Yaniv Mazor 315635722
Ayalla Reuven 314077033

Project report: MF for PCA bands:

In our project, we investigated the impact of the PCA process on the MF algorithm.

Our project involved examining the influence of the PCA procedure on the MF algorithm. We aimed to comprehend the significance of each stage and determine how we can forecast the algorithm's performance prior to its implementation. Furthermore, we observed the functionality of PCA with various filters. Our primary innovation was utilizing the PL_0MF for the PCA space. We focused on the data's stationarity and attempted to enhance the efficiency of the PCA space by incorporating local properties of each pixel.

The rest of the report going as follows:

- Short discussion on hyperspectral data and target detection & data 's Introducing
- PCA algorithm: whiting noise and compression
- Compression properties: eigenvalues, histogram & variances
- MF procedure and performances
- Innovation: Data stationarity and other algorithm performances: PL_0MF, ACE
- Conclusions

Yaniv Mazor 315635722
Ayalla Reuven 314077033

1. Hyperspectral data, target detection & Data Introducing:

Introduction:

Hyperspectral imaging finds applications in civil, environmental, and military domains. It is used for tasks such as detecting terrain features, detecting specific plant species, and identifying military vehicles for defense and intelligence purposes. [1]

Hyperspectral imaging sensors capture digital images in numerous narrow spectral bands, spanning the visible, near infrared, and mid-infrared spectrum. This allows for the construction of a continuous radiance spectrum for each pixel in the scene.[2]

In applications of hyperspectral target detection, which we are interested at, our objective is to determine the presence or absence of a rare object with a known spectral signature within the captured scene. The term "rare" indicates a relatively small quantity compared to the total number of pixels, such as a few pixels in an image consisting of millions. [2]

Data:

Our project will concentrate on the analysis of the **"Viareggio 2013 Trial"** Data, which is extensively documented in [3].

The specific data cube we will be working with, D1_F12_H1, consists of 511 bands with dimensions of 375 by 450 samples.

Detailed information about the conditions and targets transplanted into the image can be found in [4]. The "Viareggio 2013 Trial" encompassed various operations, including data detection and comparison with other data from the same area at different times.

---

12 Detection Algorithms in hyperspectral Imaging Systems, Dimitris Manolakis, Eric Truslow, Michael Pieper, Thomas Cooley, and Michael Brueggeman, IEEE SIGNAL PROCESSING MAGAZINE JANUARY 2014

2 Detection algorithms for hyperspectral imaging applications, D. Manolakis; G. Shaw, IEEE Signal Processing Magazine , January 2002

3 Hyperspectral Airborne "Viareggio 2013 Trial" Data Collection for Detection Algorithm Assessment, Nicola Acito, Alessandro Rossi, Marco Diani, IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, JUNE 2016

In our case, we just aim to investigate the feasibility of compressing the data and its impact on different target detection methods.

2. PCA algorithm: whiting noise and compression:

Noise whitening:

As we have seen in the class, the PCA algorithm is done by multiplying the data in the eigenvector matrix, E^T, which contains in each row the eigenvector of ϕ_{data}. Note that each row belong to specific eigenvalue, and the eigenvalue sorted in decrease manner.

The problem is the noise. For additive noise model, the projection of the eigenvectors can be non-optimal in terms of SNR. For that reason, we would like to whiten the noise before applying eigenvectors projection on our data.

The procedure of noise whiting is done by multiply each data's pixel in the matrix:

$$\Lambda_N^{-0.5} E_N^T$$

Where Λ_N is the diagonal matrix of the eigenvalue on the noise covariance matrix, and E_N is the matrix of the eigenvectors of the noise covariance (each column represents vector).

Just after doing so, we can project the new data (we will call it white data)

on its eigenvectors.

The noise's covariance matrix was calculated using the m_8 theorem[4]:

$$\Phi_N = \Phi_{x-m_8} = \frac{1}{Pixels^*} \sum_i \left(\overrightarrow{x_\iota - m_{8_\iota}}\right)\left(\overrightarrow{x_\iota - m_{8_\iota}}\right)^T$$

*per band

Now, we have white noise data, and we can be sure that project the data on the eigenvectors of its covariance matrix, will yield us compressed data (we will

[4] Analysis of false alarm distributions in the development and evaluation of hyperspectral point target detection algorithms

discuss the meaning of compression later) which is optimal in SNR terms, since now our noise doesn't depend on direction.

Compression:

So now, given the white data, we only need to calculate the data's covariance, and multiply the data on our E_{white}^{T} matrix, doesn't we?

One of the questions we ask ourselves was the meaning of M_{white} matrix.

