2}}\ , \ j = 1, \ldots, d-2 \quad . \\ \phi_{d-1} &= 2 \cdot \cot^{-1} \frac{\left(x_{d-1}^2 + x_d^2\right)^{1/2} - x_{d-1}}{x_d} \end{aligned} \tag{A.2}$$

A.2.3 Noise analysis of the dataset

Besides this classic hyperspectral image preprocess steps, we performed some further computation over the images. Despite the good capture conditions, there are some noise inducing conditions, as stated above. Wind is a problem, inducing spatial noise. Camera calibration, sun reflections, dust and other factors can also induce noise. To estimate the incidence of noise, we use Signal to Noise Ratio (SNR). This measurement is calculated by the formula

$$\mathrm{SNR} = 10 \cdot \log 10 \left(\frac{\text{signal power}}{\text{signal noise}} \right). \tag{A.3}$$

It is usually given in decibels, but we are going to use a linear unit, which allows to asses the magnitude of the noise more intuitively. We are interested in seeing if the shape of a spectral signal found in a hyperspectral image differs from the normal signal shape that we expect. In order to achieve that, we will define the magnitude of the mean signal as the ideal signal power, and calculate the noise as the difference between that signal and the every pixel's signal in the data cube. Thus, the signal noise will correspond to the disturbances in the expected signal shape. Let's define a hyperspectral image as a collection of pixels $X = \{\mathbf{x}_\mathrm{i}\}_{i=1}^{n}$, where each pixel is an array of the form $\mathbf{x}_\mathrm{i} = \left\{x_{i,j}\right\}_{j=1}^{d}, x_{i,j} \in \mathbb{R}$. We propose to calculate the signal to noise ratio (SNR) of a pixel using the following formula:

$$\mathrm{SNR}\left(\mathbf{x}_i\right) = \frac{\sum_1^d \left|\left(\bar{x}_j\right)^2\right|}{\sum_1^d \left|\left(x_{i,j} - \bar{x}_j\right)^2\right|}, \tag{A.4}$$

where $\bar{\mathbf{x}} = \left\{\bar{x}_j\right\}_{j=1}^{d}$ is the mean pixel vector of X. This ratio effectively gives a noise proportion measure in relation to the desired signal, which in our case is the mean pixel of the image. In order to asses the total noise present in the image, we also compute the Aggregate Signal to Noise Ratio (ASNR), which accounts for the total noise present in the image. It is also interesting to calculate the Mean Signal to Noise Ratio (MSNR), that gives an idea of the average noise present in a pixel of an hyperspectral image. These two measurements are given by the following:

$$\begin{array}{r}\mathrm{ASNR}\,(X) = \sum_{i=1}^{n} \mathrm{SNR}\,(\mathbf{x}_i) \\ \mathrm{MSNR}\,(X) = \frac{\sum_{i=1}^{n} \mathrm{SNR}(\mathbf{x}_i)}{n}\end{array} \tag{A.5}$$

Finally, we calculate the Signal to Noise Ratio Standard Deviation (SNRSD), which gives an idea of the spatial uniformity of the noise, given by the formula

$$SNRSD\,(X) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(SNR\,(\mathbf{x}_i) - MSNR\,(X)\right)^2}. \tag{A.6}$$

The noise features calculated for the images of the collection are presented in table A.2.

A.2.4 Smoothing by RLWR

In order to reduce the noise, we applied a smoothing technique to all the images, the well known Robust Locally Weighted Regression smoothing [25]. It is a moving average local method because each smoothed value is determined by neighboring data points. The method uses a regression weight function using a linear polynomial. Additionally, in order to make the system resistant to outliers, a robust weight function is used. The method is as follows:

1. For each point $x \in X$, compute the regression weights $w_i \in W$:

   $$w_i = \left(1 - \left|\frac{x - x_i}{d(x)}\right|^3\right)^3,$$

   where x_i are the nearest neighbors of x and $d(x)$ is the distance in the abscissa from x to the most distant neighboring point.

2. Perform local regression using weighted least squares method.

3. Using the residuals r_i from the regression process, calculate the robust weights:

   $$\hat{w}_i = \begin{cases} \left(1 - \left(\frac{r_i}{6 \cdot \mathrm{Md}(|\mathrm{r}|)}\right)\right) & |r_i| < 6 \cdot \mathrm{Md}\,(|r|) \\ 0 & |r_i| \geq 6 \cdot \mathrm{Md}\,(|r|) \end{cases}$$

   where $\mathrm{Md}\,(|r|)$ is the median absolute deviation of the residuals.

4. Perform local regression using weighted least squares method with the robust weights.

5. Repeat steps 3 and 4 five times.

	Name	Raw images			Smoothed images		
		ASNR	MSNR	SNRSD	ASNR	MSNR	SNRSD
Cartesian coord.	A1	2.36E+06	6.53	29.04	3.24E+06	8.94	89.20
	B2	1.65E+07	45.67	1423.14	2.62E+07	72.27	4935.72
	A3	6.56E+06	18.13	219.00	1.09E+07	30.00	936.02
	C4	6.72E+06	18.58	1982.28	7.05E+06	19.48	2533.75
	C5	1.99E+06	5.50	48.09	2.16E+06	5.97	124.67
	C5b	1.93E+06	5.32	51.47	2.09E+06	5.79	128.04
	A5	1.97E+06	5.44	53.85	2.16E+06	5.96	141.19
Hyperspherical coord.	A1	1.51E+08	417.57	65888.29	8.94E+06	24.71	21.91
	B2	4.20E+08	1161.59	328190.31	1.09E+07	30.01	6.97
	A3	6.69E+08	1848.66	1410270.03	1.14E+07	31.51	13.67
	C4	1.47E+09	4074.36	11151807.28	1.26E+07	34.75	12.70
	C5	2.47E+08	682.06	131809.10	9.97E+06	27.55	9.96
	C5b	2.53E+08	697.97	137476.46	1.00E+07	27.66	9.72
	A5	2.48E+08	685.72	132097.62	9.98E+06	27.57	9.95

Table A.2: ASNR, MSNR and SNRSD values calculated for the hyperspectral images.

The effects of smoothing on the SNR are shown in table A.2. In the case of cartesian coordinates, it is clear how the ASNR and MSNR for the smooth images have higher values, meaning that the SNR is higher comparing to the non-smoothed data. Consequently, the smoothing process helps reducing the noise in the hyperspectral images. However, it is interesting to note that images B2 and C4 have big disparities in noise from pixel to pixel. That can be due to the lights and shadows in image B2 and the spatial noise produced by wind on image C4. Where and when the images are taken has great influence, which is noticeable in images C5, C5b and A5. The three of them have similar SNR features, and were taken on the same spot during a single session. However, converting the smoothed images to hyperspheric coordinates, as seen at the bottom half of table A.2, reduces considerably the signal to noise ratio. But, as we have removed the magnitude values and retained the hyperspheric angles, the validity of these measures can be subject of discussion. Three sample pixels of skin, concrete and cloth from image B2 are plotted in figure A.10. The graphs serve as a visual example of the smoothing process.

