



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecommunicació de Barcelona



Machine Learning for analysis of hyperspectral images

Degree Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

David Cubero Valentin

In partial fulfillment
of the requirements for the degree in
Telecommunications Technologies and Services Engineering

Advisors:
Jörgen Ahlberg (LiU)
Philippe Salembier (UPC ETSETB)
Linköping, January 2022



Contents

List of Figures	3
List of Tables	4
1 Introduction	10
1.1 Initial work plan	10
1.2 Deviations from the initial work plan and incidents	12
1.3 Final Gantt Diagram	15
2 State of the art	16
2.1 Hyperspectral Images	16
2.2 Dimensionality Reduction: Principal Component Analysis	17
2.3 Detection	18
2.3.1 AMF Detector	18
2.3.2 SAM Detector	18
2.3.3 OSP Detector	19
2.3.4 Preprocessing	19
2.4 Clustering Technique: K-Means	20
2.5 Evaluation	22
3 Methodology and Results	24
3.1 Hardware: Hyperspectral Camera	24
3.2 Data collection	25
3.2.1 Blood Data	25
3.2.2 Spruces Data	27
3.3 Evaluation	28
3.4 Software Development and Results	29
3.4.1 AMF & SAM Detectors	29
3.4.2 OSP Detector	34
3.4.3 Detection: Analysis & Discussion	35
3.4.4 K-Means	35
3.4.5 K-Means + AMF & SAM Detectors	39
3.4.6 K-Means + OSP Detector	40
3.4.7 Clustering + Detection: Analysis & Discussion	43
3.4.8 Testing on new data	43
4 Budget	46
5 Conclusions	47
6 Future Work	48
References	49

List of Figures

1	Initial Work Breakdown Structure.	11
2	Final Work Breakdown Structure.	13
3	Gantt diagram of the project.	15
4	Difference between RGB images and hyperspectral images, from [1].	16
5	K-Means algorithm, from [2].	22
6	Confusion matrix, from [3].	22
7	How performance is determined by ROC curves, from [4].	23
8	Glana L4 and its specifications, from [5].	24
9	An CVF mounted on an FPA for hyperspectral imaging, from [6].	25
10	Spatio-spectral scanning, from [6].	25
11	Scenario with pig blood traces.	25
12	Little mushroom as a distractor.	26
13	Calibrating the hyperspectral camera.	26
14	Blood Baseline Image.	26
15	The drone and the drone flightmap.	27
16	Plantation field.	27
17	Spruces Baseline Image.	28
18	Ideal mask of the baseline image.	28
19	19a, 19b, 19c shows evaluation metrics; 19d shows resulting image after detection.	30
20	Comparing data structure before and after whitening with PCA in AMF.	31
21	21a, 21b, 21c shows evaluation metrics; 21d shows resulting image after detection.	31
22	Hyper-cube eigenvalues.	32
23	Results for AMF detector. 23a, 23b, 23c shows evaluation metrics; 23d shows resulting image after detection for the best result (5 eigenvalues).	33
24	Results for SAM detector. 24a, 24b, 24c shows evaluation metrics; 24d shows resulting image after detection for the best result (5 eigenvalues).	33
25	Results for OSP detector. 25a, 25b, 25c shows evaluation metrics; 25d shows resulting image after detection for the best result (2 eigenvalues).	35
26	Scatterplot from the first 2 dimensions of the raw cube (without zoom and zooming).	36
27	Baseline image with color labels for the 8 clusters obtained from K-Means.	36
28	Scatterplot from the first 2 dimensions of the low-dimensional cube (without zoom and zooming).	37
29	Baseline image with color labels for the 8 clusters obtained from K-Means.	37
30	Scatterplot from the first 2 dimensions of the low-dimensional cube (without zoom and zooming).	38
31	Baseline image with color labels for 9 clusters obtained from K-Means.	38
32	Resulting mask from taking only blood clusters.	39
33	Results for AMF detector. 33a, 33b, 33c shows evaluation metrics; 33d shows resulting image after detection for the best result (2 eigenvalues).	39
34	Results for SAM detector. 34a, 34b, 34c shows evaluation metrics; 34d shows resulting image after detection for the best result (2 eigenvalues).	40

35	Results for OSP detector. 35a, 35b, 35c shows evaluation metrics; 35d shows resulting image after detection for the best result (6 dimensions of background subspace).	41
36	Ideal mushroom mask.	42
37	Results for OSP detector. 37a, 37b, 37c shows evaluation metrics; 37d shows resulting image after detection for the best result (5 dimensions of background subspace).	42
38	Qualitative results. 38a shows K-Means results, 38b shows K-Means + AM-F/SAM results, 38c shows K-Means + OSP results.	44

Listings

List of Tables

1	Short description and internal tasks of initial WP1.	11
2	Short description and internal tasks of initial WP2.	11
3	Short description and internal tasks of initial WP3.	11
4	Short description and internal tasks of initial WP4.	12
5	Short description and internal tasks of initial WP5.	12
6	Short description and internal tasks of initial WP6.	12
7	Short description and internal tasks of final WP1.	13
8	Short description and internal tasks of final WP2.	13
9	Short description and internal tasks of final WP3.	13
10	Short description and internal tasks of final WP4.	14
11	Short description and internal tasks of final WP5.	14
12	Short description and internal tasks of final WP6.	14
13	Short description and internal tasks of final WP7.	14
14	Short description and internal tasks of final WP8.	14
15	Results from detection.	35
16	Results from clustering + detection.	43
17	Wage and material costs.	46

Abstract

With the advent of less expensive hyperspectral cameras, now hyperspectral images are increasingly used in many fields and thus, first, traditional machine learning tools should be exploited for its analysis. The project is focused on target detection to find similarities within a hyperspectral image, where we first implement the AMF, SAM and OSP detectors, where the OSP is shown to be superior. With this results, there are some distractors and noise items that the detectors are not capable of disregard. This is why we use machine learning, specifically K-Means as clustering technique, in order to remove useless information and possible noise, and then to apply the detectors only on the target cluster. Using K-Means + AMF/SAM Detector improves the results. On the contrary, with K-Means + OSP Detector the results get worse since K-Means provides a good clustering and thus, the target pixels are considered as background subspace instead of anomalies. Using new data, where K-Means does not provide good clustering, K-Means + OSP Detector provides better results than before although K-Means + AMF/SAM Detector stills being a bit better.

Resum

Amb l'arribada de càmeres hiperespectrals menys cares, ara les imatges hiperespectrals s'utilitzen cada cop més en moltes àrees i, per tant, en primer lloc, eines tradicionals de l'aprenentatge automàtic haurien de ser explotades per al seu anàlisis. El projecte es centra en la detecció d'objectius per trobar similituds en una imatge hiperespectral, on primer implementem els detectors AMF, SAM i OSP, sent l'OSP el que funciona millor. Amb aquests resultats, hi ha alguns distractors i elements de soroll que els detectors no són capaços d'ignorar. És per això que utilitzem l'aprenentatge automàtic, especialment K-Means com a tècnica d'agrupació, per tal d'eliminar informació inútil i possible soroll, i llavors aplicar els detectors només al grup on està l'objectiu. Utilitzant K-Means + Detector AMF/SAM els resultats milloren. En canvi, amb K-Means + Detector OSP els resultats empitjoren ja que K-Means fa un agrupament bo i, per tant, els pixels objectius són considerats com a subespai de fons en comptes d'anomalies. Utilitzant dades noves on K-Means no proporciona bons resultats, K-Means + Detector OSP ofereix millors resultats encara que K-Means + Detector AMF/SAM segueix sent una mica millor.

Resumen

Con la llegada de cámaras hiperespectrales menos caras, ahora las imágenes hiperespectrales se utilizan cada vez más en muchos campos y, por lo tanto, en primer lugar, herramientas tradicionales del aprendizaje automático deberían ser explotadas para su análisis. El proyecto está enfocado en la detección de objetivos para encontrar similitudes en una imagen hiperespectral, donde primero implementamos los detectores AMF, SAM y OSP siendo el OSP el que mejor funciona. Con estos resultados, hay algunos distractores y elementos de ruido que los detectores no son capaces de ignorar. Es por eso que usamos el aprendizaje automático, especialmente K-Means como técnica de agrupación, para eliminar información inútil y posible ruido, y entonces aplicar los detectores solo en el grupo donde esta el objetivo. Utilizando K-Means + Detector AMF/SAM los resultados mejoran. En cambio, con K-Means + Detector OSP los resultados empeoran ya que K-Means hace un buena agrupación, y por lo tanto, los píxeles objetivo son considerados como subespacio de fondo en vez de anomalías. Usando nuevos datos donde K-Means no proporciona buena detección, K-Means + Detector OSP proporciona mejores resultados aunque K-Means + Detector AMF/SAM sigue siendo un poco mejor.

Acknowledgments

I would like to express my gratitude to my project supervisor Jörgen Ahlberg for offering me this project, guide me through it and for the great discussions we have had. Also, I would like to thank Glana Sensors AB members Mathias Kindstedt and Mikael Adlers for spending their time sending me some software and all data collection of the project. Furthermore, thank you to Philippe Salembier for your attention from distance.

In addition, I would like to mention Linköping University for its great reception and to the Erasmus community for making possible this unbelievable experience.

For last, say thank you to my parents for helping me when I needed, to my sisters which are telecommunications engineers too for giving me advises, to my girlfriend for sharing together this stage of our lives and support each other, to my friends for spending really good moments that gave me disconnection in tough moments and for last, to my grandfather who has given me strength from above.

Revision history and approval record

Revision	Date	Purpose
0	16/10/2021	Document creation
1	27/12/2021	Document revision
2	12/01/2022	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
David Cubero Valentin	david.cubero@estudiantat.upc.edu
Jörgen Ahlberg	jorgen.ahlberg@liu.se
Philippe Salembier	philippe.salembier@upc.edu

Written by:		Reviewed and approved by:	
Date	09/12/2021	Date	12/01/2022
Name	David Cubero Valentin	Name	Jörgen Ahlberg
Position	Project Author	Position	Project Supervisor

1 Introduction

The aim of the project is to exploit traditional tools and recent advances in machine learning for hyperspectral imaging and implement a small tool set that allows users to analyse hyperspectral images interactively. One final application example could be to analyse images from a drone flying over a forest, automatically segment the images (for example, different kinds of trees) identifying the segments with known classes (spruce, birch, ...), then let the user select a certain anomaly within a class (for example a spruce damaged by bark beetles), and automatically find similar anomalies to the selected one. That being said, the project main goals are:

1. Perform a literature study of methods for supervised, semi-supervised and unsupervised learning of hyperspectral images for detection, clustering, segmentation and anomaly detection.
2. Implement selected methods for semi-supervised analysis of hyperspectral images. These methods are anticipated to form a chain of tools, starting with finding interesting areas/objects, refining them by user input, and possibly identifying the areas/objects as belonging to specific classes from a library of signatures. Due to the limited time, the number of methods to implement will be kept small.
3. Test and demonstrate the methods on a selected data set from forestry or agriculture.

The project requirements are basically starting from a target, find similarities of this within the hyperspectral image, as well as to evaluate, implement and demonstrate several methods for finding those similarities.

As project specifications, the project is not required to reach any specific score, since the purpose is only to study the different methods and compare them.

The project is carried out at Linköping University (LiU) inside the Department of Electrical Engineering (ISY) in the Computer Vision Laboratory. The project also has collaboration from Glana Sensors AB company which provides software and data. The project itself starts from scratch but it is performed under the framework of the company research (Glana Sensors AB) which has carried out some others hyperspectral images projects before. The main project initial ideas are provided by the project supervisor.

1.1 Initial work plan

The initial work plan follows the Work Breakdown Structure seen in Fig.1

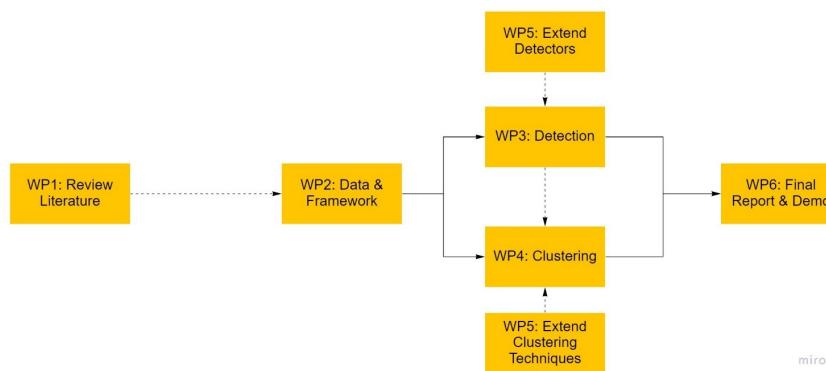


Figure 1: Initial Work Breakdown Structure.

The tasks related to the Work Packages (WP) seen are the followings:

WP1: Review Literature	
Short Description	Read papers about methods used in hyperspectral images and anomaly detection.
Internal Tasks	-Write down the most important things of each paper.

Table 1: Short description and internal tasks of initial WP1.

WP2: Data & Framework	
Short Description	Set up all the framework, acquire hyperspectral image data and learn how is structured.
Internal Tasks	-Install Pycharm in order to set up the environment of the project. -Get the data and learn how to deal with hyperspectral image data.

Table 2: Short description and internal tasks of initial WP2.

WP3: Detection	
Short Description	Implement different detectors to find similarities in the hyperspectral images, selecting the target/reference.
Internal Tasks	-Implement SAM detector. -Implement AMF detector. -Implement OSP detector.

Table 3: Short description and internal tasks of initial WP3.

WP4: Clustering	
Short Description	Implement different clustering techniques from the data and find similarities within given cluster.
Internal Tasks	-Implement K-Means. -Implement GMM. -Implement a modified GAN [7].

Table 4: Short description and internal tasks of initial WP4.

WP5: Extend Detectors/Clustering Techniques	
Short Description	Extend similarities/detectors or clustering if possible.
Internal Tasks	-Search for other options of detectors and clustering. -Implement them and see if it improves the previous ones.

Table 5: Short description and internal tasks of initial WP5.

WP6: Final Report & Demo	
Short Description	Prepare the final report and the final demonstration.
Internal Tasks	-Develop and finish the final report. -Develop and finish the final demonstration/presentation.

Table 6: Short description and internal tasks of initial WP6.