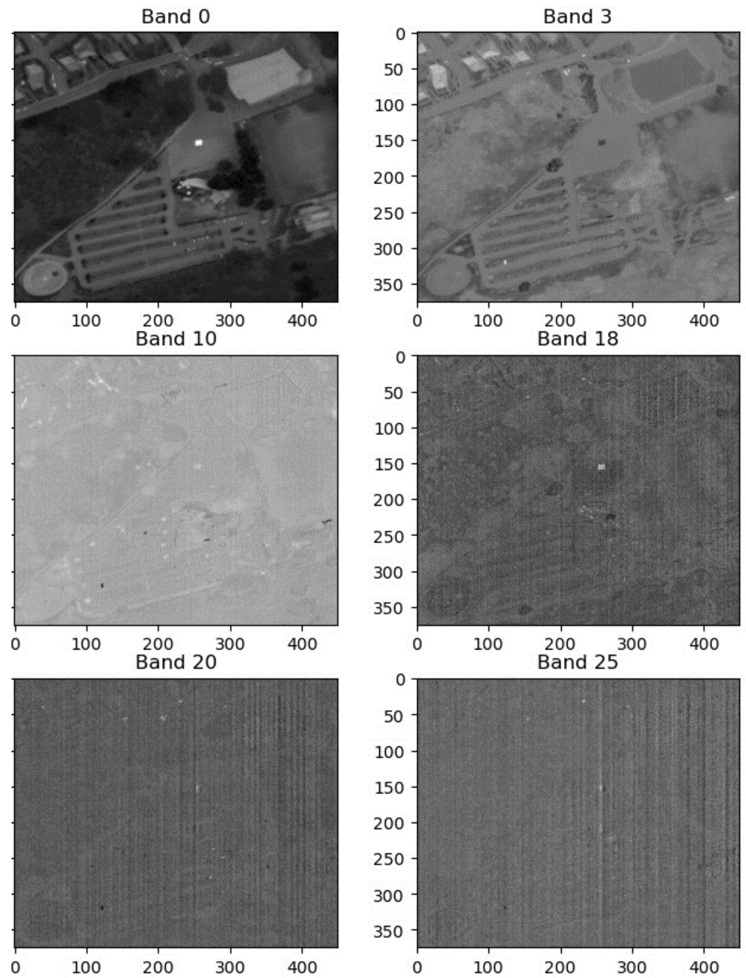
Is there a difference between project the data on the eigenvectors of the data's matrix covariance- **data algorithm**- or on the eigenvectors of the M_{white}'s matrix covariance- **M algorithm**.

Roughly speaking, we didn't find significant differences between the two methods, where the main criterion in the ROC curve of the MF, as will be discussed further.

In both cases we saw that the cube can be compressed into approximately 20 clear bands, but we did found some differences.

3. Compression results:

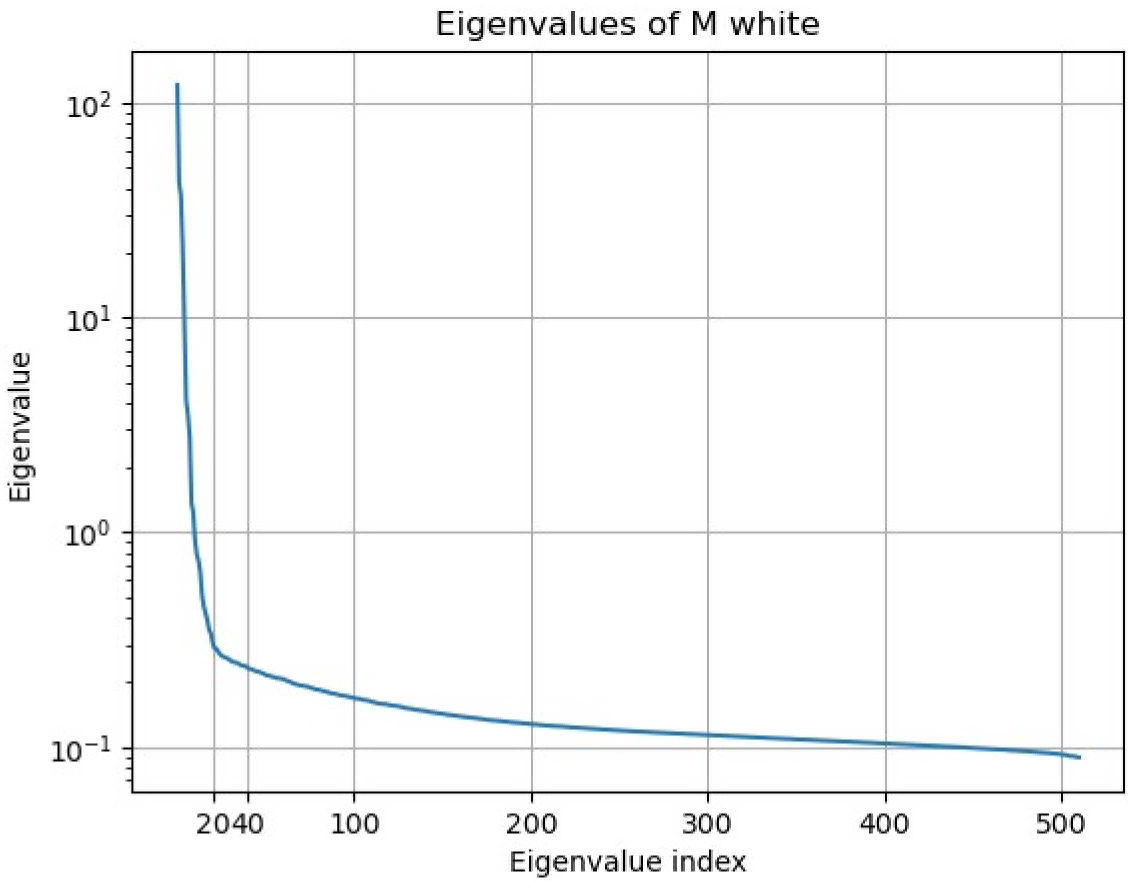
And now, let's look at a few simulations. We visualized our new hyperspectral picture: for the **M algorithm:**



We can see that the information was compressed into approximately 20 bands, and we would like to show few graphs which prove it mathematically.