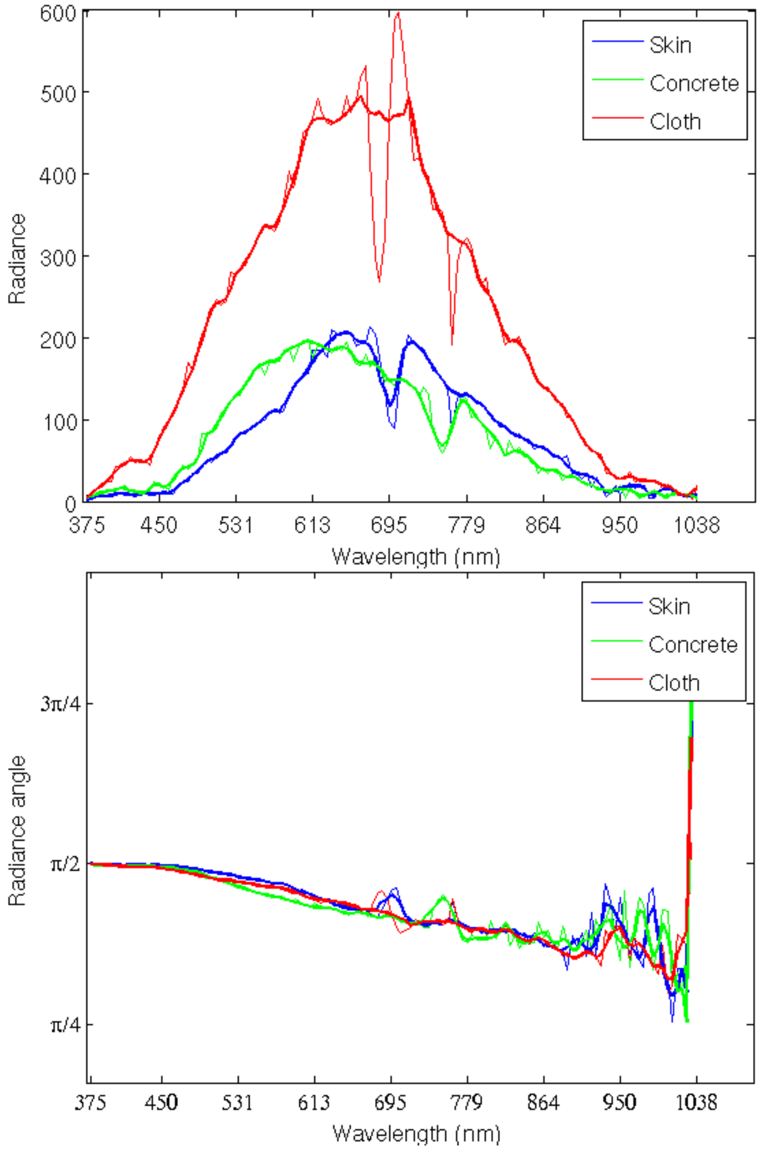

Figure A.10: Pixel examples of skin, concrete and cloth from image B2. Thin lines are the original radiance responses. Thick lines correspond to the smoothed pixels. The top image corresponds to cartesian coordinates and the bottom one corresponds to hyperspherical angle values.

Appendix B

Hyperspectral Imaging Methodology

This appendix exposes the methodological details of the experiments reported in this dissertation, concisely explaining the reasons behind the choice of the model validation technique, algorithm performance evaluating statistical measures and other methodological issues.

B.1 Segmentation performance evaluation

Skin detection in hyperspectral images can be stated as a binary classification problem. The dataset described in Appendix 1 is a badly balanced set of data -see TableB.1- the ratio of skin labeled samples is very low. The key measures will be those that indicate how well an algorithm can detect true positives, and how well it avoids false positives as well as false negatives. We can achieve that, as well as having an overall impression of the algorithms accuracy, by using the following three performance measures, explained in skin segmentation context:

- **Correct rate** (CR) or accuracy, given by the proportion of pixels correctly labeled as skin or not skin:

$$\mathrm{CR} = \frac{\mathrm{True\,Positives} + \mathrm{True\,Negatives}}{\mathrm{Total\,cases}}.$$

- **Precision** or positive predictive value, given by the proportion of true skin pixels among all the pixels classified as skin, in other words, the probability that a pixel labeled as skin is truly skin:

$$\mathrm{Precision} = \frac{\mathrm{True\,Positives}}{\mathrm{True\,Positives} + \mathrm{False\,Positives}}.$$

Image name	Number of pixels	Number of skin pixels	Percentage of skin pixels
A1	361920	7402	2.0879%
B2		15784	4.5601%
A3		10948	3.1193%
C4		5698	1.5996%
C5		11712	3.3443%
C5b		11236	3.2040%
A5		7351	2.0732%

Table B.1: Proportion of skin pixels on each image

- **Sensitivity** or true positive rate, which relates to the proportion of true skin pixels among all skin pixels on an image, that is the probability of labeling a skin pixel as skin:

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}.$$

The analysis of the experiments regard CR as a general indicator of the performance. Both Precision and Sensitivity measures are considered the most important indicators of the methods ability of skin detection.

B.2 Unmixing performance evaluation

Unmixing results can be assessed by several measures, like reconstruction signal to noise ratio, abundance root mean squared error or differences in spectral angel distance. The unmixing results shown in this work are evaluated with the following reconstruction error measures:

- Mean squared error (MSE) as reconstruction error:

  The quality of any unmixing process of equally sized images can be assessed by the sum of the Mean Squared Error (MSE) of the original hyperspectral image $\mathbf{X}$ with respect to the reconstructed one. Having the estimated endmembers $\hat{\mathbf{E}}$ and abundances $\hat{\alpha}$, the reconstructed signal will be $\hat{\mathbf{X}} = \hat{\mathbf{E}}\hat{\alpha}$. The MSE of a given image $\mathbf{X}$ is calculated as follows:

$$\varepsilon_{\text{MSE}}\left(\mathbf{X}, \hat{\mathbf{X}}\right) = \frac{\left\|\mathbf{X} - \hat{\mathbf{X}}\right\|^2}{d}. \tag{B.1}$$

  where d is the number of pixels.

- Mean absolute error (MAE) reconstruction error:

MSE can be sensible to outliers, therefore the need to also calculate the mean absolute reconstruction error (MAE):

$$\varepsilon_{\text{MAE}}\left(\mathbf{X},\hat{\mathbf{X}}\right)=\frac{\left|\mathbf{X}-\hat{\mathbf{X}}\right|}{d}. \tag{B.2}$$

- Mean angular distance (MAD) to the skin signatures:

  The quality of a proposed skin signature is measured calculating its average cosine distance to all the skin pixels. The cosine distance between two pixels $\mathbf{x}_i$ and $\mathbf{x}_j$ is given by:

$$d_{\text{COS}}\left(\mathbf{x}_i,\mathbf{x}_j\right)=1-\frac{\mathbf{x}_i\mathbf{x}_j^T}{\sqrt{\left(\mathbf{x}_i\mathbf{x}_i^T\right)\left(\mathbf{x}_j\mathbf{x}_j^T\right)}}. \tag{B.3}$$

  Then, the mean angular error of two images is given by:

$$\varepsilon_{\text{MCOS}}\left(\mathbf{X},\hat{\mathbf{X}}\right)=\frac{\sum_{i=1}^{N}d_{\text{COS}}\left(\mathbf{x}_i,\hat{\mathbf{x}}_i\right)}{N}. \tag{B.4}$$

  The rationale behind MAD as an error measurement is that, for a given pixel, the obtained endmembers and corresponding abundances should enable reconstructing said pixel with a minimum deviation from the original one. Given the signal nature of the data, angular distance is an appropriate measurement of this deviation. Averaging the distance for all the pixels gives us a reconstruction accuracy measure different from the classic MSE and MAE measurements.