1.2 Deviations from the initial work plan and incidents

While doing the project, we started to notice that while implementing the detectors, a lot of experiments could be done by adding some modifications on detectors in order to improve the results and consequently, having to track new parameters. Then on clustering + detection (with K-Means), as consequence of the results, also some modifications and new experiments had to be done (having to track new parameters too). As a result of all this, and given the amount of time spent on this, we have had to focus only in one clustering method (K-Means) having to put aside recent advances in machine learning such as the modified GAN [7].

On the other hand, as being very focused on K-Means, we wanted to obtain quantitative results, so we decided to create a ground-truth to compute performance measures as evaluation method.

Furthermore, we have decided in the end to evaluate the complete processing chain by applying the studied method (with the best possible parameters) with new data, to be able to see qualitative results of the method implemented in that new data.

As incidents, the blood data collection was supposed to be taken by a drone, but while flying the drone crashed and we had to acquire data from the ground, so in the end, the data used in this project is not the data that was expected at first.

The final WBS and the final tasks related to the WP are the followings:

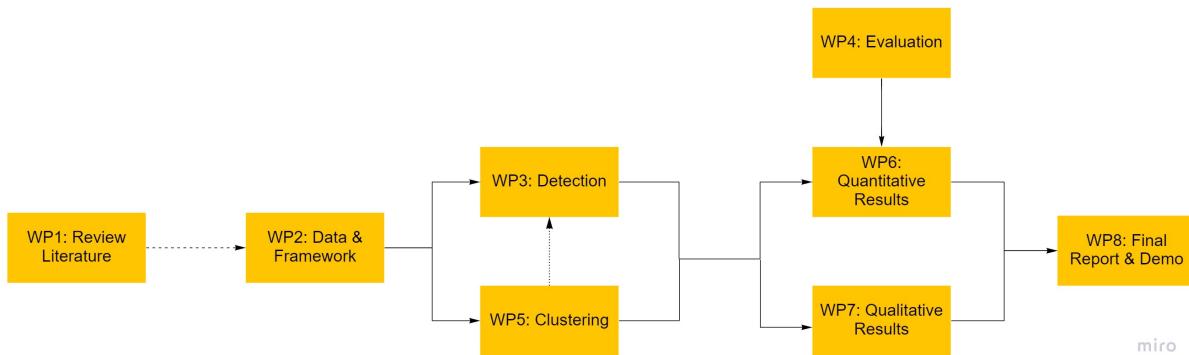


Figure 2: Final Work Breakdown Structure.

WP1: Review Literature	
Short Description	Read papers about methods used in hyperspectral images and anomaly detection.
Internal Tasks	-Write down the most important things of each paper.

Table 7: Short description and internal tasks of final WP1.

WP2: Data & Framework	
Short Description	Set up all the framework, acquire hyperspectral image data and learn how is structured.
Internal Tasks	-Install Pycharm in order to set up the environment of the project. -Get the data and learn how to deal with hyperspectral image data.

Table 8: Short description and internal tasks of final WP2.

WP3: Detection	
Short Description	Implement different detectors to find similarities in the hyperspectral images, selecting the target/reference.
Internal Tasks	-Implement detector SAM. -Implement detector AMF. -Implement detector OSP.

Table 9: Short description and internal tasks of final WP3.

WP4: Evaluation	
Short Description	Implement evaluation method in order to obtain quantitative results.
Internal Tasks	<ul style="list-style-type: none"> -Create an ideal mask of the hyperspectral image. -Write some code to obtain the following metrics: PR-curves, ROC-curves and F1-Score along the discriminant/threshold.

Table 10: Short description and internal tasks of final WP4.

WP5: Clustering	
Short Description	Implement K-Means as a clustering technique before detection.
Internal Tasks	<ul style="list-style-type: none"> -Implement K-Means. -Place the algorithm in the main code before detection.

Table 11: Short description and internal tasks of final WP5.

WP6: Quantitative Results	
Short Description	Obtain all the quantitative results.
Internal Tasks	<ul style="list-style-type: none"> -Obtain quantitative results for only detection. -Obtain quantitative results for only clustering. -Obtain quantitative results for clustering + detection.

Table 12: Short description and internal tasks of final WP6.

WP7: Qualitative Results	
Short Description	Use the best quantitative results obtained for analyzing qualitatively results in a new data.
Internal Tasks	<ul style="list-style-type: none"> -Adapt the code for the new data and solve possible issues that could appear. -Analyze qualitatively the results on that new data.

Table 13: Short description and internal tasks of final WP7.

WP8: Final Report & Demo	
Short Description	Prepare the final report and the final demonstration.
Internal Tasks	<ul style="list-style-type: none"> -Develop and finish the final report. -Develop and finish the final demonstration/presentation.

Table 14: Short description and internal tasks of final WP8.

1.3 Final Gantt Diagram

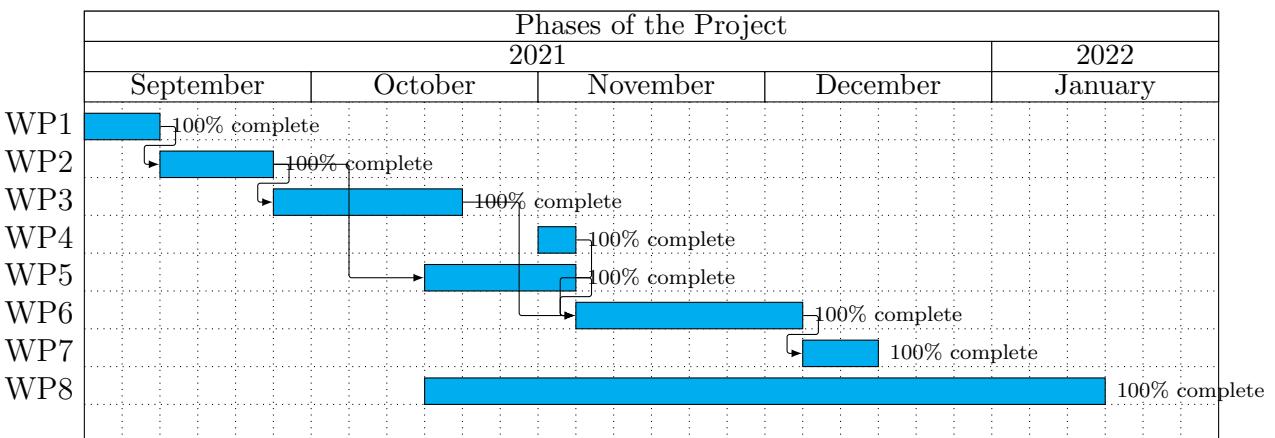


Figure 3: Gantt diagram of the project.

2 State of the art

In this section, we are going to explain all the theory fields related with the project in order to put some context on it as well as to be able to understand all the necessary background involved in order to develop the project correctly.

2.1 Hyperspectral Images

So, first, what is the meaning of hyperspectral cameras? A hyperspectral camera is a camera that works with many different wavelength bands. In a hyperspectral image, each pixel forms a (spectral) vector of measurements in the different bands. This vector, the observed spectral signature, contains information on the material(s) present in the scene, and can be exploited for detection, classification, and recognition. This is due to the fact that materials reflect and emit different amounts of light at different wavelengths – which is also the reasons we can see colors. Essentially, the spectral signature is a high-dimensional generalization of the concept of color [6]. This means basically that a hyperpsectral image not only can deal with the visible spectrum (e.g RGB images) but also with infrared spectrum. Nevertheless, on the other hand, it should be mentioned that there are some hyperspectral cameras that can be, for example, longwave infrared only.

Fig. 4 shows an illustration of how hyperspectral images can differ from RGB images.

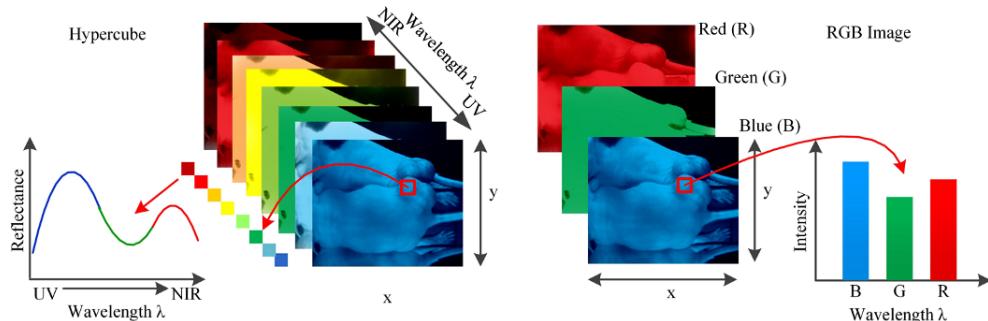


Figure 4: Difference between RGB images and hyperspectral images, from [1].

As seen in Fig. 4, hyperspectral images are often stored as a hyperspectral data-cube, or hyper-cube. A hypercube is a three dimensional matrix with the spatial information in two of the dimensions and the spectral information in the third dimension [8]. The size of the spectral dimension depends on the number of spectral bands covering the range of wavelengths [9].

Hyperspectral images have been used for a long time for remote sensing for military purposes and for applications like land-use classification. With the advent of less expensive hyperspectral cameras (and drones), hyperspectral imaging is anticipated to be increasingly used for applications in precision agriculture, forestry, and industrial inspection.

Some specific examples would be finding camouflaged targets since the spectral signature of a target can differ significantly from the background even if it is camouflaged. In

agriculture, for instance, the spectral signature can reveal plant diseases, stress, water and nitrogen content. Or another case for example, in food inspection, the spectral signature can be crucial for quality assessment of fruits and vegetables [6].

2.2 Dimensionality Reduction: Principal Component Analysis

Given a set of N -dimensional data samples $\{\mathbf{x}_k\}_{k=1}^K$ to analyze, a problem is often the massive amount of data, especially if the samples \mathbf{x}_k are high-dimensional. One solution is to reduce the dimensionality of the data, provided that this can be done without losing important information. A simple and popular way to reduce dimensionality and also to extract interesting features is *principal component analysis* (PCA). The procedure is as follows [10].

- Compute the mean and covariance of the training data:

$$\boldsymbol{\mu} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k \quad \boldsymbol{\Gamma} = \frac{1}{K-1} \sum_{k=1}^K (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^T$$

- Perform a *singular value decomposition* (SVD) or an *eigenvalue decomposition* of the covariance matrix, i.e., find the matrices $\boldsymbol{\Gamma} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$, where the columns of \mathbf{U} contain the subspace basis. $\boldsymbol{\Sigma}$ is a diagonal matrix where the elements σ_i^2 of the diagonal indicate the energy distribution of the training samples along the directions in the corresponding columns of \mathbf{U} . We assume in the following that the columns of \mathbf{U} (and $\boldsymbol{\Sigma}$) are ordered so that $\sigma_1^2 > \sigma_2^2 > \dots$
- The M -dimensional subspace basis is spanned by $\boldsymbol{\Phi} = [\mathbf{u}_1, \dots, \mathbf{u}_M]$, where the vectors \mathbf{u}_i are called the *principal components*. Typically, M is chosen so that a certain amount q , say 99 percent, of the signal energy is preserved, i.e.,

$$\frac{\sum_{i=1}^M \sigma_i^2}{\sum_{i=1}^N \sigma_i^2} \geq q.$$

- To project a sample \mathbf{x} on the subspace, i.e., to reduce the dimensionality, compute

$$\mathbf{y} = \boldsymbol{\Phi}^T(\mathbf{x} - \boldsymbol{\mu})$$

- To *whiten data*, i.e., to transform it so that its components are uncorrelated and have equal variance (see specific steps in Sec. 2.3.4), compute

$$\tilde{\mathbf{x}} = \boldsymbol{\Gamma}^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu})$$

In the case of $\boldsymbol{\Gamma}$ not having full rank, which means that the training data is fully contained in a space of lower dimensionality, or if a dimensionality reduction should be performed for other reasons, use

$$\tilde{\mathbf{y}} = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\Phi}^T(\mathbf{x} - \boldsymbol{\mu})$$

2.3 Detection

In this subsection, we are going to talk about detectors which are basically mathematical tools set extracted from hypothesis that helps to find similarities between data. We are going to define three detectors that are useful while knowing the target, and discuss which features contains each one as well as which preprocessing steps can be applied on each.

2.3.1 AMF Detector

The situation where we try to detect a known signal disturbed by additive Gaussian noise is common in communication systems, for example, a specific radio wave might be transmitted and the receiver receives the signal plus background noise. In remote sensing, the signal would be the spectral signature for a certain material, and the noise would be the background clutter, atmospheric effects, and sensor noise [6] [10].

The hypotheses are

$$\begin{aligned} H_0 : \mathbf{x} &= \mathbf{b}, & \mathbf{b} &\sim N(\boldsymbol{\mu}, \boldsymbol{\Gamma}) \\ H_1 : \mathbf{x} &= \mathbf{t} + \mathbf{b} \end{aligned}$$

This signal can then be correlated with the *matched filter* \mathbf{t} , giving us a scalar output $y = \mathbf{t}^T \mathbf{x}$ and the one-dimensional problem:

$$\begin{aligned} H_0 : x &= b, & b &\sim N(\mu, \Gamma) \\ H_1 : x &= \|\mathbf{t}\|^2 + b \end{aligned}$$

Creating $y' = \frac{y}{\|\mathbf{t}\|^2}$ to make the mean (under H_1) equal one, the resulting detector is

$$D_{AMF}(\mathbf{x} | \mathbf{B}, \mathbf{T}) = \left[\frac{\mathbf{t}^T \mathbf{x}}{\|\mathbf{t}\|^2} > t \right]$$

This is called the *adaptive matched filter* (AMF) detector. This detector is optimal when the target pixels(s) does not contain any background, that is, for resolved targets [6].

2.3.2 SAM Detector

If we know the target signature except for the norm, for example in the subpixel case or due to transmission effects, we instead have the following hypotheses [6] [10]:

$$\begin{aligned} H_0 : \mathbf{x} &= \mathbf{b}, & \mathbf{b} &\sim N(\boldsymbol{\mu}, \boldsymbol{\Gamma}) \\ H_1 : \mathbf{x} &= k\mathbf{t} + \mathbf{b} \end{aligned}$$

where k is an unknown parameter.

An alternative formulation is that we know the *structure* but not the *level* of the noise.