Yaniv Mazor 315635722
Ayalla Reuven 314077033

First of all, we can take a look in the eigenvalues chart:



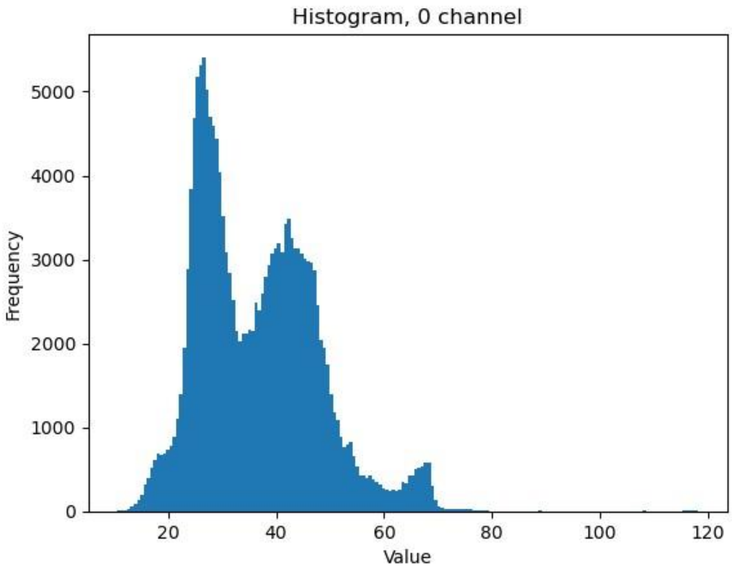
We sorted the eigenvalues in decreasing order, and we saw that most of the information found in the first 20 bands.

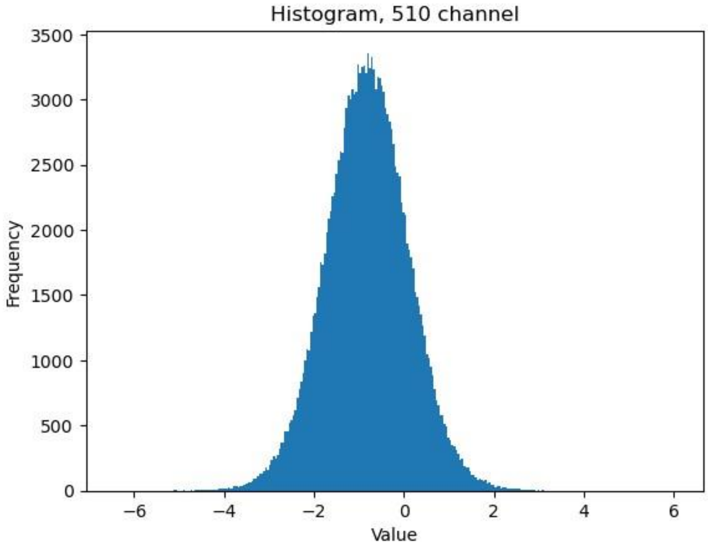
$$\frac{\sum_0^{20} \quad \lambda_i}{\sum_i \quad \lambda_i} = 0.8$$

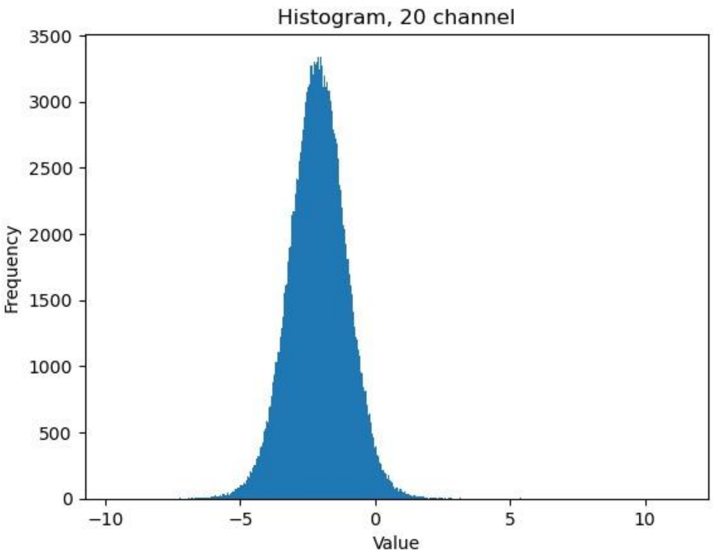
In addition, the first channel standard deviation is 11.2, where the last channel std is 0.9. Interesting to see that channel 18 std is already 1.1.