Appendix C

Fundamentals of Lattice Computing

This Appendix gathers some definitions and results that are the theoretical background of lattice computing based endmember induction algorithms.

The work on Lattice Associative Memories (LAMs) stems from the consideration of the bounded lattice ordered group $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +')$ as the alternative to the algebraic framework $(\mathbb{R}_{\pm\infty}, +, \cdot)$ for the definition of Neural Networks computation [105, 106], where $\mathbb{R}_{\pm\infty} = \mathbb{R} \cup \{-\infty, +\infty\}$ is the set of extended real numbers, the operators $\vee$ and $\wedge$ respectively denote the discrete max and min operators (sup and inf in a continuous setting), and $+$,$+'$ respectively denote addition and its dual operation such that $x +' y = y + x$, $\forall x \in \mathbb{R}$, $\forall y \in \mathbb{R}_{\pm\infty}$; $\infty +' (-\infty) = \infty = (-\infty) +' \infty$ and $\infty + (-\infty) = -\infty = (-\infty) + \infty$. Thus, the additive conjugate is given by $x^* = -x$.

Given a set of input/output pairs of patterns $(\mathbf{X}, \mathbf{Y}) = \left\{ \left(\mathbf{x}^{\xi}, \mathbf{y}^{\xi} \right); \xi = 1, .., k \right\}$, where $\mathbf{X} = \left\{ \mathbf{x}^1, \dots, \mathbf{x}^k \right\} \subset \mathbb{R}^n$ and $\mathbf{Y} = \left\{ \mathbf{y}^1, \dots, \mathbf{y}^k \right\} \subset \mathbb{R}^m$ are two finite sets of pattern vectors, a linear hetero-associative neural network based on the pattern's cross correlation [59] is built up as $\mathbf{W} = \sum_{\xi} \mathbf{y}^{\xi} \cdot \left(\mathbf{x}^{\xi} \right)'$. Mimicking this constructive procedure authors in [105, 106] proposed the following constructions of LAMs (denoted in those works as Morphological Associative Memories), the min memory $\mathbf{W}_{XY}$ and the max memory $\mathbf{M}_{XY}$, both of size $m \times n$:

$$\mathbf{W}_{XY} = \bigwedge_{\xi=1}^{k} \left[\mathbf{y}^{\xi} \times \left(-\mathbf{x}^{\xi} \right)' \right] \text{ and } \mathbf{M}_{XY} = \bigvee_{\xi=1}^{k} \left[\mathbf{y}^{\xi} \times \left(-\mathbf{x}^{\xi} \right)' \right], \tag{C.1}$$

where $\times$ is any of the $\boxed{\vee}$ or $\boxed{\wedge}$ operators, reducing the notational burden since $\mathbf{y}^{\xi} \boxed{\vee} \left(-\mathbf{x}^{\xi} \right)' = \mathbf{y}^{\xi} \boxed{\wedge} \left(-\mathbf{x}^{\xi} \right)'$. Here $\boxed{\vee}$ and $\boxed{\wedge}$ denote the max and min matrix product, respectively defined as follows:

$$C = A \boxed{\vee} B = [c_{ij}] \Leftrightarrow c_{ij} = \bigvee_{k=1..n} \left\{ a_{ik} + b_{kj} \right\}, \tag{C.2}$$

$$C = A \boxed{\wedge} B = [c_{ij}] \Leftrightarrow c_{ij} = \bigwedge_{k=1..n} \left\{ a_{ik} + b_{kj} \right\}. \tag{C.3}$$

If $\mathbf{X} = \mathbf{Y}$ then $\mathbf{W}_{XX}$ and $\mathbf{M}_{XX}$ are called Lattice Auto-Associative Memories (LAAMs):

$$\mathbf{W}_{XX} = \bigwedge_{\xi=1}^{k} \left[\mathbf{x}^{\xi} \times \left(-\mathbf{x}^{\xi} \right)' \right] \text{ and } \mathbf{M}_{XX} = \bigvee_{\xi=1}^{k} \left[\mathbf{x}^{\xi} \times \left(-\mathbf{x}^{\xi} \right)' \right], \tag{C.4}$$

LAAMs present some interesting properties: perfect recall for an unlimited number of stored patterns and convergence in one step for any input pattern. The following theorems [105, 106] prove these properties:

Theorem 1. $\mathbf{W}_{XX} \boxed{\vee} \mathbf{X} = \mathbf{X} = \mathbf{M}_{XX} \boxed{\wedge} \mathbf{X}$.
$\mathbf{W}_{XX} \boxed{\vee} \mathbf{x} = \mathbf{x}$ *iff* $\mathbf{M}_{XX} \boxed{\wedge} \mathbf{x} = \mathbf{x}$.
If $\mathbf{W}_{XX} \boxed{\vee} \mathbf{z} = \mathbf{v}$ *and* $\mathbf{M}_{XX} \boxed{\vee} \mathbf{z} = \mathbf{u}$, *then* $\mathbf{W}_{XX} \boxed{\vee} \mathbf{v} = \mathbf{v}$ *and* $\mathbf{M}_{XX} \boxed{\wedge} \mathbf{u} = \mathbf{u}$.

Definition 2. A vector $\mathbf{x} \in \mathbb{R}^n_{\pm\infty}$ is called a fixed point or stable state of $\mathbf{W}_{XX}$ iff $\mathbf{W}_{XX} \boxed{\vee} \mathbf{x} = \mathbf{x}$. Similarly, $\mathbf{x}$ is a fixed point of $\mathbf{M}_{XX}$ iff $\mathbf{M}_{XX} \boxed{\wedge} \mathbf{x} = \mathbf{x}$.

Theorem 1 establishes that $\mathbf{W}_{XX}$ and $\mathbf{M}_{XX}$ are perfect recall memories for any number of uncorrupted input vectors, theorem 1 implies one step convergence, and theorem 1 says that $\mathbf{W}_{XX}$ and $\mathbf{M}_{XX}$ share the same set of fixed points represented by $\mathscr{F}(\mathbf{X})$ [127, 111].

Theorem 3. *For every* $\mathbf{x} \in \mathbb{R}^n_{\pm\infty}$, *we have* $\mathbf{W}_{XX} \boxed{\vee} \mathbf{x} = \hat{\mathbf{x}}$ *and* $\mathbf{M}_{XX} \boxed{\wedge} \mathbf{x} = \check{\mathbf{x}}$, *where* $\hat{\mathbf{x}}$ *denotes the supremum of* $\mathbf{x}$ *in the set of fixed points of* $\mathbf{W}_{XX}$, *and* $\check{\mathbf{x}}$ *denotes the infimum of* $\mathbf{x}$ *in the set of fixed points of* $\mathbf{M}_{XX}$.