The angle α between \mathbf{x} and \mathbf{t} is uniformly distributed in the interval $[-\pi, \pi]$ under H_0 and centered around zero under H_1 . The resulting detector is called the *spectral angle mapper*

(SAM). Using the cosine instead of the angle, we get

$$\begin{aligned} D_{SAM}(\mathbf{x}|\mathbf{T}) &= [\cos \alpha > t,] \\ &= \left[\frac{\mathbf{t} \cdot \mathbf{x}}{\|\mathbf{t}\| \|\mathbf{x}\|} > t \right]. \end{aligned}$$

That can be interpreted as the *normalized cross correlation* (NCC).

2.3.3 OSP Detector

Considering that an observed spectrum is a mixture of the spectra corresponding to the materials in the pixel's footprint, and assuming linearity in the mixing process, all observed spectra should lie in the subspace spanned by the spectra of the materials in the scene. Thus, a linear subspace model should be useful, and a widely used detector is the *orthogonal subspace projector* (OSP) that uses a subspace model for the background [6]:

$$\begin{aligned} H_0 : \mathbf{x} &= \Phi_B \mathbf{a}_B + \mathbf{n}, & \mathbf{n} &\sim N(\mathbf{0}, \boldsymbol{\sigma}^2 \mathbf{I}) \\ H_1 : \mathbf{x} &= \Phi_B \mathbf{a}_B + \mathbf{t} + \mathbf{n} \end{aligned}$$

By removing the component of \mathbf{x} within the background subspace and matching with the target signature we get the detector

$$D_{OSP}(\mathbf{x}|\mathbf{B}, \mathbf{T}) = [\mathbf{t}^T \mathbf{P}_B^\perp \mathbf{x} > t]$$

where \mathbf{P}_B^\perp is the projection matrix for the subspace complementary to the background subspace and it's defined as follows [10]:

$$\begin{aligned} \mathbf{P}_B^\perp &= \mathbf{I} - \mathbf{P}_B \\ &= \mathbf{I} - \mathbf{B}(\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \end{aligned}$$

This detector is basically the same as the AMF detector in the subspace complementary to the background subspace [6] [10].

2.3.4 Preprocessing

For the AMF and SAM detector the hypothesis can be simplified in case the noise \mathbf{b} is colored (i.e. $\boldsymbol{\Gamma} \neq \boldsymbol{\sigma}^2 \mathbf{I}$) by whitening \mathbf{x} with respect to \mathbf{b} [6] [10],

$$\tilde{\mathbf{x}} = \boldsymbol{\Gamma}^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu})$$

giving us the simpler hypotheses for the respective detectors.

AMF detector:

$$\begin{aligned} H_0 : \tilde{\mathbf{x}} &= \tilde{\mathbf{b}}, & \tilde{\mathbf{b}} &\sim N(\mathbf{0}, \mathbf{I}) \\ H_1 : \tilde{\mathbf{x}} &= \tilde{\mathbf{t}} + \tilde{\mathbf{b}} \end{aligned}$$

$$\begin{aligned} D_{AMF}(\tilde{\mathbf{x}}|\mathbf{B}, \mathbf{T}) &= \left[\frac{\tilde{\mathbf{t}}^T \tilde{\mathbf{x}}}{\|\tilde{\mathbf{t}}\|^2} > t \right] \\ &= \left[\frac{(\mathbf{t} - \boldsymbol{\mu})^T \mathbf{\Gamma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}{(\mathbf{t} - \boldsymbol{\mu})^T \mathbf{\Gamma}^{-1} (\mathbf{t} - \boldsymbol{\mu})} \right] \end{aligned}$$

SAM detector:

$$\begin{aligned} H_0 : \tilde{\mathbf{x}} &= \tilde{\mathbf{b}}, & \tilde{\mathbf{b}} &\sim N(\mathbf{0}, \mathbf{I}) \\ H_1 : \tilde{\mathbf{x}} &= k\tilde{\mathbf{t}} + \tilde{\mathbf{b}} \end{aligned}$$

$$\begin{aligned} D_{SAM}(\tilde{\mathbf{x}}|\mathbf{T}) &= [\cos \alpha > t,] \\ &= \left[\frac{\tilde{\mathbf{t}} \cdot \tilde{\mathbf{x}}}{\|\tilde{\mathbf{t}}\| \|\tilde{\mathbf{x}}\|} > t \right] \\ &= \left[\frac{(\mathbf{t} - \boldsymbol{\mu})^T \mathbf{\Gamma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}{\sqrt{(\mathbf{t} - \boldsymbol{\mu})^T \mathbf{\Gamma}^{-1} (\mathbf{t} - \boldsymbol{\mu})} \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{\Gamma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}} \right] \end{aligned}$$

Strictly speaking, in order to whiten data, one possible method would be the following [11]:

1. Extract mean from the data in order to get zero-centered data.
2. Get eigenvalues and eigenvectors from the covariance of the zero-centered data:
 - (a) Solve the simultaneous equations to calculate ‘ λ ’ (i.e. eigenvalues): $\mathbf{\Gamma} - \lambda \mathbf{I} = 0$
 - (b) Substitute the eigenvalues ‘ λ ’ from above along with the covariance matrix ‘ $\mathbf{\Gamma}$ ’ to calculate the eigenvectors: $\mathbf{\Gamma} \mathbf{v} = \lambda \mathbf{v}$
3. Use the eigenvalues and eigenvectors computed for whitening the data using either PCA (principal component analysis method, see 2.2) or ZCA (zero component analysis method).

$$PCA: \quad \mathbf{W}_{PCA} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^T \mathbf{X},$$

$$ZCA: \quad \mathbf{W}_{ZCA} = \mathbf{U} \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^T \mathbf{X},$$

where $\mathbf{\Lambda}$: eigenvalues matrix, \mathbf{U} : eigenvectors matrix, \mathbf{X} : zero-centered data

2.4 Clustering Technique: K-Means

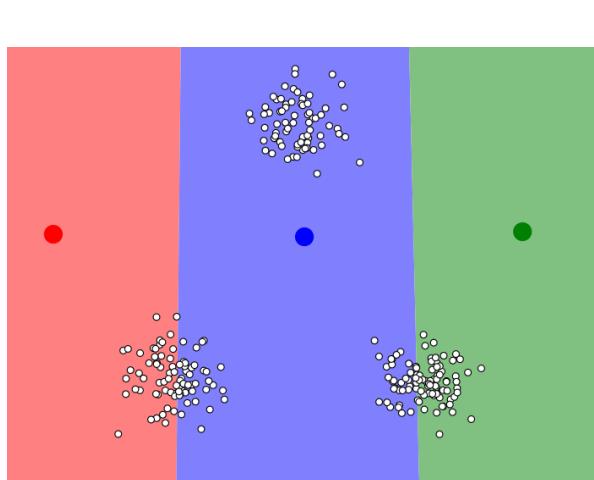
K-means clustering aims to partition data into k clusters in a way that data points in the same cluster are similar and data points in the different clusters are further apart [2].

The algorithm works as an iterative process. It is built on expectation-maximization algorithm. After a number of clusters are determined, it works by executing the following steps [2]:

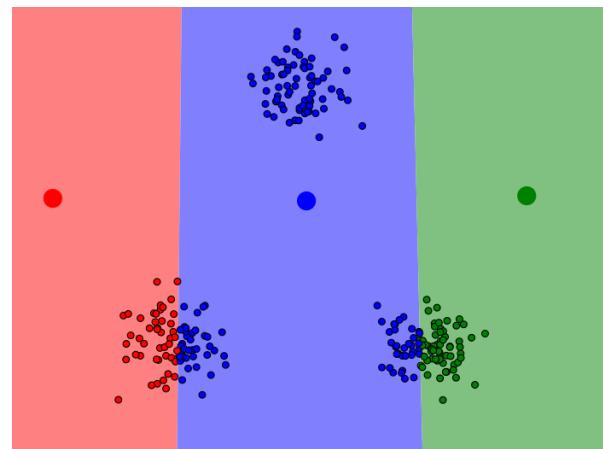
1. Randomly select centroids (center of cluster) for each cluster.

2. Compute the distance of all data points to the centroids.
3. Assign data points to the closest cluster.
4. Find the new centroids of each cluster by taking the mean of all data points in the cluster.
5. Repeat steps 2, 3 and 4 until all points converge and cluster centers stop moving.

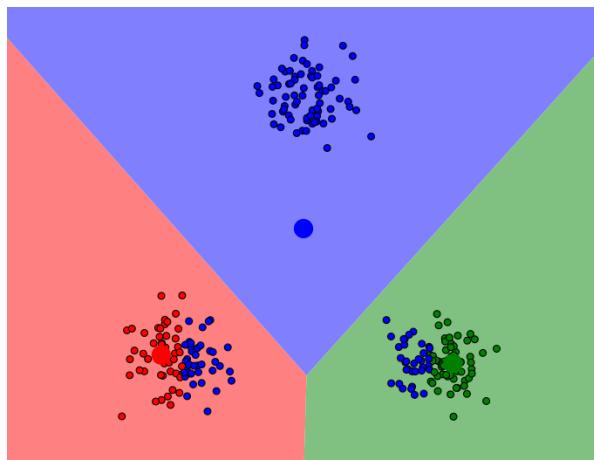
In order to be illustrative, a visual example of the steps mentioned can be found in Fig. 5:



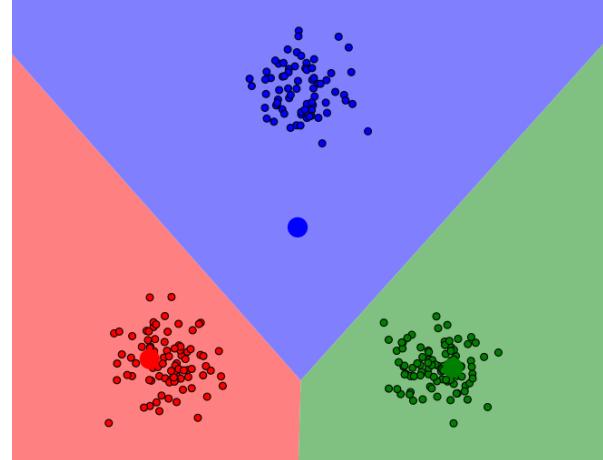
(a) Randomly select centroids.



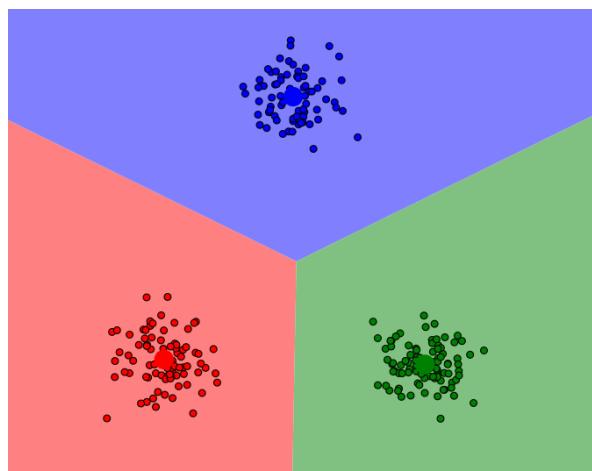
(b) Compute the distance of all data points to centroids and assign data points to the closest cluster.



(c) Compute the new centroids of each cluster by taking the mean.



(d) Re-assign data points to the closest cluster center.



(e) Model converges when cluster centers (centroids) no longer move.

Figure 5: K-Means algorithm, from [2].

2.5 Evaluation

In image processing, computer vision, big data, machine learning, etc; in order to measure performance, there are some equations and curves that are very useful for this purpose.

Given a ground-truth and predicted samples first we should compute a confusion matrix such as the one in Fig. 6 in order to get TP, FP, TN, and FN samples

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 6: Confusion matrix, from [3].

where [3]:

TP (True Positives): the number of samples where the classifier correctly predicts the positive class as positive.

TN (True Negatives): the number of samples where the classifier correctly predicts the negative class as negative.

FP(False Positives): the number of samples where the classifier incorrectly predicts the negative class as positive.

FN(False Negatives): the number of samples where the classifier incorrectly predicts the positive class as negative.

After obtaining those variables, we can compute performance measures such as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

For last, if we need to see the performance over different thresholds of the classifier and even compare that performance with other experiments, then Precision-Recall curves, ROC curves (TPR as a function of FPR) and F1-score curves could be useful.

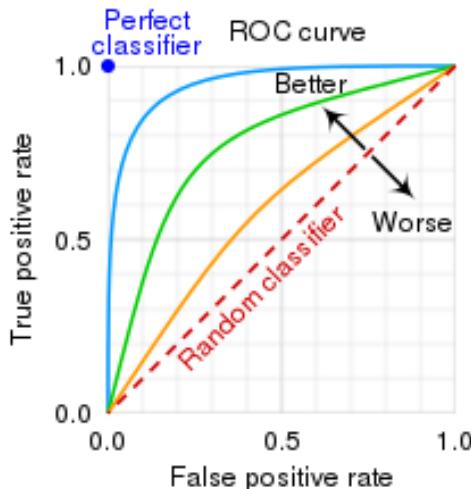


Figure 7: How performance is determined by ROC curves, from [4].

3 Methodology and Results

In this section, first we are going to pay attention at how the data has been collected: hardware used and procedure to acquire data. (Sec. 3.1, 3.2)

Then, we are going to focus on the software part by describing the method used to get quantitative results (see Sec. 3.3) and by explaining step by step the experiments made with the corresponding results of each one (see Sec. 3.4).

3.1 Hardware: Hyperspectral Camera

In order to acquire hyperspectral data a hyperspectral camera is needed for this purpose. For this project, the precise model used was a **Glana L4** which is shown in Fig. 8 with its pertinent specifications.



Glana L4	16 MP	29 MP
Optical format	35 mm	35 mm
Resolution	4864 x 3232	6576 x 4384
Pixel size	7.4 μ m	5.5 μ m
Framerate	12 fps	6 fps
Bit depth	14 bit	14 bit
Spectral range	450 – 850 nm	450 – 850 nm
Spectral bandwidth	~10 nm	~10 nm
Bands	96*	96*
Lens	Selectable	Selectable
Lens mount	Canon EF	Canon EF
Weight	~600 g**	~600 g**

* With normal (FWHM) overlap, 46 independent
** Camera body without lens

Figure 8: Glana L4 and its specifications, from [5].