Yaniv Mazor 315635722
Ayalla Reuven 314077033

We also checked the histograms of the channels:



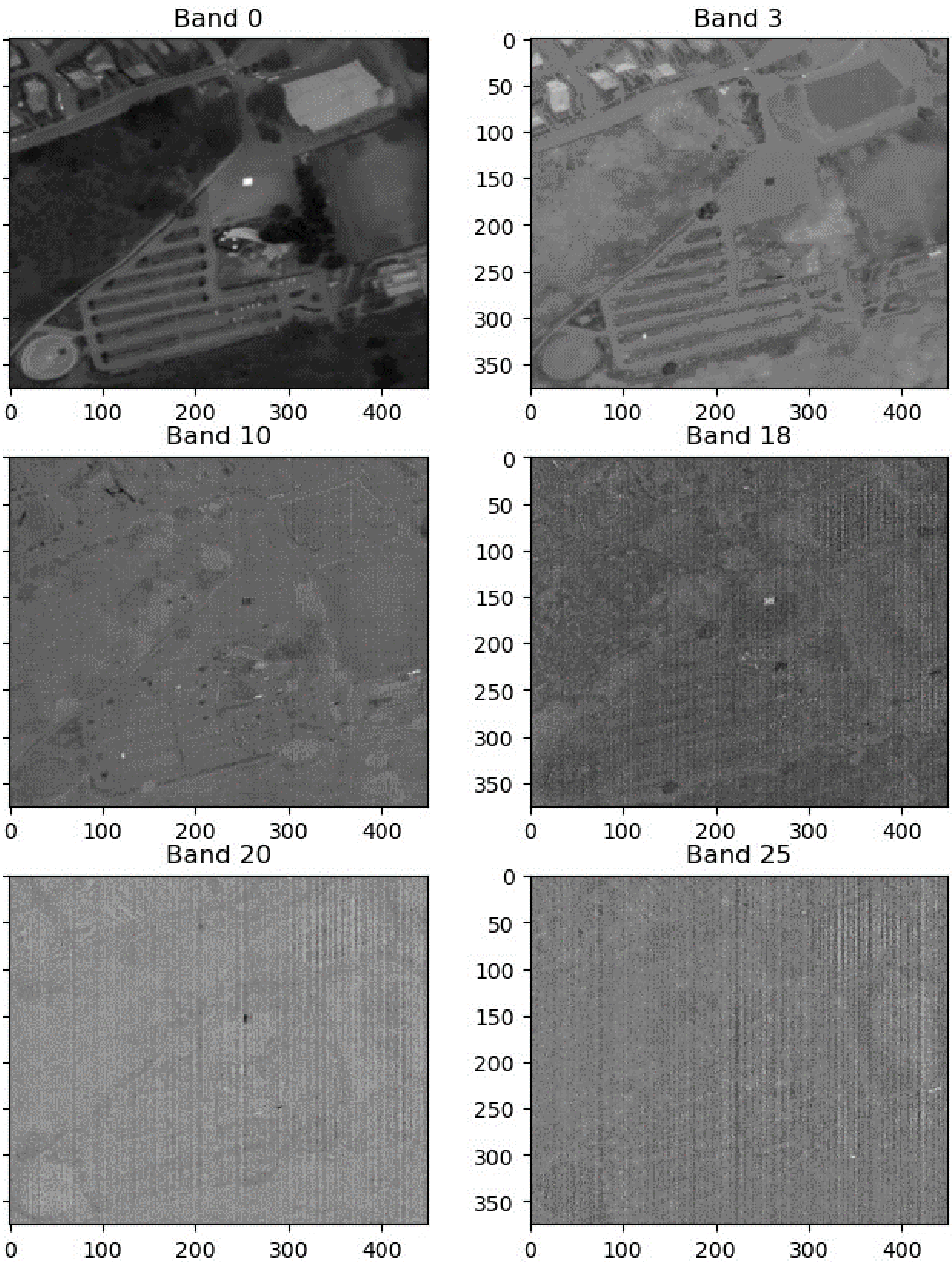




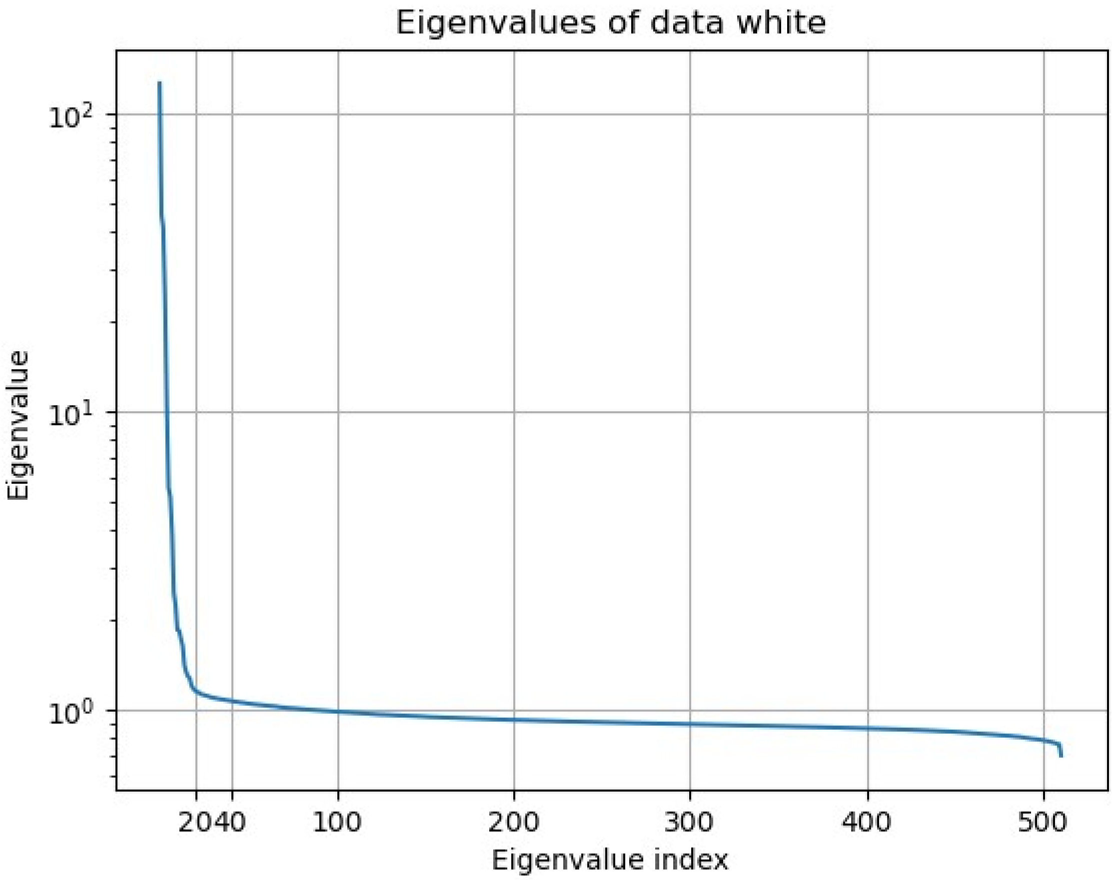
And again, we have indication to the fact that most of the relevant information compressed into the first 20 channels.

Yaniv Mazor 315635722
Ayalla Reuven 314077033

Data algorithm:



Yaniv Mazor 315635722
Ayalla Reuven 314077033

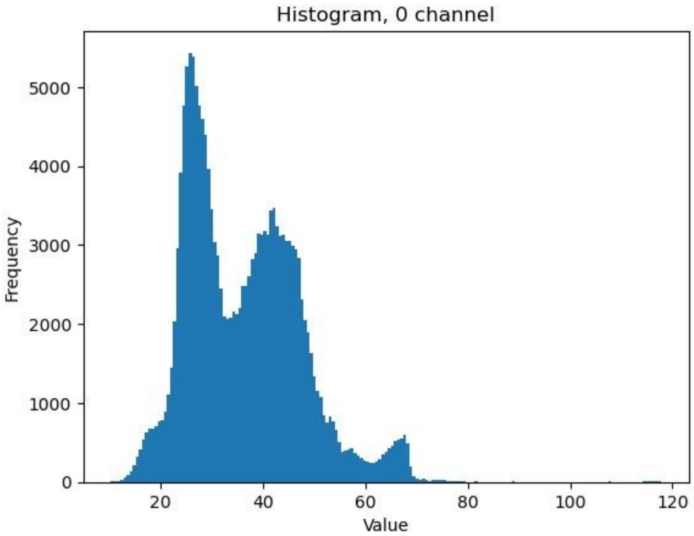


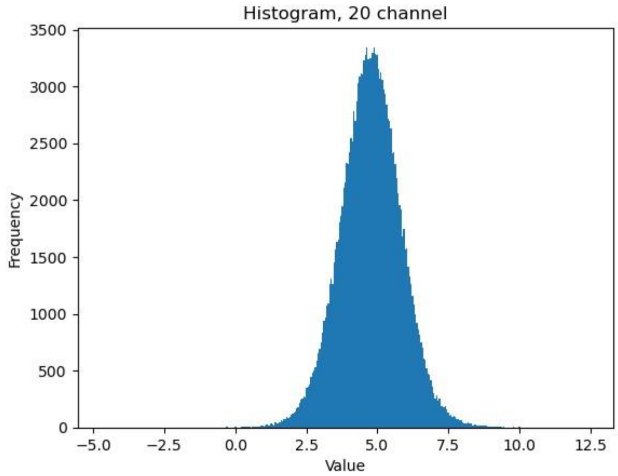
$$\frac{\sum_0^{20} \quad \lambda_i}{\sum_i \quad \lambda_i} = 0.4$$

From the channel’s std perspective, we saw that the first channel standard deviation is 11.2, where the last channel std is 0.83. As in the previous case, channel 18 std is already 1.1.