LAAMs are extremely robust to erosive or dilative noise, but not to the presence of both. A distorted version $\tilde{\mathbf{x}}^{\gamma}$ of the pattern $\mathbf{x}^{\gamma}$ has undergone an erosive change whenever $\tilde{\mathbf{x}}^{\gamma} \leq \mathbf{x}^{\gamma}$, and a dilative change when $\tilde{\mathbf{x}}^{\gamma} \geq \mathbf{x}^{\gamma}$. Particularly, the erosive LAAM, $\mathbf{W}_{XX}$, is extremely robust to erosive changes, while the dilative LAAM, $\mathbf{M}_{XX}$, is so to dilative changes. Research on robust recall [105, 127, 128, 111, 102, 107] based on the so-called kernel patterns lead to the notion of Lattice Independence (LI) and the recall exact description in terms of the LAMs fixed points and their basis of attractions. The definition of Lattice Independence is closely tied to the study of the LAAM fixed points when they are interpreted as lattice transformations, as stated by the following theorems:

Definition 4. Given a set of vectors $\left\{\mathbf{x}^1,...,\mathbf{x}^k\right\} \subset \mathbb{R}^n$ a *linear minimax combination* of vectors from this set is any vector $\mathbf{x} \in \mathbb{R}^n_{\pm\infty}$ which is a *linear minimax sum* of these vectors:

$$x = \mathscr{L}\left(\mathbf{x}^1,...,\mathbf{x}^k\right) = \bigvee_{j\in J}\bigwedge_{\xi=1}^{k}\left(a_{\xi j} + \mathbf{x}^{\xi}\right),$$

where J is a finite set of indexes and $a_{\xi j} \in \mathbb{R}_{\pm\infty}\ \forall j \in J$ and $\forall \xi = 1,...,k$.

The *linear minimax span* of vectors $\left\{\mathbf{x}^1,...,\mathbf{x}^k\right\} = \mathbf{X} \subset \mathbb{R}^n$ is the set of all linear minimax sums of subsets of $\mathbf{X}$, denoted $LMS\left(\mathbf{x}^1,...,\mathbf{x}^k\right)$.

Given a set of vectors $\mathbf{X} = \left\{\mathbf{x}^1,...,\mathbf{x}^k\right\} \subset \mathbb{R}^n$, a vector $\mathbf{x} \in \mathbb{R}^n_{\pm\infty}$ is *lattice dependent* iff $x \in LMS\left(\mathbf{x}^1,...,\mathbf{x}^k\right)$. The vector $\mathbf{x}$ is *lattice independent* iff it is not lattice dependent on $\mathbf{X}$. The set $\mathbf{X}$ is said to be *lattice independent* iff $\forall\lambda \in \{1,...,k\}$, $\mathbf{x}^{\lambda}$ is lattice independent of $\mathbf{X}\setminus\left\{\mathbf{x}^{\lambda}\right\} = \left\{\mathbf{x}^{\xi} \in \mathbf{X} : \xi \neq \lambda\right\}$.

The definition of lattice independence supersedes the early definitions of erosive and dilative morphological independence.

Theorem 5. *Given a set of vectors* $\mathbf{X} = \left\{\mathbf{x}^1,...,\mathbf{x}^k\right\} \subset \mathbb{R}^n$, *a vector* $\mathbf{y} \in \mathbb{R}^n_{\pm\infty}$ *is a fixed point of* $\mathscr{F}(\mathbf{X})$*, that is* $\mathbf{W}_{XX} \boxvee \mathbf{y} = \mathbf{y} = \mathbf{M}_{XX} \boxwedge \mathbf{y}$*, iff* $\mathbf{y}$ *is lattice dependent on* $\mathbf{X}$.

Definition 6. A set of vectors $\mathbf{X} = \left\{\mathbf{x}^1,...,\mathbf{x}^k\right\} \subset \mathbb{R}^n$ is said to be *max dominant* if and only if for every $\lambda \in \{1,...,k\}$ there exists and index $j_{\lambda} \in \{1,...,n\}$ such that

$$x^{\lambda}_{j_{\lambda}} - x^{\lambda}_{i} = \bigvee_{\xi=1}^{k}\left(x^{\xi}_{j_{\lambda}} - x^{\xi}_{i}\right) \forall i \in \{1,...,n\}.$$

Similarly, $\mathbf{X}$ is said to be *min dominant* if and only if for every $\lambda \in \{1,...,k\}$ there exists and index $j_{\lambda} \in \{1,...,n\}$ such that

$$x^{\lambda}_{j_{\lambda}} - x^{\lambda}_{i} = \bigwedge_{\xi=1}^{k}\left(x^{\xi}_{j_{\lambda}} - x^{\xi}_{i}\right) \forall i \in \{1,...,n\}.$$

The expressions that compound this definition appeared in the early theorems about perfect recall of Morphological Associative Memories [106, 105]. Their value as an identifiable property of the data has been discovered in the context of the formalization of the relationship between strong lattice independence, defined below, and the affine independence in the classical linear analysis.

Definition 7. A set of lattice independent vectors $\mathbf{X} = \left\{\mathbf{x}^1,...,\mathbf{x}^k\right\} \subset \mathbb{R}^n$ is said to be Strongly Lattice Independent (SLI) iff $\mathbf{X}$ is *max dominant* or *min dominant* or both.

Appendix D

Extreme Learning Machines or no-prop Neural Networks

Standard Single Layer Feed-forward Neural Network (SLFNs) training is too slow because of: (1) Usual gradient-based learning algorithms are slow and (2) all the parameters of the networks are tuned iteratively by using such learning algorithms. An Extreme Learning Machine (ELM) is a learning method that aims to overcome these limitations by randomly choosing weights connecting input vectors to hidden nodes and threshold values of hidden nodes [65, 64]. This Appendix give a brief historic background and a formal description of ELMs.

D.1 Introduction

ELMs can be included in the broader group of multilayer Neural Network learning techniques that do not require a back-propagation (BP) algorithm. Many researchers explored the universal approximation capabilities of standard multilayer feedforward neural networks in the nineties. Hornik proved that SLFNs are universal approximators [61]. Moreover, if the activation function is continuous, bounded and nonconstant, then continuous mappings can be approximated in measure by neural networks over compact input sets [60]. Leshno improved these results and proved that feedforward networks with a with a locally bounded piecewise continuous non-polynomial activation function can approximate any continuous functions. In the early 2000s, Huang further advanced these researches and proposed the ELM as a SLFN learning tool that did not require backpropagation for network parameter tuning [64, 65]. Since then, ELMs have seen great growth, both in new learning algorithm development and applications. Later in 2013, Bernard Widrow (co-invertor of the least mean squares filter (LMS) adaptive algorithm that would later lead to the BP technique) et al. proposed the No-Prop multilayer neural network learning algorithm [139]. The difference between ELM and the No-Prop algorithm lies in the training method for the output layer. The No-Prop algorithm uses the LMS gradient algorithm to do this. The objective is to minimize Mean Square Error (MSE). The ELM algorithm does this by essentially inverting the covariance matrix of the neuron inputs and multiplying by the vector of crosscorrelations between the neuron's inputs and its desired response. This is a direct method for finding a solution that minimizes MSE. No-Prop uses a gradient method and ELM uses matrix inversion. The experiments conducted in this work use the ELM method,which is formally described in the next section.