Glana's patented solution consists of a state-of-the-art continuously variable optical filters (CVF) combined with advanced image reconstruction algorithms. This means superior spectral resolution in a shorter amount of capture time as well as a very high spatial resolution [5].

The CVF is a band-pass filter letting radiation pass only in a narrow wavelength band, centered at a wavelength λ_c . This center wavelength varies continuously along one of the dimensions (here called u) of the filter, so that the center wavelength depends on u only, that is $\lambda_c = ku$ for some constant k (see Fig. 9) [6].

As the CVF is mounted on, or very close to, the image sensor, the light registered by a sensor element at position (u, v) will only contain wavelengths close to $\lambda_c(u, v)$. When the camera and the observed surface are static, each point on the object will thus be observed in a specific wavelength. By rotating the camera, it can be used as a spectrometer. Alternatively, it can be used in the same way as a push-broom camera. In either case,

the scanning is made as in Fig. 10, and the data cube is assembled (and interpolated) by software [6].

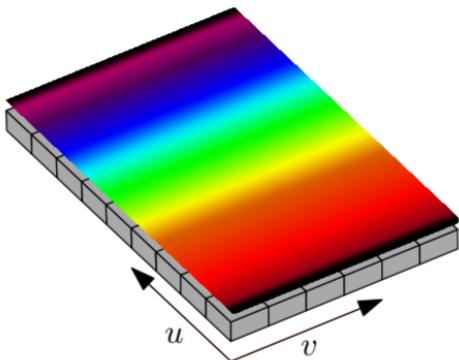


Figure 9: An CVF mounted on an FPA for hyperspectral imaging, from [6].

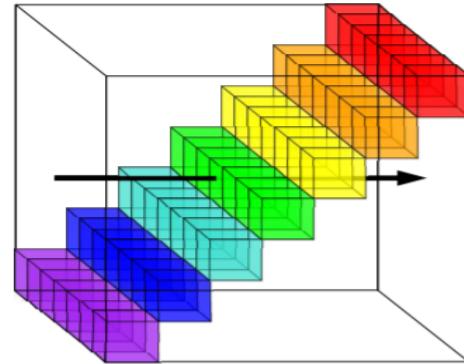


Figure 10: Spatio-spectral scanning, from [6].

3.2 Data collection

In this section, we are going to explain the data used in this project in order to make quantitative and qualitative experiments and how it has been collected.

3.2.1 Blood Data

In order to analyze some detectors and clustering like Sec. 3.4 do, at least, one hyperspectral image must be collected.

Since the essence of the project is to detect some targets that could be anomalies, a scenario had to be created in order to experiment with those targets. For that reason, the data collection took part in the middle of a forest and the scenario was created by throwing some pig blood traces into it that could serve as anomalies and adding as well some distractors such as a red mushroom, as shown in Figs. 11, 12:



Figure 11: Scenario with pig blood traces.



Figure 12: Little mushroom as a distractor.

After setting the scenario, the final step was to use the hyperspectral camera to capture the hyperspectral image of it. Part of the procedure is shown in Fig. 13



Figure 13: Calibrating the hyperspectral camera.

Every hyperspectral image from a camera like the Glana L4 camera is a merge of many spatirospectral images. The reconstructed blood hyperspectral image is created from around 200 spatirospectral images, its size is 2958x1270 pixels and contains 95 wavelengths.

This blood hyperspectral image serves as the baseline data in the following sections to make quantitative experiments. With the help of an app created by Glana Sensors AB, the hyperspectral image could also be shown as an RGB image, see Fig. 14.



Figure 14: Blood Baseline Image.

3.2.2 Spruces Data

In order to test and demonstrate qualitatively in a new data the methods implemented and analyzed in Sec. 3.4, Glana Sensors AB, provided us data about spruce trees captured from a drone which carried the Glana L4 (see Sec. 3.1).

The pictures were taken in a plantation field where baby spruces trees were growing around other vegetation which is very useful since other plants can serve as distractors. Thus, the target in this case is to detect spruces trees within all vegetation.

In order to see how the data was collected, the plantation field, the drone's flightmap and the drone itself are shown in Figs. 15, 16.

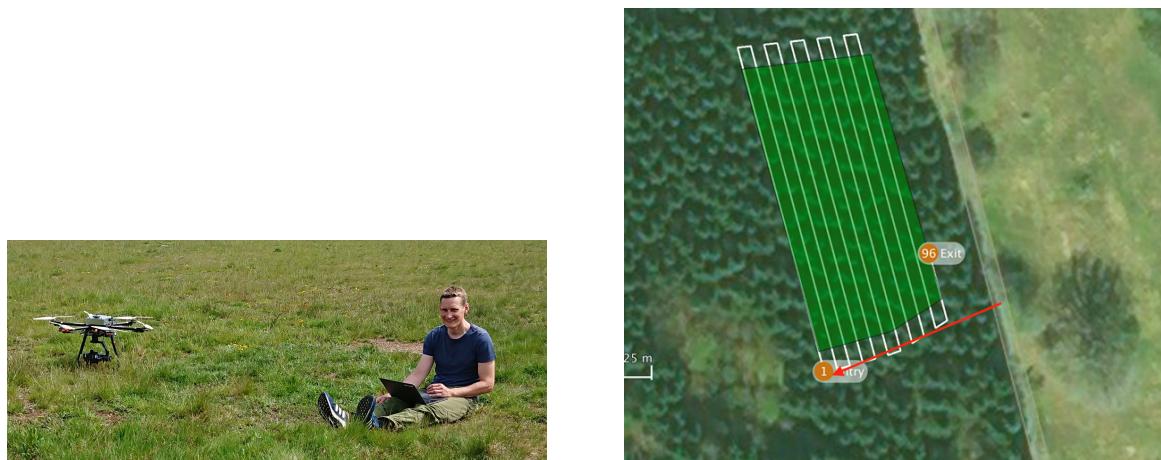


Figure 15: The drone and the drone flightmap.



Figure 16: Plantation field.

In this case, the resulting hyperspectral image that is going to be the one analyzed qualitatively in Sec. 3.4.8, is created from around 300 spatirospectral images, its size is 2000x3000 pixels and contains 24 wavelengths. The RGB version is shown in Fig. 17. Notice that this reconstruction introduces a bit of noise (black parts of the image).

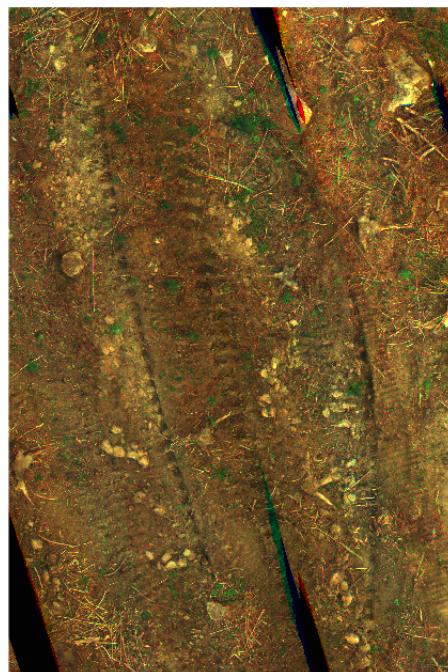


Figure 17: Spruces Baseline Image.

3.3 Evaluation

In order to achieve quantitative results with the blood baseline image (Fig. 14) so that we would be able to compare the different detectors and techniques, one logical thing to create is an ideal mask (which will act as the ground-truth) in order to separate the target pixels to detect from the rest. This ground-truth was created manually using morphological filters and Paint 3D. The ground-truth image is shown in Fig. 18.



Figure 18: Ideal mask of the baseline image.

In order to compare results with the ideal mask and extract from that comparison some

quantitative results, the steps to do are the following:

1. Get an image with the results of the detection.
2. Binarize this image by setting a threshold.
3. Compare pixel by pixel the ideal mask with the binarized image and extract TP, FP, TN and FN (see Sec. 2.5).
4. Compute Precision, Recall, FPR and F1-Score (see Sec. 2.5).
5. Repeat 3, 4, 5 with different thresholds.
6. Finally plot PR curve, ROC curve and F1-Score along the threshold values.

3.4 Software Development and Results

The first thing to mention is that all the software development has been coded in Python, starting from almost scratch.

I mean almost because, as the baseline of the project, Glana Sensors AB, not only provided data but also the code to be able to load the hyper-cube (see Fig. 4) from file. The hyper-cube is basically a 3D matrix (x, y, λ) that contains the value of each pixel in each band, which means that all the information about the hyperspectral image is inside this hyper-cube.

As seen in Sec. 3.2, the two reconstructed hyperspectral images used in this project contains too much data (2958x1270 pixels and 2000x3000 pixels respectively), that is why we have subsampled by 4 both images as preprocessing step. Another preprocessing step done is reshaping this hyper-cube in order to manage pixel-wise processing easier by having only 2D matrix (pixels, wavelengths).

In order to perform all the experiments, we should discuss first which pixel is it going to be **the target pixel**. This is really necessary in the context of a given application since the user must be allowed to pick any anomaly/target it considers in order to find similarities of it within the hyperspectral image.

From the baseline image (Fig. 14), we are going to choose as target pixel, one pixel inside the left blood pot that you can see in the lower right of the image. Assume this for all the quantitative experiments done.

Since AMF and SAM detectors are very similar compared to OSP detector, we are going to explain the software development by joining AMF and SAM and separating them from OSP.

3.4.1 AMF & SAM Detectors

After reshaping the cube, defining the target pixel and taking into consideration the equations of each detector (see Secs. 2.3.1, 2.3.2), we can make the first experiment where we create a new image where each pixel of the hyper-cube passes through the detector

and gets an output value. Then we can do the evaluation process as explained in Sec. 3.3. The results of this first experiment are shown in Fig. 19.

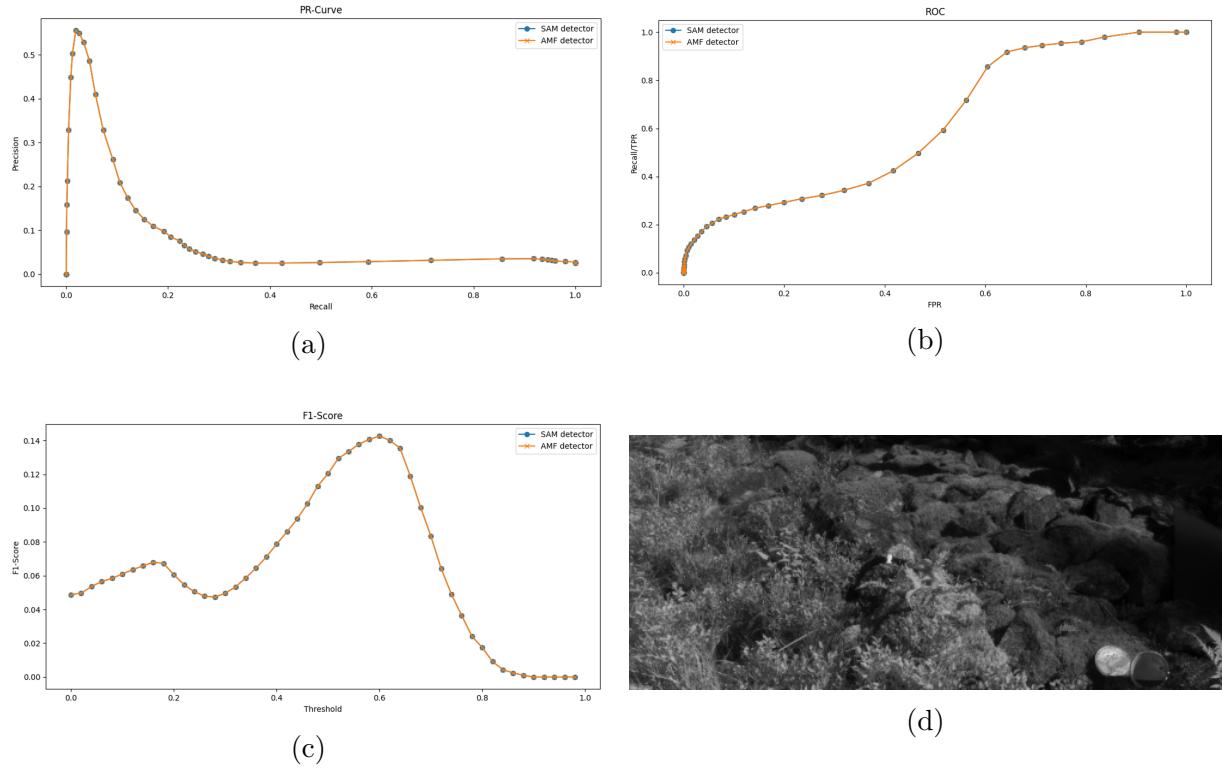


Figure 19: 19a, 19b, 19c shows evaluation metrics; 19d shows resulting image after detection.

As we can see, the PR curve decays really quickly, the ROC is almost flat and the F1-Score is really poor (note that both cases, SAM and AMF, obtains almost the same results, by the end of this section we will discuss why). Furthermore, it seems like in Fig. 19d it only detects the target pixel, its surroundings and the mushroom, even though it is almost unnoticeable. That could be because the noise is colored, so we should try to do whitening of the data as explained in Sec. 2.3.4. How the data is distributed before and after whitening can be seen in Fig. 20.

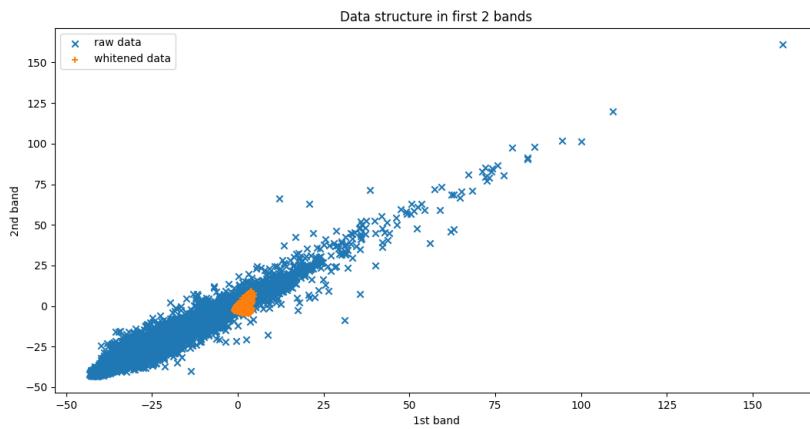


Figure 20: Comparing data structure before and after whitening with PCA in AMF.