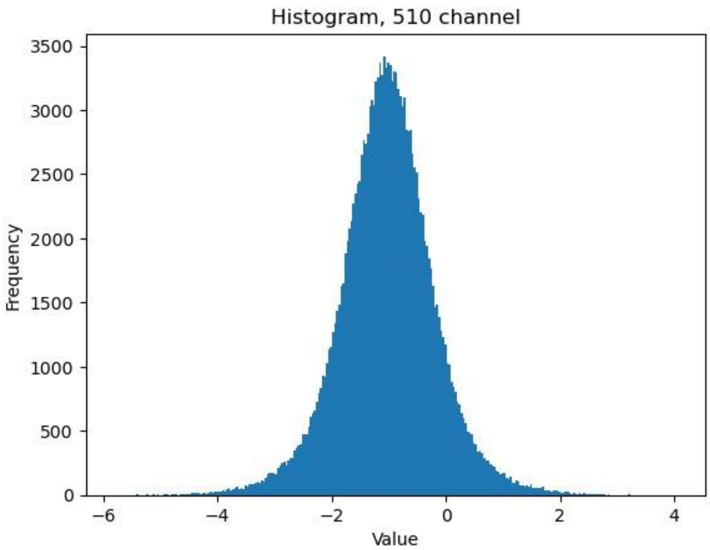
Yaniv Mazor 315635722
Ayalla Reuven 314077033

Let's look at the histogram as well:





To sum up, we see that there is some difference between the two methods. For the M algorithm we got a little better result, but the improvement wasn't significant (especially when considering the target detection's filters results). And yet, note that for $\frac{\Sigma_0^{20} \quad \lambda_i}{\Sigma_i \quad \lambda_i}$ we got two times higher results for the M-algorithm. That may indicate that the compressed of the M process was more efficient. For those reasons, we will use from now on in the M algorithm. Anyway, the findings suggest that utilizing approximately 20 PCA bands for the MF can yield satisfactory performance.



<u>4. MF procedure and performances:</u>

So, after we compressed our data, and moved into the PCA space, we can finally talk about the MF filter.

As we said previously, at the end of the day our main goal is to detect targets in hyperspectral images.

One of the popular filters, which we discussed widely in our course, is the Match Filter (MF). There are few versions for this filter, such as the SD filter[5], which was made for dealing with data edges, or segmented MF[1], which is trying to deal with non-stationary noise.

Before trying to expand our filter for such cases, we will first discuss the basic operation of the Match Filter. For each pixel we apply the formula:

$$MF(\vec{x_\iota}) = \vec{t}^T \Phi_n^{-1} \left(\overrightarrow{x_\iota - m_{8_\iota}}\right)$$

Where we assume that the target signature, t spectrum, is known and in addition to additive noise, we have target in the form of:

$$x_{wt} = x_{nt} + pt$$

For the case where t is unknown, the RX algorithm is used. As we have seen in the class, this algorithm check the anomaly of each pixel, and based on the fact that large anomaly for Gaussian distribution may point on the presence of target[5]. We find this algorithm demand high p for be effective, so we didn't investigate him deeply in our simulations. Another algorithm which is based on anomaly is the SSRX algorithm- which could me more relevant for as since its subspace projection variation of the RX[8].

After applying the MF, we determine threshold (decision rule) and tag the pixel as pixel with or without target.

For examine our filter performance, and in order to choose the threshold well, we compare between cube with and without target.

This procedure of making histogram and creating ROC curve was detailed in the "Debbie Cube" mini-project.

We can see that the mathematical computation of the filter is proportional to the number of channels, and that exactly where the PCA operation become relevant.

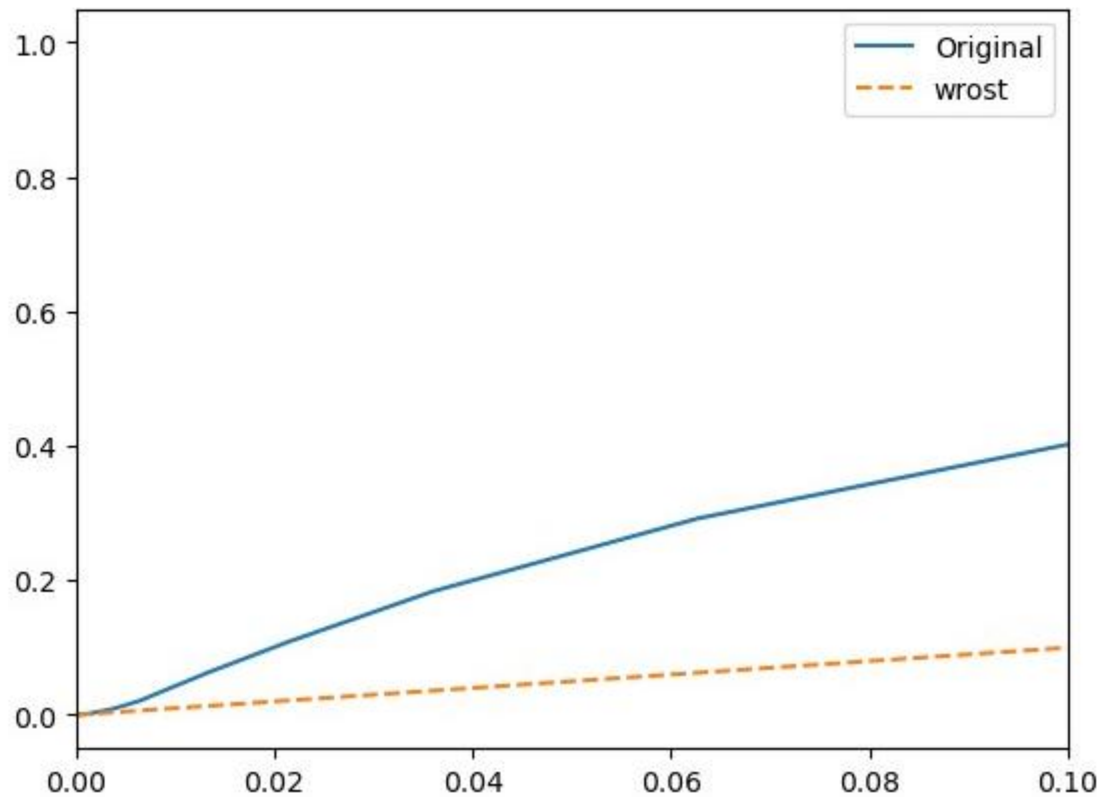
We would like to compare the performances of the MF for the original cube (511 channels) and the compressed cube (21 channels).