D.2 Formal definition of ELM

Given N arbitrary distinct samples (x_i,t_i), where $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \mathbb{R}^n$ are the data vectors and $t_i = [t_{i1}, t_{i2}, ..., t_{im}]^T \in \mathbb{R}^m$ are the target classes, a standard SLFN can be mathematically modeled as:

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(w_i x_j + b_i) = t_j, \tag{D.1}$$

where $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ is the weight vector connecting the ith hidden node and the input nodes, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ is the weight vector connecting the ith hidden node and the output nodes , b_i is the threshold of the ith node and $\tilde{N}$ is the number of hidden nodes. In matrix form:

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(w_i x_j + b_i) = t_j \longrightarrow H\beta = T, \tag{D.2}$$

where these matrices are defined as

$$H = \begin{bmatrix} g(w_1 x_1 + b_1) & \dots & g(w_{\tilde{N}} x_j + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 x_N + b_i) & \dots & g(w_{\tilde{N}} x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}, \tag{D.3}$$

$$\beta = \begin{bmatrix} \beta_i^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \text{ and } T = \begin{bmatrix} t_i^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \tag{D.4}$$

Matrix H is called the hidden layer output matrix. It's ith column is the ith hidden node output. For any SLFN, H is invertible and $||H\beta - T = 0||$. There also exists an error $\varepsilon < \|H\beta - T\|$ for a given $\tilde{N} \leq N$ ([65]. The solution to the traditional SLFN would be: Find $\hat{\beta}$, $\hat{w}$ and $\hat{b}$ so that $\left\|\hat{H}\hat{\beta} - \hat{T}\right\| = \min_{w_i, b_i, \beta} \|H\beta - T\|$.

The ELM learning approach proposes the following: For fixed input weights w_i and the hidden layer biases b_i, to train a SLFN is equivalent to finding least-squares solution $\hat{\beta}$ of the linear system

$$H\beta = T. \tag{D.5}$$

The smallest norm least-squares solution of the above system is

$$\hat{\beta} = H^{\dagger}T, \tag{D.6}$$

where $H^{\dagger}$ is the Moore–Penrose generalized inverse of H. On a side note, $H^{\dagger}$ can be calculated using Singular Value Decomposition or doing $(H^T H)^{-1} H^T$.

Finally, an ELM algorithm can be summarized as follows: Given training set of N (x_i, t_i) samples, an activation function $g(x)$, and hidden node number $\tilde{N}$,

1. Randomly assign w_i and b_i.
2. Calculate H.
3. Calculate $\beta = H^{\dagger}T$.

The ELM described above is the basic ELM which was first proposed on ([64]). Many more have been developed, in ([63]) -

Random hidden layer feature mapping based ELM, The orthogonal projection method can be used to obtain $H^{\dagger}$: $H^{\dagger} = (H^T H)^{-1} H^T$. In that case, we can add a ridge parameter $1/\lambda$ to the diagonal of $(H^T H)$. This regularization approach, known as ridge regression, stabilizes the solution ([58]. Thus, the calculation of the output weights β is:

$$\beta = \left(\frac{I}{\lambda} + H^T H\right)^{-1} H^T T \tag{D.7}$$

where I is an identity matrix the same size as H. This variation of the basic ELM is called Random hidden layer feature mapping based ELM ([63]). We will call it ELM-FM for convenience.

In addition to those described above, many more ELMs have been developed:, ([63]): Kernel based ELM, sequential ELMs, incremental ELMs, etc.

Bibliography

[1] At&t database of faces. AT&T Laboratories Cambridge.

[2] Psychological image collection at stirling pics. pics.stir.ac.uk.

[3] N. Acito, M. Diani, and G. Corsini. Hyperspectral signal subspace identification in the presence of rare signal components. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(4):1940–1954, 2010.

[4] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural computation*, 9(7):1545–1588, 1997.

[5] C. M Bachmann, T. L Ainsworth, and R. A Fusina. Exploiting manifold geometry in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):441– 454, March 2005.

[6] C. M Bachmann, T. L Ainsworth, and R. A Fusina. Improved manifold coordinate representations of Large-Scale hyperspectral scenes. *IEEE Transactions on Geoscience and Remote Sensing*, 44(10):2786–2803, October 2006.

[7] P. Bajorski. On the reliability of PCA for complex hyperspectral data. In *Proceedings of WHISPERS*, pages 1–5, Grenoble, 2009.

[8] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[9] A. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[10] P. N. Bellhumer, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(7):711–720, 1997.

[11] T. Blumensath and M.E. Davies. Gradient pursuits. *Signal Processing, IEEE Transactions on*, 56(6):2370 –2382, june 2008.

[12] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[13] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[14] Leo Breiman and E. Schapire. Random forests. In *Machine Learning*, volume 45, pages 5–32, 2001.

[15] Deng Cai, Xiaofei He, and Jiawei Han. Semi-supervised discriminant analysis. In *in Proc. of the IEEE Int. Conf. on Comp. Vision (ICCV)*, Rio De Janeiro, 2007.

[16] S. Chakraborty, V. Balasubramanian, and S. Panchanathan. Adaptive batch mode active learning. *Neural Networks and Learning Systems, IEEE Transactions on*, PP(99):1–1, 2014.

[17] Tsung-Han Chan, A Ambikapathi, Wing-Kin Ma, and Chong-Yung Chi. Robust affine set fitting and fast simplex volume max-min for hyperspectral endmember extraction. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(7):3982–3997, July 2013.

[18] C.-I. Chang and Q. Du. Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 42(3):608–619, 2004.

[19] C.-I. Chang and A. Plaza. A fast iterative algorithm for implementation of pixel purity index. *Geoscience and Remote Sensing Letters, IEEE*, 3(1):63–67, 2006.

[20] C.-I. Chang and H. Safavi. Progressive dimensionality reduction by transform for hyperspectral imagery. *Pattern Recognition*, 44(10-11):2760–2773, 2011.

[21] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[22] Rama Chellappa, Pawan Sinha, and P. Jonathon Phillips. Face recognition by computers and humans. *IEEE Computer*, 43(2):46–55, 2010.

[23] Fen Chen and Yan Zhang. Sparse hyperspectral unmixing based on constrained L_p - L_2 optimization. *Geoscience and Remote Sensing Letters, IEEE*, 10(5):1142–1146, 2013.

[24] Jae Y. Choi, Yong M. Ro, and Konstantinos N. Plataniotis. A comparative study of preprocessing mismatch effects in color image based face recognition. *Pattern Recognition*, 44(2):412–430, 2011.

[25] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.

[26] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15:201–221, 1994. 10.1007/BF00993277.