As seen, we can verify that the noise is colored since the raw data seems to be correlated. The whitening removes the correlation of the data and thus the new data is uncorrelated taking as covariance matrix the identity (see Sec. 2.3.4). The results of doing whitening of the data are shown in Fig. 21:

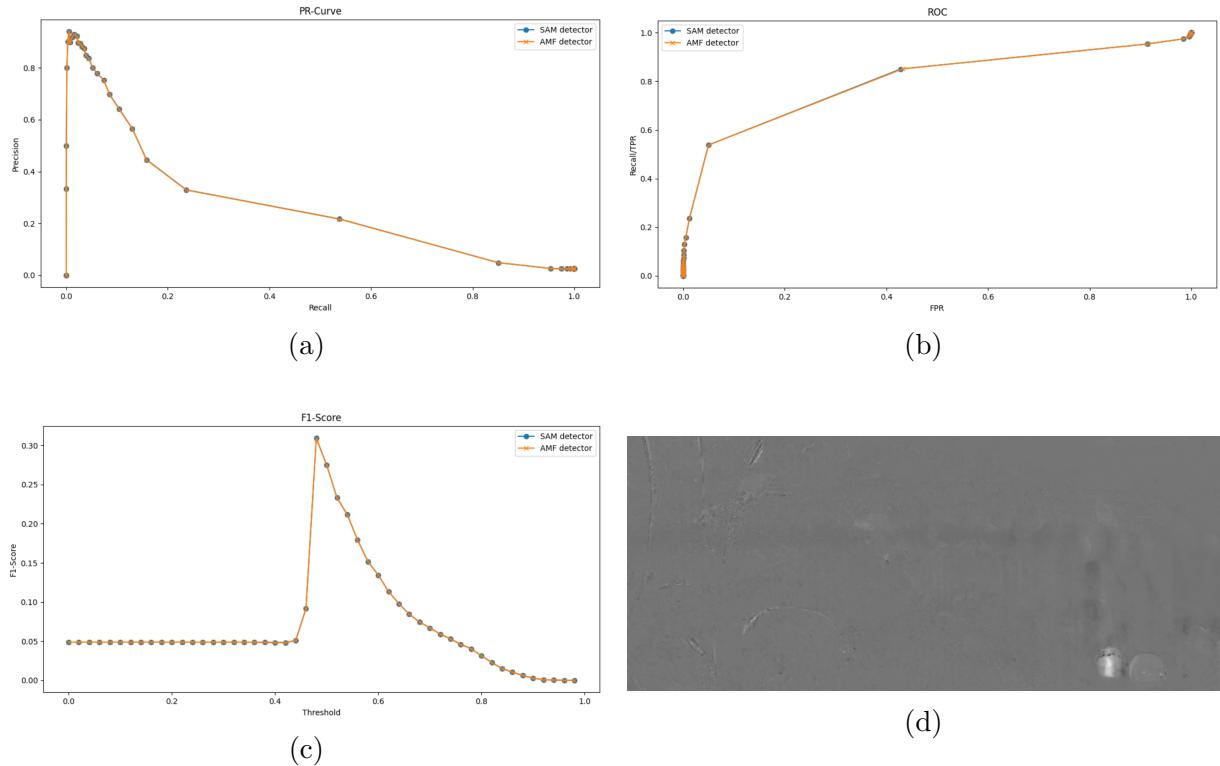


Figure 21: 21a, 21b, 21c shows evaluation metrics; 21d shows resulting image after detection.

Now, compared with the previous results, the PR curve decreases less quickly, the ROC is

less flat and we achieve a better F1-Score peak. On the other hand, looking into Fig. 21d it seems like the white noise appeared from the whitening process is predominant and prevents detecting the blood traces even better. That is because, another important aspect, is that we must take into account that not all the bands of the hyper-cube gives information. In fact, a lot of bands contain noise which should be reduced. It is important to remark that this noise comes from the background clutter and not the camera sensor. To reduce the noise, we compute the covariance matrix of the hyper-cube in order to find the eigenvalues and eigenvectors; and after having them, we pick only the eigenvalues with more power. That is basically use PCA (see Sec. 2.2) to reduce the dimensionality of the cube and keep with only useful information. That can be seen in Fig. 22.

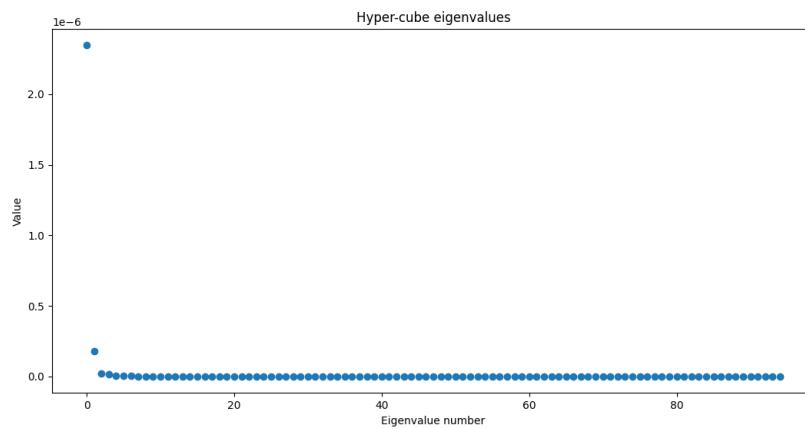
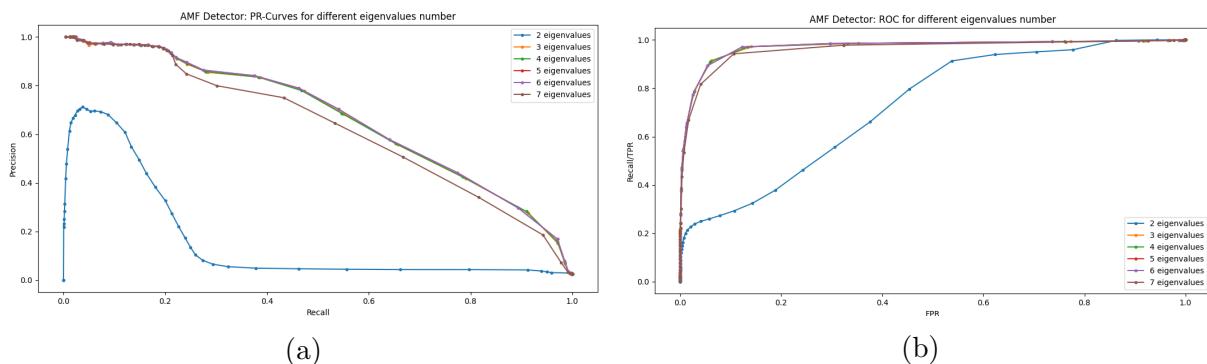


Figure 22: Hyper-cube eigenvalues.

So as Fig. 22 shows, picking between 2 and 7 eigenvalues is good to reduce the noise. So let us see what happens to the results having reduced the spectral dimensionality of the cube.



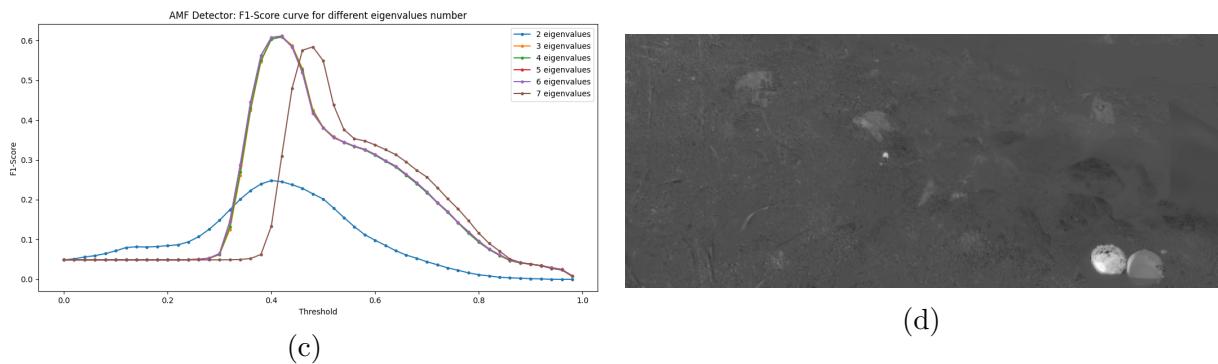


Figure 23: Results for AMF detector. 23a, 23b, 23c shows evaluation metrics; 23d shows resulting image after detection for the best result (5 eigenvalues).

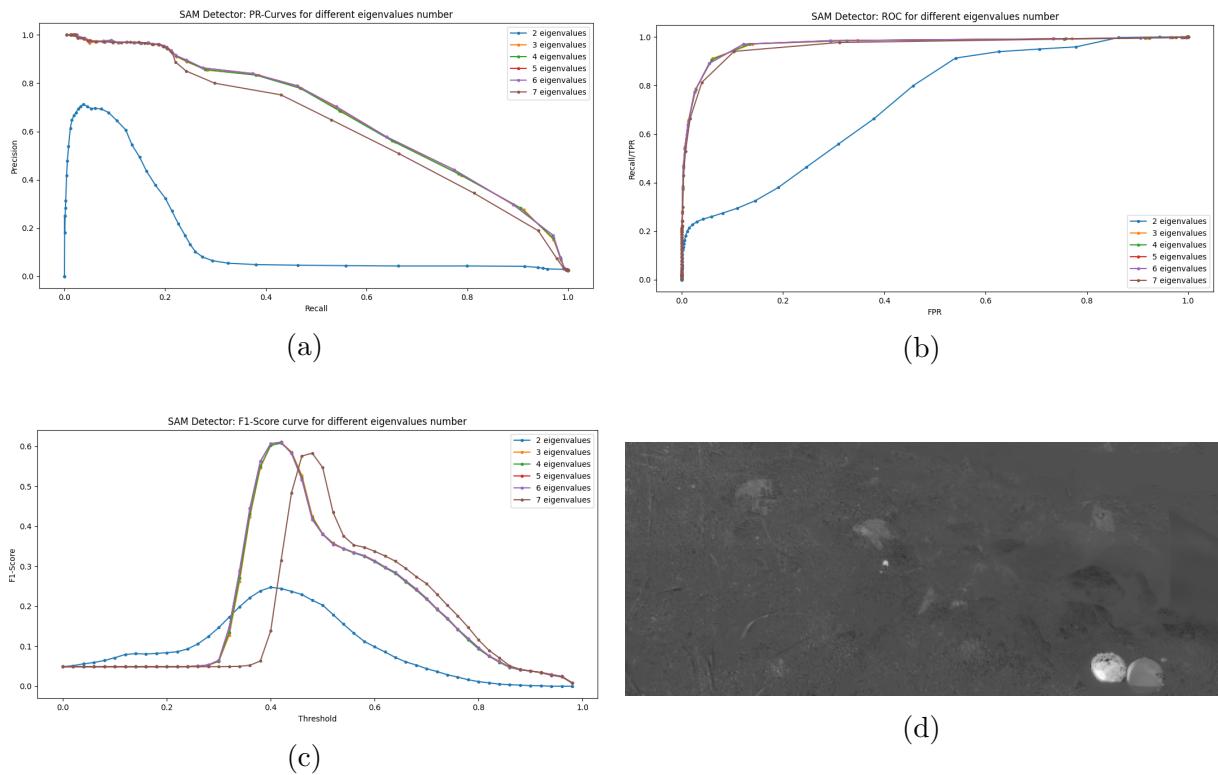


Figure 24: Results for SAM detector. 24a, 24b, 24c shows evaluation metrics; 24d shows resulting image after detection for the best result (5 eigenvalues).

In general, in all the experiments made, both AMF and SAM, have provided similar results. This is because, the only difference between both is when the target pixels contains different amounts of brightness, since SAM normalizes and thus treats all the pixels equally, whereas AMF treats the target pixels depending on its brightness.

As we can observe, picking between 3 and 6 number of eigenvalues, gives almost the same results which are pretty better than the previous ones. In Sec. 3.4.3, we will discuss a bit more the results presented.

3.4.2 OSP Detector

In the OSP detector case, taking into account the equations in Sec. 2.3.3, we should first find \mathbf{B} which is an $n \times b$ matrix spanning the b -dimensional background subspace.

The b -dimensional background subspace is basically a way to generalize how the background spectrum looks like in order to identify how much differs the target pixels from that subspace. We use it in order to get the projections in the complementary background subspace which is the one that is useful because it contains the pixels that are not part of the background subspace (which would be anomaly pixels) and thus be able to see which anomaly pixels are closer to the target pixel.

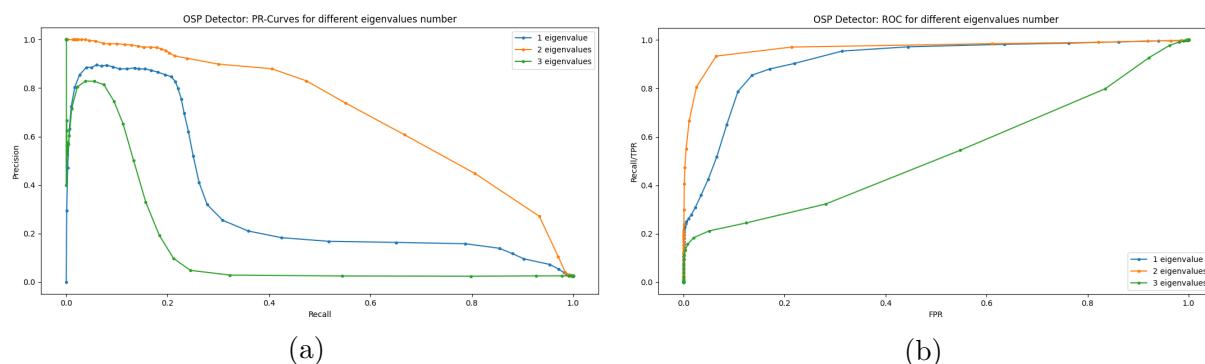
Typically, \mathbf{B} is guesstimated as being spanned by the first b principal components of the cube. That is equivalent to pick the first b eigenvalues as before and define the $n \times b$ matrix (\mathbf{B}) with the corresponding eigenvectors matrix. In other words, that is basically using PCA to find such background subspace (see Sec. 2.2).