---

[5] Theoretical foundations of NRL spectral target detection algorithms, ALAN SCHAUM, Vol. 54, No. 31, 2015, Applied Optics

Later, we will try to improve the filter, according to the cube properties, and we will also investigate other learned filters (innovation part).

MF for the original cube:



Original performance, using 'Area Test':

Yaniv Mazor 315635722
Ayalla Reuven 314077033

th= 0.001: 0.00015855060589282686

th= 0.01: 0.015341601284888082

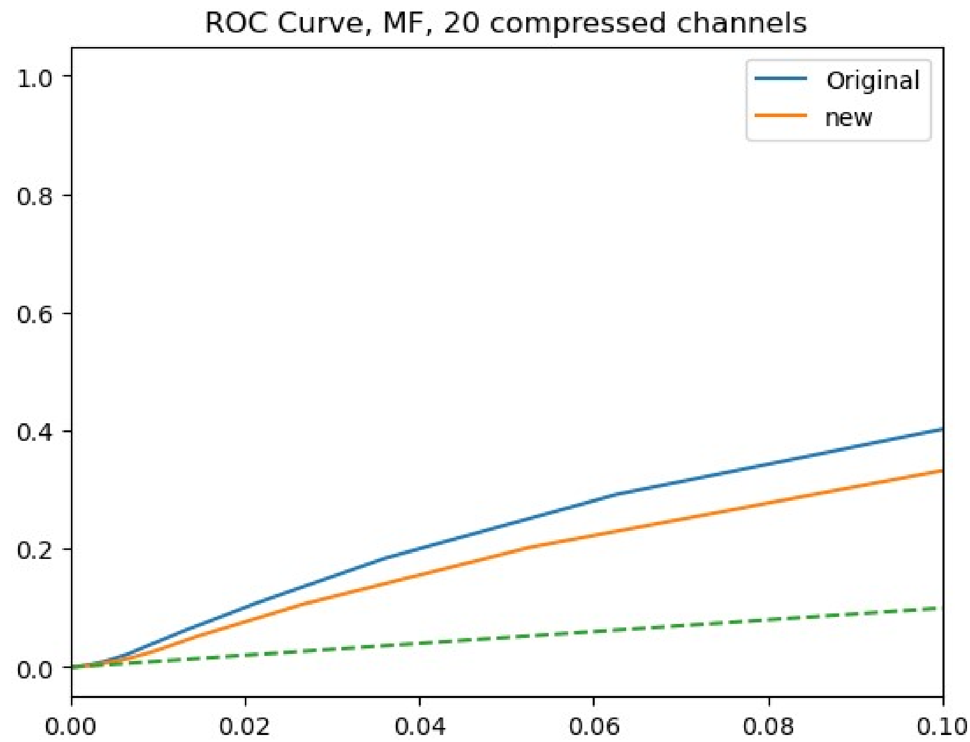
th= 0.1: 0.11765503889105476

That our starting point, we would like to get the same results for the compressed cube.

The comparison between the original cube's ROC curve and the new compressed cube's ROC curve will be used from now on to examine the influence of the PCA compression on target detection.

We will see that different algorithms will give us different results.

MF on PCA bands:



	Original	Compressed
$A_{0.001}$	0.00015855060589282686	9.657393854674394e-05
$A_{0.01}$	0.015341601284888082	0.011956546518601285

$A_{0.1}$	0.11765503889105476	0.059586134719514836

It is observed that when compressing the channels to 20, the resulting curve is similar to the original curve. However, the expectation was to achieve better performance in terms of the "Area Test"[5]. We saw that for 100 channels, the results were already quite similar to the original one.

One possible explanation for this could be the limited computational resources available.

Nevertheless, the option of compressing 511 channels into 20 (or 100) channels can still be useful in various fields, despite the slightly lower performance.

5. Innovation: Data stationarity and other algorithm performances:

We examined the influence of PCA compression on the MF algorithm and explored various methods to enhance its performance. In the subsequent phase, we explored alternative algorithms and leveraged the properties of the data to achieve improved outcomes. Among the algorithms we investigated, we found the PL_0MF algorithm particularly intriguing. ACE was already discussed in class, but we deemed it valuable to explore them further to gain a comprehensive understanding of the subject matter.