[27] Corinna Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[28] Lehel Csato, Manfred Opper, and Ole Winther. Tap gibbs free energy, belief propagation and sparsity. In *Advances in Neural Information Processing Systems*. MIT Press, 2001.

[29] Danmarks-Tekniske-Universitet. Ica:dtu toolbox, 2002. http://cogsys.imm.dtu.dk/toolbox/.

[30] Bruce A. Draper, Kyungim Baek, Marian Stewart Bartlett, and J. Ross Beveridge. Recognizing faces with pca and ica. *Computer Vision and Image Understanding*, 91(1-2):115 – 137, 2003. Special Issue on Face Recognition.

[31] O. Duran and M. Petrou. Robust endmember extraction in the presence of anomalies. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6):1986–1996, June 2011.

[32] O. Eches, N. Dobigeon, C. Mailhes, and J.-Y. Tourneret. Bayesian estimation of linear mixtures using the normal compositional model. application to hyperspectral imagery. *Image Processing, IEEE Transactions on*, 19(6):1403–1413, 2010.

[33] Yu Fang, Hao Li, Yong Ma, Kun Liang, Yingjie Hu, Shaojie Zhang, and Hongyuan Wang. Dimensionality reduction of hyperspectral images based on robust spatial information using locally linear embedding. *Geoscience and Remote Sensing Letters, IEEE*, 11(10):1712–1716, 2014.

[34] Yifan Fu, Bin Li, Xingquan Zhu, and Chengqi Zhang. Active learning without knowing individual instance labels: A pairwise label homogeneity query approach. *Knowledge and Data Engineering, IEEE Transactions on*, 26(4):808–822, April 2014.

[35] Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and Information Systems*, 35(2):249–283, 2013.

[36] Keionosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[37] Xiurui Geng, Kang Sun, Luyan Ji, and Yongchao Zhao. A fast volume-gradient-based band selection method for hyperspectral image. *Geoscience and Remote Sensing, IEEE Transactions on*, 52(11):7111–7119, Nov 2014.

[38] M. Graña. A brief review of lattice computing. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pages 1777 –1781, 2008.

[39] M. Graña, D. Chyzhyk, M. García-Sebastián, and C. Hernández. Lattice independent component analysis for functional magnetic resonance imaging. *Information Sciences*, 181:1910–1928, 2010.

[40] M. Graña, A. Manhaes-Savio, M. García-Sebastián, and E. Fernandez. A lattice computing approach for on-line fMRI analysis. *Image and Vision Computing*, 28(7):1155–1161, 2010.

[41] M. Graña, A.M. Savio, M. Garcia-Sebastian, and E. Fernandez. A lattice computing approach for on-line fMRI analysis. *Image and Vision Computing*, 28(7):1155–1161, July 2010.

[42] M. Graña, I. Villaverde, J.O. Maldonado, and C. Hernandez. Two lattice computing approaches for the unsupervised segmentation of hyperspectral images. *Neurocomputing*, 72(10-12):2111–2120, 2009.

[43] M. Graña, I. Villaverde, J.O. Maldonado, and C. Hernandez. Two lattice computing approaches for the unsupervised segmentation of hyperspectral images. *Neurocomputing*, 72(10-12):2111–2120, 2009.

[44] Manuel Graña, Darya Chyzhyk, Maite García-Sebastián, and Carmen Hernández. Lattice independent component analysis for functional magnetic resonance imaging. *Information Sciences*, 181(10):1910 – 1928, 2011. Special Issue on Information Engineering Applications Based on Lattices.

[45] Manuel Graña and MiguelA Veganzones. Endmember induction by lattice associative memories and multi-objective genetic algorithms. *EURASIP Journal on Advances in Signal Processing*, 2012(1), 2012.

[46] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explorer Newsetter.*, 11(1):10–18, November 2009.

[47] L. K. Hansen, J. Larsen, and T. Kolenda. *On Independent Component Analysis for Multimedia Signals*. CRC Press, 2000.

[48] L.K. Hansen, J. Larsen, and T. Kolenda. Blind detection of independent dynamic components. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 5, pages 3197 –3200, 2001.

[49] J. Harsanyi, W. Farrand, and C.-I Chang. Determining the number and identity of spectral endmembers: An integrated approach using neyman-pearson eigenthresholding and iterative constrained rms error minimization. In *Proc. 9th Thematic Conf. Geologic Remote Sensing*, 1993.

[50] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Proceedings of the Conference on Advances in Nerual Information Processing Systems*, 2003.

[51] Xiaofei He, Shuicheng Yan, Yuxiao Hu, P. Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[52] D. Heinz, C-I Chang, and M.L.G. Althouse. Fully constrained least-squares based linear unmixing [hyperspectral image classification]. In *Geoscience and Remote Sensing Symposium, 1999. IGARSS '99 Proceedings. IEEE 1999 International*, volume 2, pages 1401–1403 vol.2, 1999.

[53] D.C. Heinz and Chein-I Chang. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 39(3):529–545, Mar 2001.

[54] Guillaume Heusch and Sebastien Marcel. A novel statistical generative model dedicated to face recognition. *Image and Vision Computing*, 28(1):101 – 110, 2010.

[55] R. Heylen, D. Burazerovic, and P. Scheunders. Non-linear spectral unmixing by geodesic simplex volume maximization. *Selected Topics in Signal Processing, IEEE Journal of*, 5(3):534–542, June 2011.

[56] R. Heylen and P. Scheunders. Non-linear fully-constrained spectral unmixing. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pages 1295–1298, July 2011.

[57] T.K. Ho. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(8):832–844, 1998.

[58] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, February 1970.

[59] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, April 1982.

[60] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251 – 257, 1991.

[61] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[62] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multiclass classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, PP(99):1 –17, 2011.

[63] Guang-Bin Huang, Dianhui Wang, and Yuan Lana. Extreme learning machines: A survey. *International Journal of Machine Leaning and Cybernetics*, in Press:107–122, April 2011.

[64] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985 – 990, july 2004.

[65] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70:489–501, 2006.

[66] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(10):1936–1949, Oct 2014.

[67] Jason Van Hulse and Taghi Khoshgoftaar. Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 68(12):1513 – 1542, 2009. Including Special Section: 21st IEEE International Symposium on Computer-Based Medical Systems (IEEE CBMS 2008) - Seven selected and extended papers on Biomedical Data Mining.

[68] Aapo Hyvarinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13:411–430, 2000.

[69] Pedro A.d.F.R. HÃžjen-SÃžrensen, Ole Winther, and Lars Kai Hansen. Mean-field approaches to independent component analysis. *Neural Computation*, 14(4):889–918, 2002.

[70] M.-D. Iordache, J.M. Bioucas-Dias, and A. Plaza. Sparse unmixing of hyperspectral data. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(6):2014–2039, 2011.

[71] R. Green J. Boardman, F. Kruse. Mapping target signatures via partial unmixing of aviris data. 1995.

[72] Wu Jun, Wang Shitong, and Fu-lai Chung. Positive and negative fuzzy rule system, extreme learning machine and image classification. *International Journal of Machine Learning and Cybernetics*, 2:261–271, 2011. 10.1007/s13042-011-0024-1.