We use PCA to find such background subspace because using PCA allows estimating better that background subspace since it keeps the important features of it. Furthermore, it saves a lot of computations since as seen in Sec. 2.3.3 we should invert and multiply some matrices and that can be computationally expensive if the dimensions are not reduced.

Here, there is no need of whitening the data as a preprocessing step since the essence of the OSP is getting the complementary background subspace and from it get the projections to see the similarity between the target pixel and the pixel in observation. If we whiten, we would remove the correlation and the background subspace will not be correct as it would include target pixels on it.

Then in the end, what we have is the projection on the target signature in the subspace complementary to the background subspace.

The results of using OSP detector are shown in Fig. 25:



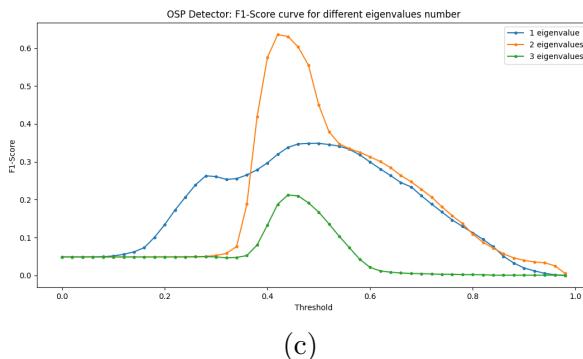


Figure 25: Results for OSP detector. 25a, 25b, 25c shows evaluation metrics; 25d shows resulting image after detection for the best result (2 eigenvalues).

As seen, in the OSP detector, compared to the other detectors, here taking one eigenvalue less or more makes a big difference.

3.4.3 Detection: Analysis & Discussion

To summarize the best results seen so far from F1-Score, let us take a look at Table 15

	AMF Detector	SAM Detector	OSP Detector
Raw	14.26%	14.26%	63.58%
Whitening	30.85%	30.92%	-
Whitening + PCA	61.13%	61.08%	-

Table 15: Results from detection.

As we can see, for the AMF and SAM detector, results are very similar since the target pixels might contain similar amounts of brightness. Apart from that, we can clearly say that doing whitening and reducing dimensionality with PCA works the best in detection. In case of the OSP detector, using the raw method without doing any preprocessing step works even better in detection than the other detectors. From now, for the following experiments assume that we use the detectors with the best results achieved (AMF & SAM with whitening + PCA).

In Figs. 23d, 24d, 25d, the blood traces are detected pretty well even though for instance, the mushroom is detected as well. That is why we should try to do clustering first, to see if we are able to allocate the mushroom and other noise items detected into a different cluster from the blood, and thus, improve the results by using the detectors only in the blood cluster.

3.4.4 K-Means

As told before, we use K-Means (see Sec. 2.4) as a clustering technique to improve results.

In order to illustrate the results of this section, we will show you:

1. A scatterplot where appears the data, the blood pixels and the cluster centers in the first 2 dimensions, in order to see where the cluster centers are allocated by K-Means with respect to the blood pixels obtained from the ideal mask.
2. The blood baseline image (Fig. 14) labelled as a result of the clustering in order to see which pixels are in the same cluster.

The first experiment made is to implement K-Means (with 8 clusters as it is not very large nor short number) using directly the raw data cube as an input. The results are shown in Figs. 26, 27.

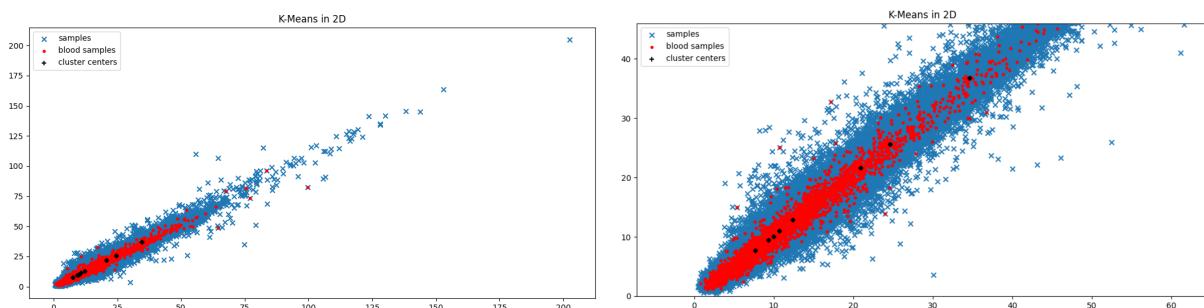


Figure 26: Scatterplot from the first 2 dimensions of the raw cube (without zoom and zooming).

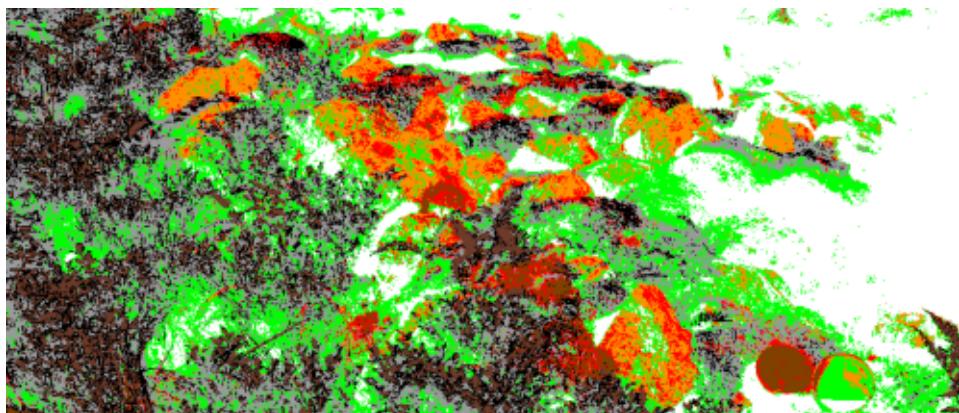


Figure 27: Baseline image with color labels for the 8 clusters obtained from K-Means.

As it can be seen, the results are not good since the raw data is correlated and thus the cluster centers takes values in a straight line. In fact, this line gives information about brightness of the pixels. Thus, what in the end in this case K-Means does is separate into different clusters, pixels from different brightness, not from different materials.

In order to solve this problem, what we should do is using PCA again (see Sec. 2.2) to reduce dimensionality but in this case, skipping the 1st principal component, since it is

the one that takes the information about brightness. Apart from solving that problem, we also reduce dimensionality before applying K-Means in order to remove unwanted features and save computations. For this following experiments assume we use the 3 first principal components after skipping the first one. The results of this experiment are shown in Figs. 28, 29.

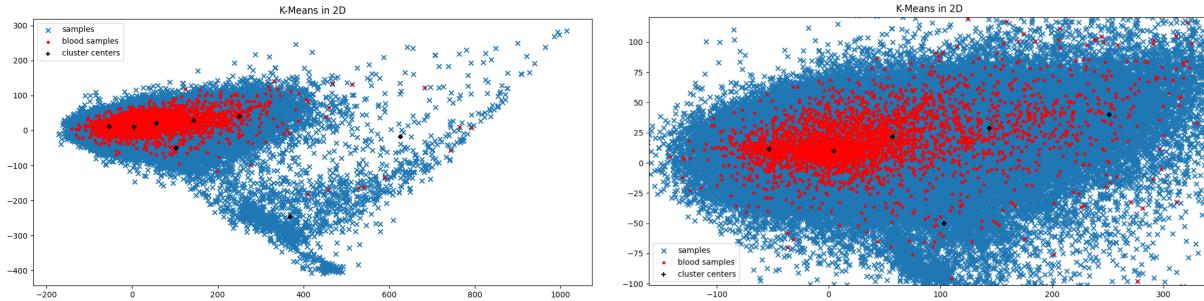


Figure 28: Scatterplot from the first 2 dimensions of the low-dimensional cube (without zoom and zooming).

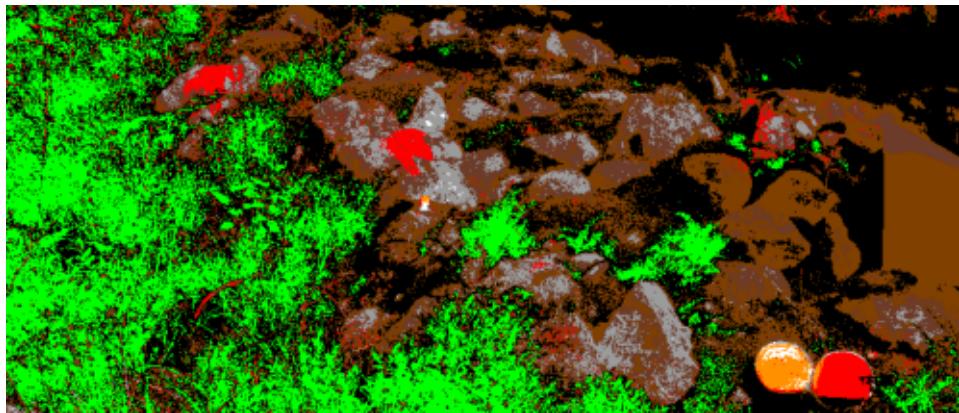


Figure 29: Baseline image with color labels for the 8 clusters obtained from K-Means.

As seen in the results, now K-Means works as expected by separating into clusters pixels from different materials/objects. Even so, the blood stills being in two separate clusters (orange and red labels) and the mushroom forms part of the orange cluster.

The first issue is not an inconvenient since later in the detection we can choose which clusters we want to apply detection. The second one is more difficult to resolve. One possible way, is applying K-Means with more clusters to see if it separates the mushroom from the blood. That is why we tried to see what happens if we use 9 clusters:

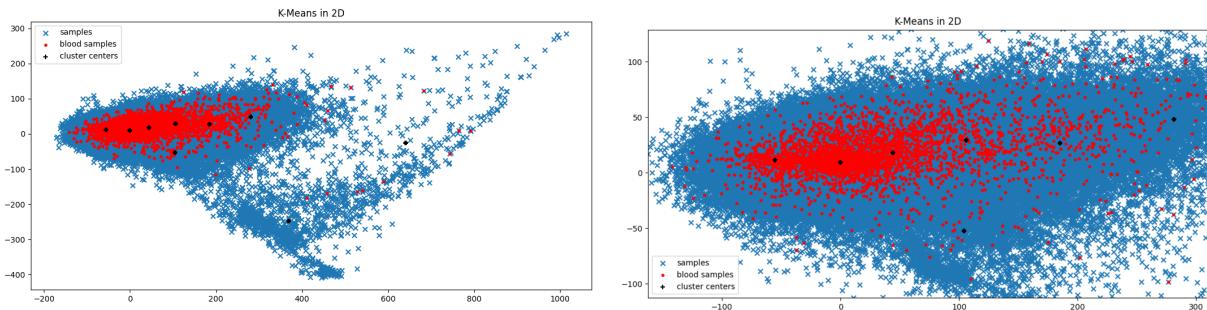


Figure 30: Scatterplot from the first 2 dimensions of the low-dimensional cube (without zoom and zooming).

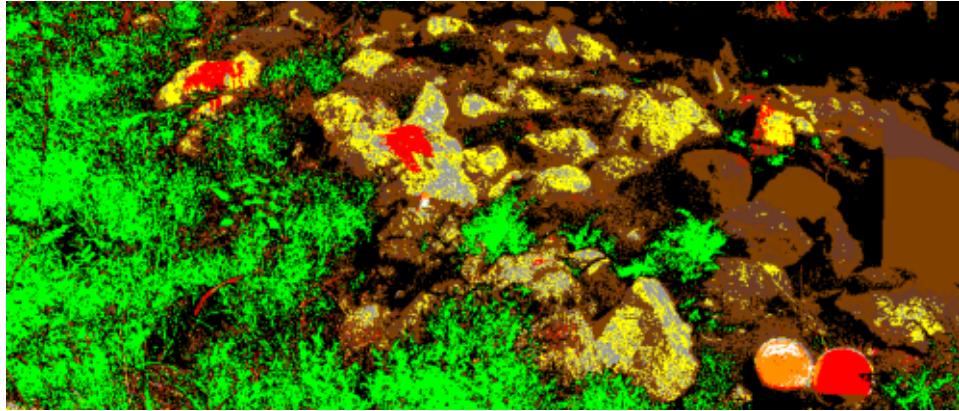


Figure 31: Baseline image with color labels for 9 clusters obtained from K-Means.

As seen, in the scatterplot it seems like it introduces the 9th cluster center outside the blood samples area, and thus, in the labelled image, the new cluster is nothing related with blood.

Despite we could still trying to apply K-Means with more clusters until the mushroom is separated from the blood (which in addition would increase the complexity of the software), since the essence is using K-Means combined with the detectors, we should trust that detectors are going to be the ones that separates the mushroom from the blood.

That is why in the end, assume for the rest of the project that we are going to use 8 clusters as in the beginning.

The first quantitative experiment that we could make is basically take the blood cluster pixels from the labelled image, create a mask with only that pixels and compare it with the ideal mask. This will allow us to obtain a baseline quantitative result by using only K-Means without detection. The resulting mask from the clustering is shown in Fig. 32 and the corresponding quantitative result is shown in Table 16.

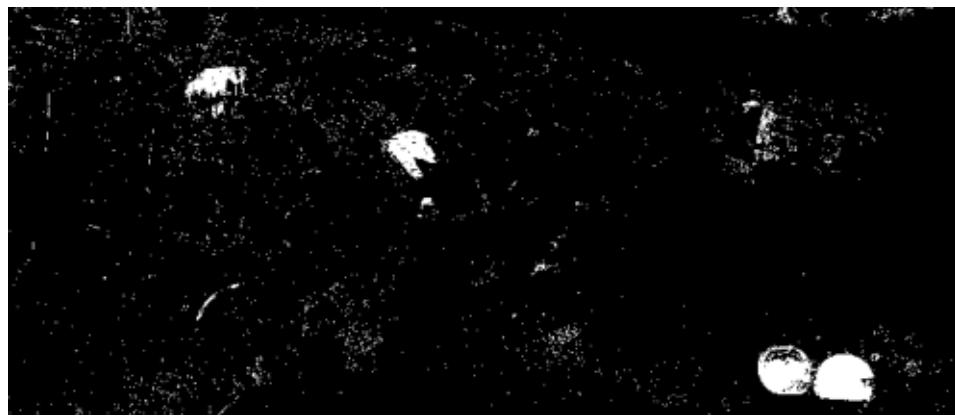


Figure 32: Resulting mask from taking only blood clusters.

3.4.5 K-Means + AMF & SAM Detectors

Since in Sec. 3.4.3 we saw that the best results were with whitening + PCA, now we are going to take advantage of reducing dimensionality of the cube before the clustering step (explained in Sec. 3.4.4) and use that low-dimensional cube with only the clusters selected (blood clusters), whiten it and proceed to the detection step. The results of this experiment are shown in Figs. 33, 34.

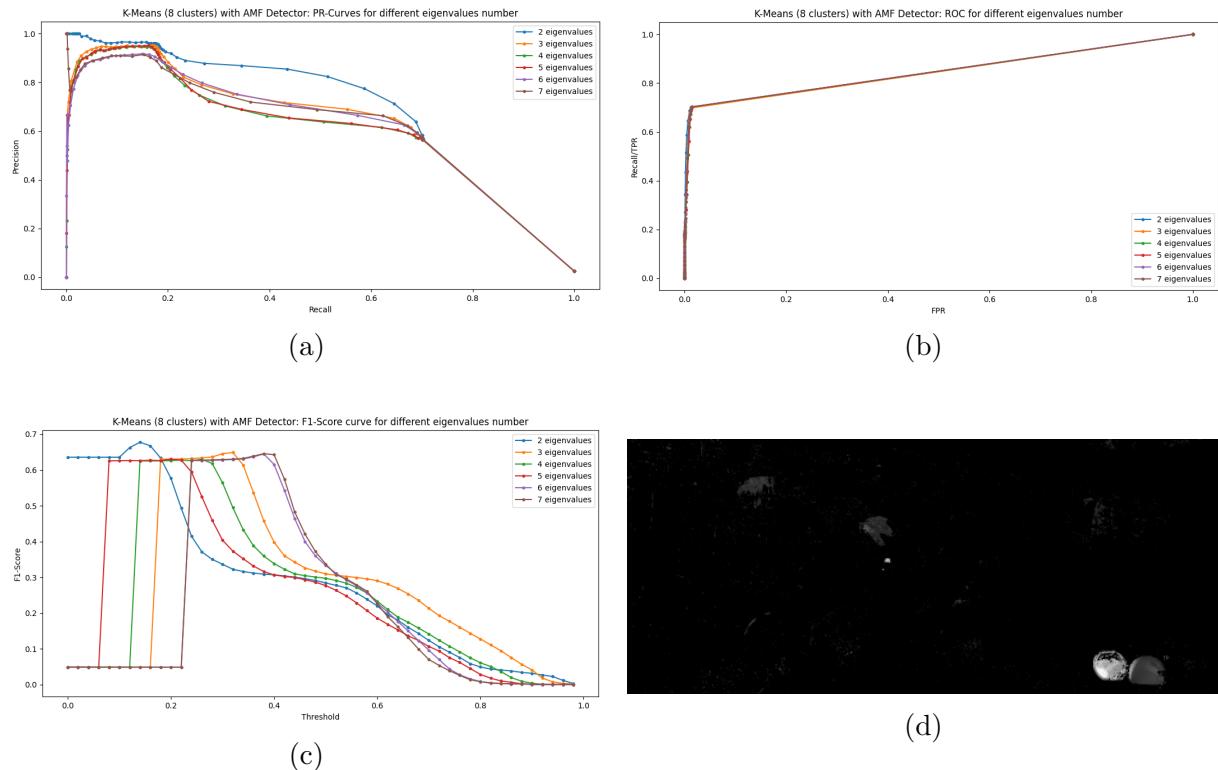


Figure 33: Results for AMF detector. 33a, 33b, 33c shows evaluation metrics; 33d shows resulting image after detection for the best result (2 eigenvalues).

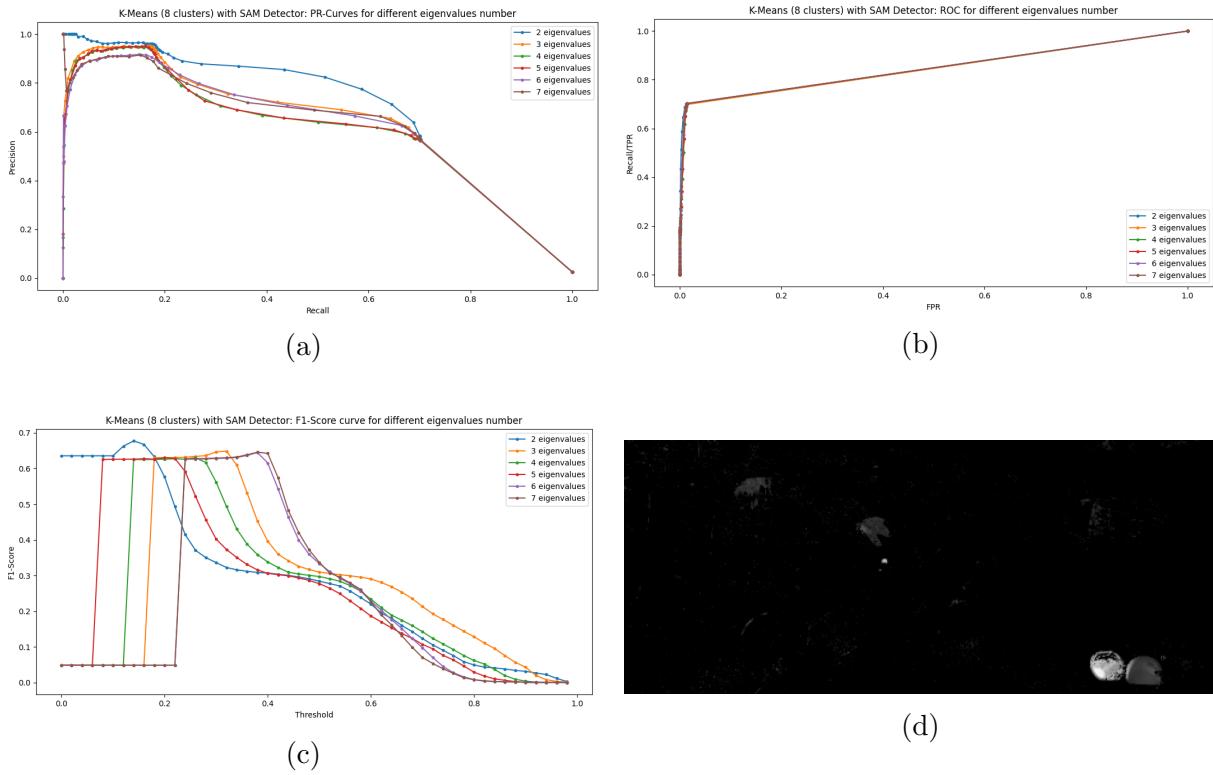


Figure 34: Results for SAM detector. 34a, 34b, 34c shows evaluation metrics; 34d shows resulting image after detection for the best result (2 eigenvalues).

The results again as in Sec. 3.4.1, between AMF and SAM are almost completely the same. If we focus on PR curves and ROC curves is interesting to see that with 2 principal components the PR curve is less flat than the rest but on the contrary the ROC-curves do not vary for any number of principal components. Compared to Sec. 3.4.1, here varying the number of eigenvalues makes a bigger difference in terms of quantitative results as seen in Figs. 33c, 34c.

3.4.6 K-Means + OSP Detector

In case of the OSP detector, here we apply the detection step to the raw cube instead of the low-dimensional cube as in the case of AMF & SAM detectors. That is because we will already use PCA to find the background subspace of the blood clusters in the detection step.

So now we have to take into account 2 parameters in order to search for the optimal results. On the one hand, we have the number of principal components used in order to reduce the dimensionality of the cube before applying K-Means and on the other hand we have the dimensionality of the background subspace used in the OSP detector.

The best results achieved are using the 2 first principal components (after skipping the first one) to reduce the dimensionality of the cube and 6 dimensions for the background

subspace. Let us see the results in Fig. 35.

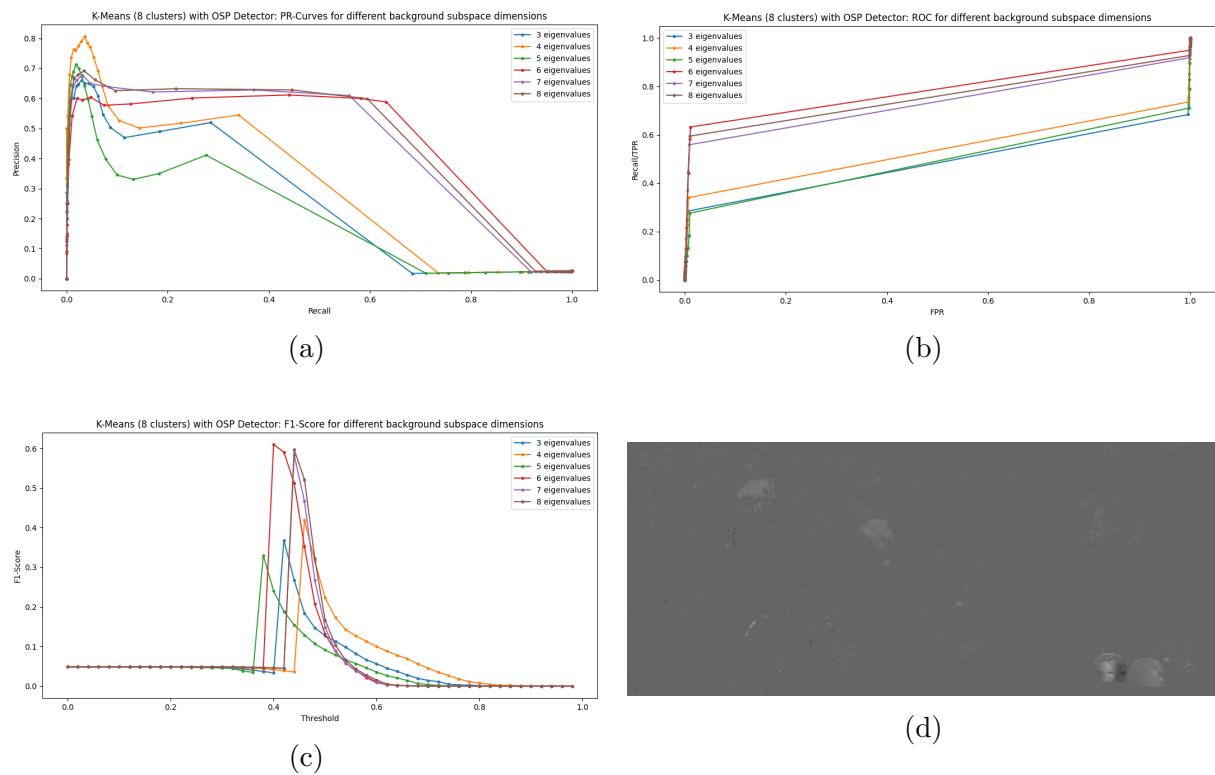


Figure 35: Results for OSP detector. 35a, 35b, 35c shows evaluation metrics; 35d shows resulting image after detection for the best result (6 dimensions of background subspace).

As seen, the results looks very poor compared to the AMF & SAM detectors and even with the results of only detection for the OSP case (see Sec. 3.4.2). The reason why is because while obtaining the background subspace, the fact of only having the blood clusters, has made an impact on obtaining that background subspace and thus, the detector has probably interpreted the blood pixels as the background subspace. Since the target pixel is a blood pixel too, then forms part of the background subspace and thus we get poor results because there is not any projection on the complementary background subspace.

In order to check the hypothesis that we are making, one good experiment to do now is changing the target pixel into the mushroom and running the OSP detector. In order to get quantitative results of this part, we have been forced to create an ideal mushroom mask too. That mask is shown in Fig. 36

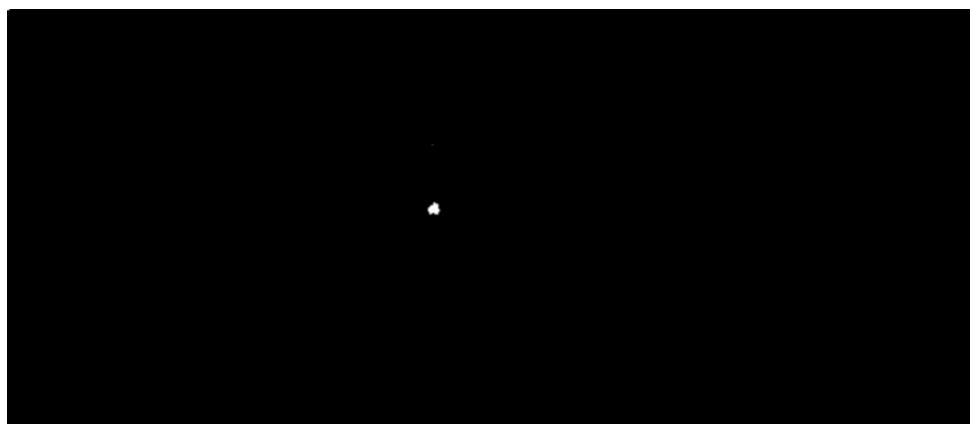


Figure 36: Ideal mushroom mask.

And the pertinent results are shown in Fig. 37.