[73] J.Yang, D.Zhang, A.F.Frangi, and J.Yang. Two-dimensional pca: a new approach to appearance based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):131–137, 2004.

[74] Sarvani Kare, Ashok Samal, and David Marx. Using bidimensional regression to assess face similarity. *Machine Vision and Applications*, 21(3):261–274, 2008.

[75] N. Keshava and J. F. Mustard. Spectral unmixing. *Signal Processing Magazine, IEEE*, 19(1):44–57, 2002.

[76] Jinkwon Kim, Hang Sik Shin, Kwangsoo Shin, and Myoungho Lee. Robust algorithm for arrhythmia classification in ecg using extreme learning machine. *Biomedical Enineering Online*, 8:article–number 31, 2009.

[77] C.L. Lawson and R.J. Hanson. *Solving Least-Squares Problems*, chapter 23, page 161. Prentice-Hall, 1974.

[78] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[79] DD Lee and HS Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[80] R Lienhart and J Maydt. An extended set of haar-like features for rapid object detection. In *2002 International Conference On Image Processing, Proceedings*, volume 1, pages 900–903. IEEE Signal Proc Soc, 2002.

[81] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, and S.Z. Li. Ensemble-based discriminant learning with boosting for face recognition. *IEEE Transactions on Neural Networks*, 17(1):166–178, January 2006.

[82] X. Lu, H. Wu, Y. Yuan, P. Yan, and X. Li. Manifold regularized sparse nmf for hyperspectral unmixing. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(5):2815–2826, 2013.

[83] Ion Marques and Manuel Graña. Experiments on lattice independent component analysis for face recognition. In *New Challenges on Bioinspired Applications*, volume 6687 of *Lecture Notes in Computer Science*, pages 286–294. Springer Berlin / Heidelberg, 2011.

[84] Ion Marques and Manuel Graña. Face recognition with lattice independent component analysis and extreme learning machines. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, in press:1–13, 2012.

[85] Ion Marques and Manuel Graña. Hybrid sparse linear and lattice method for hyperspectral image unmixing. In *Hybrid Artificial Intelligence Systems*, volume 8480 of *Lecture Notes in Computer Science*, pages 266–273. Springer International Publishing, 2014.

[86] S. Milborrow, J. Morkel, and F. Nicolls. The muct landmarked face database. *Pattern Recognition Association of South Africa*, 2010.

[87] Abdul A. Mohammed, Q. M. Jonathan Wu, and Maher A. Sid-Ahmed. Application of wave atoms decomposition and extreme learning machine for fingerprint classification. In *Image Analysis and Recognition, 2010, PT II, Proceedings*, volume 6112 of *Lecture Notes in Computer Science*, pages 246–255. Springer-Verlag, 2010.

[88] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72:3634–3637, 1994.

[89] Martin F. Moller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533, 1993.

[90] A.V. Nefian. Embedded bayesian networks for face recognition. In *Proc. of the IEEE International Conference on Multimedia and Expo*, volume 2, pages 133–136, Lusanne, Switzerland, August 2002.

[91] H.B. Nielsen. Ucminf - an algorithm for unconstrained, nonlinear optimization. Technical Report IMM-TEC-0019, IMM, Technical University of Denmark, 2001.

[92] Manfred Opper and Ole Winther. Adaptive and self-averaging thouless-anderson-palmer mean-field theory for probabilistic modeling. *Phys. Rev. E*, 64(5):056131, Oct 2001.

[93] Manfred Opper and Ole Winther. Tractable approximations for probabilistic models: The adaptive thouless-anderson-palmer mean field approach. *Phys. Rev. Lett.*, 86(17):3695–3699, Apr 2001.

[94] Yaozhang Pan, Shuzhi Sam Ge, Hongsheng He, and Lei Chen. Real-time face detection for human robot interaction. In *RO-MAN 2009: The 18th IEEE International Symposium On Robot And Human Interactive Communication*, volume 1 and 2, pages 15–20, 2009.

[95] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1090–1104, 2000.

[96] P.J. Phillips, H. Wechsler, and P. Rauss J. Huang. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[97] A. Plaza and C.-I. Chang. Impact of initialization on design of endmember extraction algorithms. *Geoscience and Remote Sensing, IEEE Transactions on*, 44:3397–3407, 2006.

[98] Mattia C. F. Prosperi, Andre Altmann, Michal Rosen-Zvi, Ehud Aharoni, Gabor Borgulya, Fulop Bazso, Anders Sonnerborg, Eugen Schuelter, Daniel Struck, Giovanni Ulivi, Anne-Mieke Vandamme, Jurgen Vercauteren, Maurizio Zazzi, and EuResist Virolab Study Grp. Investigation of expert rule bases, logistic regression, and non-linear machine learning techniques for predicting response to antiretroviral treatment. *Antiviral Therapy*, 14(14):433–442, 2009.

[99] H. Pu, Z. Chen, B. Wang, and W. Xia. Constrained least squares algorithms for nonlinear unmixing of hyperspectral imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(3):1287–1303, March 2015.

[100] Hanye Pu, Zhao Chen, Bin Wang, and Geng-Ming Jiang. A novel Spatial-Spectral similarity measure for dimensionality reduction and classification of hyperspectral imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 52(11):7008–7022, 2014.

[101] Lishan Qiao, Songcan Chen, and Xiaoyang Tan. Sparsity preserving discriminant analysis for single training image face recognition. *Pattern Recognition Letters*, 31(5):422 – 429, 2010.

[102] B. Raducanu, M. Gra na, and F. X. Albizuri. Morphological scale spaces and associative morphological memories: Results on robustness and practical applications. *Journal of Mathematical Imaging and Vision*, 19(2):113–131, 2003.

[103] Chuan-Xian Ren and Dao-Qing Dai. Incremental learning of bidirectional principal components for face recognition. *Pattern Recognition*, 43(1):318 – 330, 2010.

[104] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *IEEE International Conference on Neural Networks*, volume 1, pages 586–591, 1993.

[105] G. X. Ritter, J. L. Diaz-de-Leon, and P. Sussner. Morphological bidirectional associative memories. *Neural Networks*, 12(6):851–867, July 1999.

[106] G. X. Ritter, P. Sussner, and J. L. Diaz-de-Leon. Morphological associative memories. *Neural Networks, IEEE Transactions on*, 9(2):281–293, 1998.

[107] G. X. Ritter, G. Urcid, and L. Iancu. Reconstruction of patterns from noisy inputs using morphological associative memories. *Journal of Mathematical Imaging and Vision*, 19(2):95–111, 2003.

[108] Gerhard X. Ritter and Gonzalo Urcid. A lattice matrix method for hyperspectral image unmixing. *Information Sciences*, 181(10):1787–1803, May 2010.

[109] Gerhard X. Ritter and Gonzalo Urcid. A lattice matrix method for hyperspectral image unmixing. *Information Sciences*, 181(10):1787–1803, 2011.

[110] Gerhard X. Ritter, Gonzalo Urcid, and Mark S. Schmalz. Autonomous single-pass endmember approximation using lattice auto-associative memories. *Neurocomput.*, 72(10-12):2101–2110, 2009.