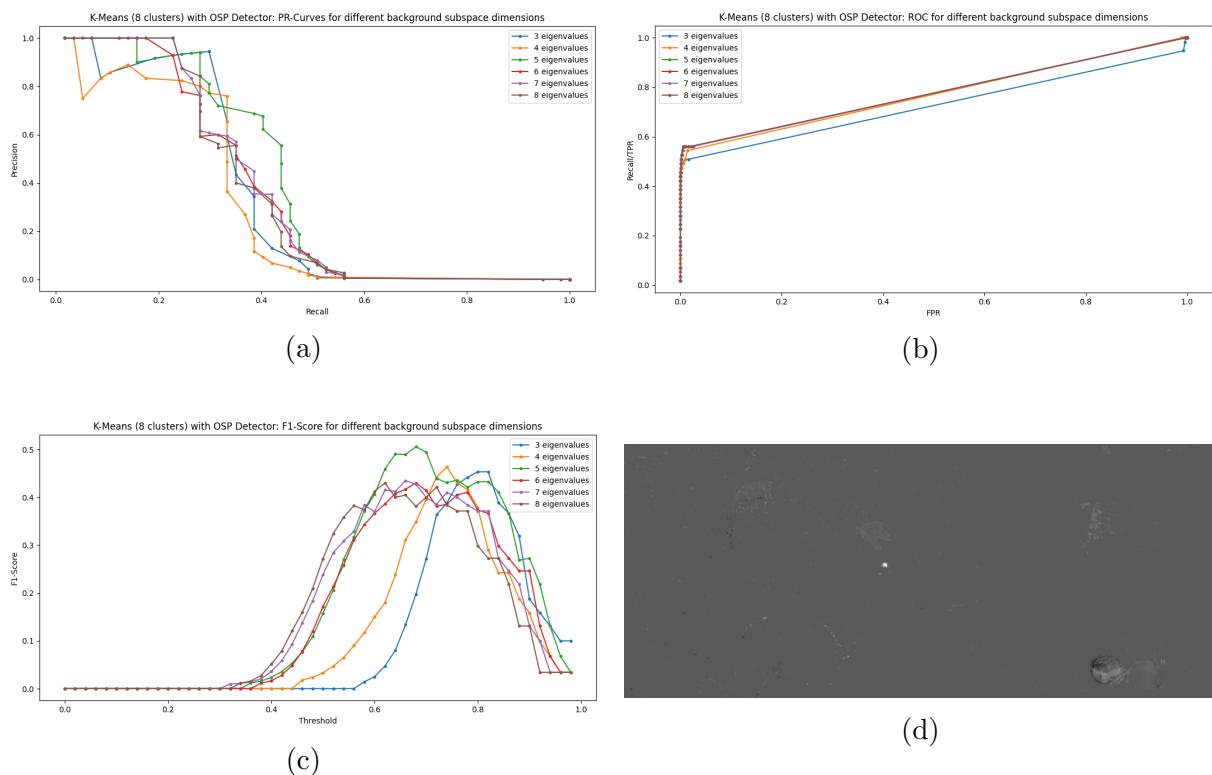


Figure 37: Results for OSP detector. 37a, 37b, 37c shows evaluation metrics; 37d shows resulting image after detection for the best result (5 dimensions of background subspace).

As seen in Fig. 37d, the mushroom is extremely well detected which means we are in the right hypothesis and thus, the mushroom is not contained inside the background subspace

and is treated as an anomaly. That is really important because it has been the 1st time we have been able to separate the mushroom from the blood even though we have needed to change the target pixel to the mushroom.

If we pay attention to Figs. 37a, 37b, 37c, curves results does not look really well apparently. That is because there are very few positives in the mushroom mask which means it penalizes more than usual having any false positives or false negatives.

3.4.7 Clustering + Detection: Analysis & Discussion

To sum up the results seen in clustering + detection and analyze those, let us see first Table 16.

	F1-Score
K-Means	62.97%
K-Means + AMF Detector	67.74%
K-Means + SAM Detector	67.73%
K-Means + OSP Detector	60.96%

Table 16: Results from clustering + detection.

The first thing to mention is that comparing the results to Table 15, seems like applying K-Means without doing any detection later it is already a better way than applying for instance AMF & SAM detectors. That means, using only K-Means can be an useful way of detecting targets/anomalies.

As clearly seen, even though in Figs. 33d, 34d seems like the mushroom stills being detected, AMF & SAM detectors combined with clustering, provides the best quantitative results of all project since K-Means helps to remove all useless information, possible noise and allows focusing this detectors only in the cluster/s where the target appears which gives more effectiveness. On the contrary, OSP detector it is a bad choice if the clustering itself realizes a good detection because then as said before, the background subspace tends to be the target pixels.

3.4.8 Testing on new data

Taking into account the quantitative results seen in Secs. 3.4.3, 3.4.7, as last experiment, we are going to use new data to test and demonstrate qualitatively that the best methods implemented works on new data. We are going to use as new data, the spruces data (see Sec. 3.2.2).

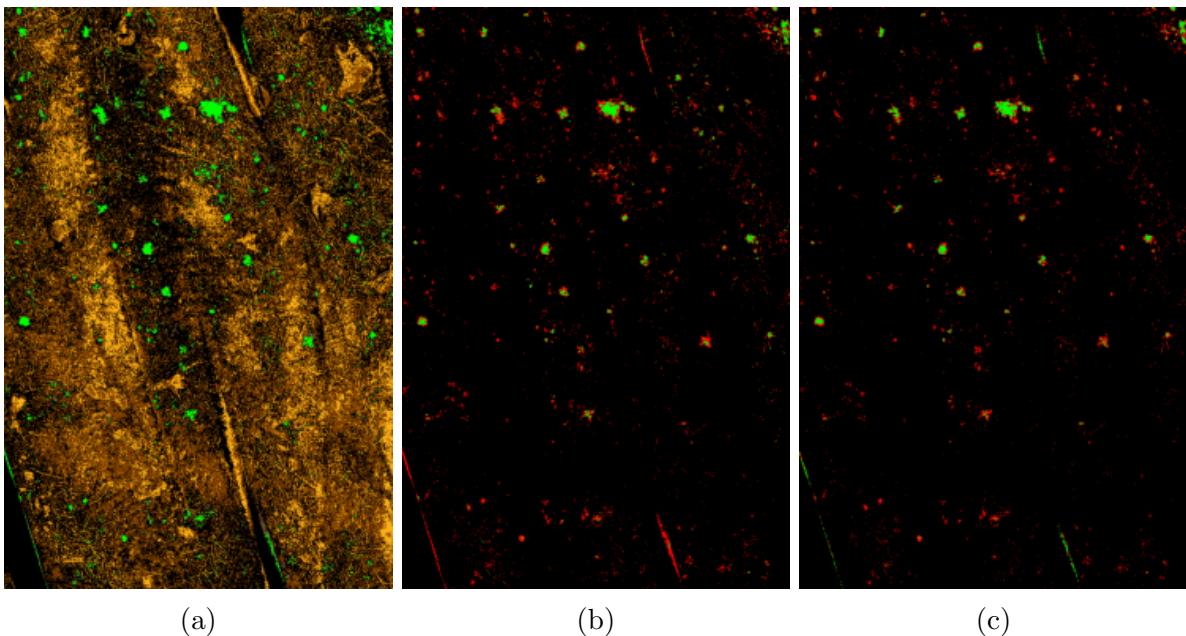


Figure 38: Qualitative results. 38a shows K-Means results, 38b shows K-Means + AMF/SAM results, 38c shows K-Means + OSP results.

Taking the spruces baseline image data (see Fig. 17), first we applied K-Means with 8 clusters (see Fig. 38a), and as consequence, K-Means grouped all vegetation with a bit of noise (green cluster) coming from the reconstruction process of the hyperspectral spruce baseline image. Then, picking only the green cluster, we applied the detectors on it by selecting a spruce tree as target. For last, we set manually a threshold that decides whether to consider each pixel as a spruce (green pixels) or not (red pixels), see Figs. 38b, 38c. The thresholds values are **0.47** for AMF/SAM case and **0.76** for the OSP case.

As seen in Fig. 38, we have used K-Means + AMF/SAM detector, since they have been the ones that have given the best results, and K-Means + OSP detector, even though it has not demonstrated good results, because in this case, there are a lot of distractors in the scene (e.g. other vegetation) that forms part in the same cluster of the spruces after applying K-Means as seen in Fig. 38a. Thus, we might obtain better results since K-Means does not distinguish between different kind of plants and thus the background subspace is not going to include in majority target pixels.

Looking at Figs. 38b, 38c, even though we do not know exactly if the results are optimal and if there is any missclassification, at least, we can qualitatively say that both methods gives pretty similar classification results although AMF/SAM detector seems to work a bit better. The reason why is because AMF/SAM catches the noise (tiny lines that appears) created by the reconstruction of the spruces baseline image, and on the other hand, OSP detector interprets parts of this lines as spruces. In any case, K-Means + OSP Detector has demonstrated that when K-Means does not provide a good clustering, then paradoxically improves its results.

In order to get the best qualitative results, we have had to retune some parameters. In the AMF & SAM case, it has not been necessary since we use again (as in Sec. 3.4.5) the 2 first principal components (after skipping the 1st one) to reduce the dimensionality of the cube before applying K-Means. On the contrary for the OSP case respect to Sec. 3.4.6, now the best qualitative results have been achieved with 3 first principal components (after skipping the 1st one) to reduce the dimensionality of the cube before applying K-Means and 2 dimensions for the background subspace.

4 Budget

In this section, we are going to make an estimation of the total budget necessary to carry on this project.

First, we should take into account the wage costs. This project is made by one undergraduate person who under this conditions earns 9€/hour. Then, we should take into account that the duration of the project has been 20 weeks (5 months) and the person has worked 25 hours/week.

Another aspect to take into consideration are the material costs. In this project, we have used as hardware a drone, a hyperspectral camera and a laptop. Since we have used this materials for almost half a year, and its annual depreciation is around 20% of the purchased price, then the materials costs are going to be the 10% of the purchased price. All the software used includes free license since the project has been coded in Python.

The wage and material costs are depicted in Table 17:

Concept	Amount
Undergraduate engineer	9€/hour · 25hours/week · 20 weeks = 4.500€
Drone usage	18.000€ · 10% = 1.800€
Hyperspectral camera (Glana L4) usage	15.000€ · 10% = 1.500€
Laptop usage	2.500€ · 10% = 250€
TOTAL:	8.050€

Table 17: Wage and material costs.

For last, we should include data storage costs since a hyperspectral image contains a lot of data. We have used two hyperspectral images that occupies **1.4 GB** and **0.56 GB** respectively. That makes a total data storage cost of **1.96 GB**.

To sum up this section, the total budget necessary to carry on this project is **8.050€** and also, at least, the project requires **1.96 GB** of free memory.

5 Conclusions

In order to start the project, as first project goal, perform a literature study of methods for analysis of hyperspectral images has been necessary. We have arrived to the conclusion of using AMF, SAM and OSP detectors since they are not very complex detectors and, in addition, they are useful for finding similarites between data, given a target.

With provided data in a hyper-cube to analyze each method and get quantitative results, we have implemented first AMF and SAM where it has been necessary:

1. To do some preprocessing such as whitening data because the noise was colored.
2. To reduce dimensionality using PCA in order to keep with only useful information since a lot of bands of the hyper-cube contained noise coming from the background clutter.

AMF and SAM results have been almost the same in all project since we have seen that both are really similar detectors that only differs when the target pixels contains different amounts of brightness.

After that, we have implemented the OSP Detector, where we have used again PCA to find the background subspace since it estimates better this and saves computations.

The results of this part have been shown in Table 15, where OSP detector has given the best results of all detectors. Even so, some distractors and noise items have continued to be detected as seen in Figs. 23d, 24d, 25d.

That is why, we have used machine learning in this project using K-Means as clustering technique, since it is a traditional method that allows to segment data in an easy way that could help us to allocate distractors, useless information and noise items into a different cluster from the target and thus, lead us to improve the results by applying the detectors only in the target cluster.

That worked for the K-Means + AMF/SAM Detector as seen in Table 16, since they have achieved the best quantitative results of all the project. On the contrary, K-Means + OSP Detector has given poor results due to the fact that K-Means provides a good clustering and thus, the background subspace is plenty of target pixels. This good clustering reflects that it is possible to use K-Means as a detector. In fact with this data, K-Means obtains better results than using AMF and SAM detectors.

Also, we have seen that in order to avoid K-Means separate into clusters pixels from different brightness instead of materials, we must use PCA again (before using K-Means) by skipping the 1st principal component since it is the one that provides information about brightness.

As last project goal, we have used forestry data in order to test and demonstrate qualitatively that the methods with best results works as well on new data. In fact, with this new data, we have seen that K-Means + OSP Detector works better than with the blood data, since in that forestry data, K-Means does not provide a good clustering. Nevertheless, with this new data, it has seemed K-Means + AMF/SAM Detector has given better qualitative results as well.

6 Future Work

While the project was being developed, some experiments that were not possible to implement it because of lack of time, came to our minds.

The main one that should be done in order to follow the project is implementing the modified GAN [7], since it is a recent advance in machine learning that is useful to detect anomalies and to mitigate the problem of anomalies contamination in the training data. Since it is a deep learning approach it would require larger datasets.

Also another possibility could be, for instance, using GMM's instead of K-Means as clustering technique and see if it improves the results (or any other clustering technique).

The rest of possible experiments that could improve the results are:

- Use ZCA (see Sec. 2.3.4) instead of PCA for whitening in AMF & SAM case.
- Use K-Means with more clusters until all distractors are separated from the target (although it would increase complexity).
- When K-Means itself already provides a good clustering, then in K-Means + OSP Detector, use more clusters in detection instead of using only the target clusters in order to not having plenty of target pixels in the background subspace.
- Also in K-Means + OSP Detector, use the low-dimensional cube instead of the raw cube for the detection step.

References

- [1] Nikhil Gangwar. Hyperspectral image reconstruction from rgb image. <https://ai.plainenglish.io/hyperspectral-image-reconstruction-from-rgb-169bc62a387f>, 2020.
- [2] Soner Yildirim. K-means clustering — explained. <https://towardsdatascience.com/k-means-clustering-explained-4528df86a120>, 2020.
- [3] Joydwip Mohajon. Confusion matrix for your multi-class machine learning model. <https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826>, 2020.
- [4] Wikipedia. Receiver operating characteristic. https://en.wikipedia.org/wiki/Receiver_operating_characteristic.
- [5] Glana Sensors AB. Glana l4. <https://glanasensors.org/>.
- [6] Jörgen Ahlberg, Amanda Berg, and Claes Nelsson. Thermal and multispectral imaging, 2018.
- [7] Amanda Berg, Michael Felsberg, and Jörgen Ahlberg. Unsupervised adversarial learning of anomaly detection in the wild. *IOS Press*, 2020.
- [8] Hannes Freij. Hyperspectral image registration and construction from irregularly sampled data, 2021.
- [9] Natalie Syrén Grönfelt. Pretraining a neural network for hyperspectral images using self-supervised contrastive learning, 2021.
- [10] Jörgen Ahlberg and Ingmar Renhorn. Multi- and hyperspectral target and anomaly detection, 2004.
- [11] Learndataa. Data preprocessing: Whitening or sphering in python. <https://learndataa.com/2020/09/15/data-preprocessing-whitening-or-sphering-in-python/>, 2020.