[111] G.X. Ritter and P. Gader. Fixed points of lattice transforms and lattice associative memories. In Peter Hawkes, editor, *Advances in Imaging and Electron Physics,*, volume 144, pages 165–242. Academic Press, 2006.

[112] G.X. Ritter, P. Sussner, and J.L. Diaz de Leon. Morphological associative memories. *Neural Networks, IEEE Transactions on*, 9(2):281 –293, 1998.

[113] G.X. Ritter and G. Urcid. Lattice algebra approach to endmember determination in hyperspectral imagery. In Peter W. Hawkes, editor, *Advances in Imaging and Electron Physics,*, volume 160, pages 113–169. Academic Press, Burlington, 2010.

[114] G.X. Ritter and G. Urcid. A lattice matrix method for hyperspectral image unmixing. *Information Sciences*, 181(10):1787–1803, May 2011.

[115] G.X. Ritter, G Urcid, and Schmalz M.S. Autonomous single-pass endmember approximation using lattice auto-associative memories. *Neurocomputing*, 72(10-12):2101–2110, 2009.

[116] S. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[117] Nicholas Roy and Andrew Mccallum. Toward optimal active learning through sampling estimation of error reduction. In *In Proc. 18th International Conf. on Machine Learning*, pages 441–448. Morgan Kaufmann, 2001.

[118] B. Scholkopf, A. Smola, and KR Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[119] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13:1443–1471, July 2001.

[120] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

[121] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[122] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, MAY 4 2009.

[123] Chen Shi and Le Wang. Incorporating spatial information in spectral unmixing: A review. *Remote Sensing of Environment*, 149(0):70 – 87, 2014.

[124] Kang Sun, Xiurui Geng, and Luyan Ji. A new sparsity-based band selection method for target detection of hyperspectral image. *Geoscience and Remote Sensing Letters, IEEE*, 12(2):329–333, Feb 2015.

[125] Kang Sun, Xiurui Geng, Luyan Ji, and Yun Lu. A new band selection method for hyperspectral image based on data quality. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 7(6):2697–2703, June 2014.

[126] Zhan-Li Sun, Tsan-Ming Choi, Kin-Fan Au, and Yong Yu. Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, 46(1):411–419, December 2008.

[127] P. Sussner. Fixed points of autoassociative morphological memories. In *IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000*, volume 5, pages 611–616 vol.5. IEEE, 2000.

[128] Peter Sussner. Generalizing operations of binary autoassociative morphological memories using fuzzy set theory. *Journal of Mathematical Imaging and Vision*, 19(2):81–93, September 2003. ACM ID: 859104.

[129] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319 – 2323, December 2000.

[130] K.E. Themelis, A.A. Rontogiannis, and K.D. Koutroumbas. A novel hierarchical bayesian approach for sparse semisupervised hyperspectral unmixing. *Signal Processing, IEEE Transactions on*, 60(2):585–599, 2012.

[131] Hui-Xin Tian and Zhi-Zhong Mao. An ensemble elm based on modified adaboost.rt algorithm for predicting the temperature of molten steel in ladle furnace. *IEEE Transactions On Automation Science And Engineering*, 7(1):73–80, January 2010.

[132] D. Tuia, E. Pasolli, and W.J. Emery. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 2011.

[133] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neurosicence*, 3(1):71–86, 1991.

[134] MA Veganzones and M Graña. Endmember extraction methods: A short review. In *Knowledge-Based Intelligent Information and Engineering Systems, pt 3*, volume 5179 of *Lecture Notes In Computer Science*, 2008.

[135] Ivan Villaverde, Borja Fernandez-Gauna, and Ekaitz Zulueta. Lattice independent component analysis for mobile robot localization. In E Corchado, MG Romay, and AM Savio, editors, *Hybrid Artificial Intelligence Systems, pt 2*, volume 6077 of *Lecture Notes in Artificial Intelligence*, pages 335–342. Springer-Verlag, 2010.

[136] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 511–518, 2001.

[137] J. Wang and C.-I. Chang. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *Geoscience and Remote Sensing, IEEE Transactions on*, 44(6):1586–1600, 2006.

[138] Liguo Wang, Fangjie Wei, Danfeng Liu, and Qunming Wang. Fast implementation of maximum simplex volume-based endmember extraction in original hyperspectral data space. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 6(2):516–521, April 2013.

[139] Bernard Widrow, Aaron Greenblatt, Youngsik Kim, and Dookun Park. The no-prop algorithm: A new learning algorithm for multilayer neural networks. *Neural Networks*, 37(0):182 – 188, 2013. Twenty-fifth Anniversay Commemorative Issue.

[140] M.E. Winter. N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspectral data. In *Imaging Spectrometry V*, volume 3753 of *SPIE Proceedings*, pages 266–275. SPIE, 1999.

[141] W.S. Yambor. *Analysis of PCA-based and Fisher Discriminant-Based Image Recognition Algorithms*. Technical report cs-00-103, Computer Science Department, Colorado State University, July 2000.

[142] Shuyuan Yang, PengLei Jin, Bin Li, Lixia Yang, Wenhui Xu, and Licheng Jiao. Semisupervised dual-geometric subspace projection for dimensionality

reduction of hyperspectral image data. *Geoscience and Remote Sensing, IEEE Transactions on*, 52(6):3587–3593, June 2014.

[143] Yuan Yuan, Guokang Zhu, and Qi Wang. Hyperspectral band selection by multitask sparsity pursuit. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(2):631–644, Feb 2015.

[144] A Zare, P. Gader, and G. Casella. Sampling piecewise convex unmixing and endmember extraction. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(3):1655–1665, March 2013.

[145] B. Zhang, X. Sun, L. Gao, and L. Yang. Endmember extraction of hyperspectral remote sensing images based on the discrete particle swarm optimization algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4173–4176, November 2011.

[146] Bing Zhang, Jianwei Gao, Lianru Gao, and Xu Sun. Improvements in the ant colony optimization algorithm for endmember extraction from hyperspectral images. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 6(2):522–530, April 2013.

[147] Daoqiang Zhang and Zhi-Hua Zhou. (2D)2PCA: two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing*, 69:224–231, 2005.

[148] Xiangrong Zhang, Yudi He, Nan Zhou, and Yaoguo Zheng. Semisupervised dimensionality reduction of hyperspectral images via local scaling cut criterion. *Geoscience and Remote Sensing Letters, IEEE*, 10(6):1547–1551, Nov 2013.

[149] Yanfei Zhong, Lin Zhao, and Liangpei Zhang. An adaptive differential evolution endmember extraction algorithm for hyperspectral remote sensing imagery. *Geoscience and Remote Sensing Letters, IEEE*, 11(6):1061–1065, June 2014.

[150] Dake Zhou and Zhenmin Tang. Kernel-based improved discriminant analysis and its application to face recognition. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 14(2):103–111, 2009.

[151] S. Zhou and R. Chellappa. Multiple-exemplar discriminant analysis for face recognition. In *Proc. of the 17th International Conference on Pattern Recognition, ICPR'04*, pages 191–194, Cambridge, UK, 2004.