There are others filters found in the literature, but the minor performance improvements achieved by more advanced detectors are insignificant in practical applications due to the limitations and uncertainties surrounding various aspects of the deployment filters[6].

PL_0MF On PCA Bands[1]:

One of the main point we try to investigate was the stationary of the picture and the options of using some local patterns in order to achieve better performance. In [1] we saw suggestion for substitute the global eigenvalues of the MF algorithm, in the local variance of each pixel.

[6] PROCEEDINGS OF SPIE Manolakis, D., Lockwood, R., Cooley, T., Jacobson, J. : SPIE Defense, Security, and Sensing, 2009

Yaniv Mazor 315635722
Ayalla Reuven 314077033

We know that the SNR is proportional to the eigenvalues, and the switch of global eigenvalues in local variance seems to us like operation that can improve our filter, especially in non-stationary data (like in segmentation).

According to [1], the new filter will be as follows:

$$MF_{old} = \vec{t}^T \Phi_n^{-1}\left(\overrightarrow{x_\iota - m_{8_\iota}}\right) = \vec{t}^T E \Lambda^{-1} E^T \left(\overrightarrow{x_\iota - m_{8_\iota}}\right) \downarrow$$

$$MF_{new} = \vec{t}^T \Phi_{n\_local}^{-1}\left(\overrightarrow{x_\iota - m_{8_\iota}}\right) = \vec{t}^T E \Lambda_{loc}^{-1} E^T \left(\overrightarrow{x_\iota - m_{8_\iota}}\right)$$

Λ_{loc} is diagonal matrix, where each element represent the local variance of the PCA PIXEL'S neighbors. Note that if we will take too many bands into account, the SD will be very low (because of the compression).

This new covariance matrix is called Quasilocal Covariance Matrix [7].

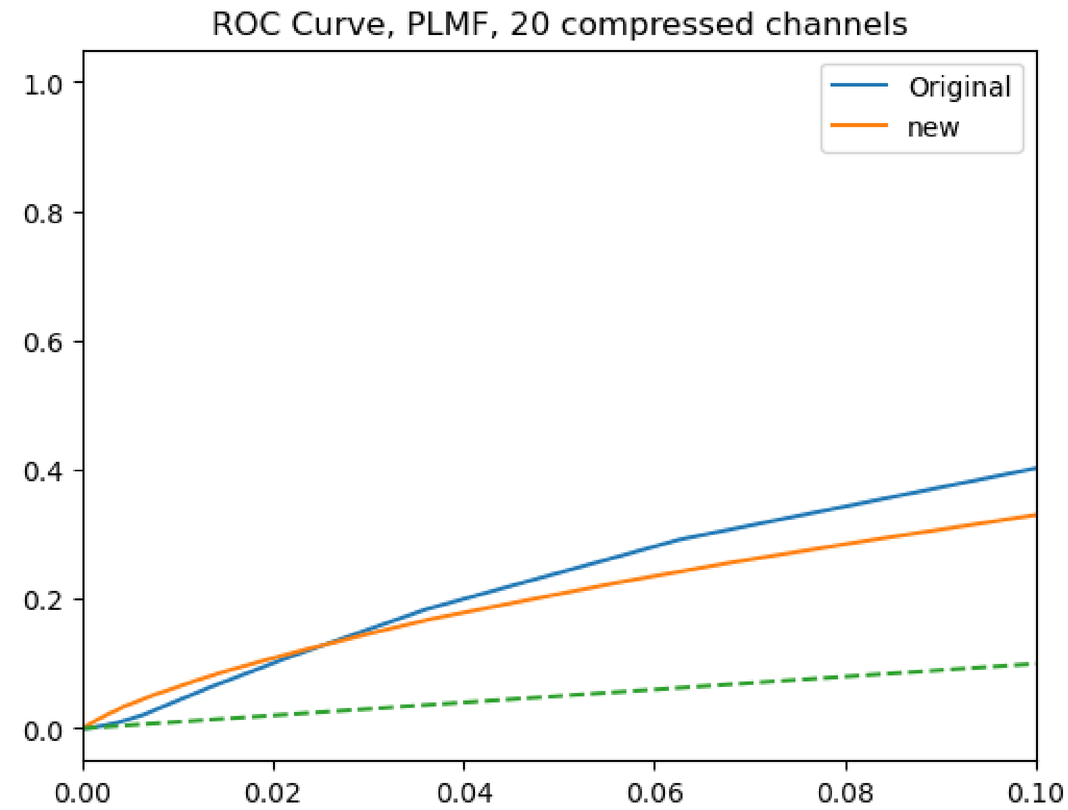
As have been offered in the mentioned article, for making the algorithm even better, and overcome the problem of low variance in high PCA bands, we will prefer to take the maximal value between the local variance and the global eigenvector, so our final Λ matrix will be diagonal matrix, with the element:

$$\Lambda_{ii} = max(\lambda_{i_g}, V_i)$$

Where λ_{i_g} is the global eigenvalue and V_i is the variance of our pixel in the i_{th} band.

And the results:

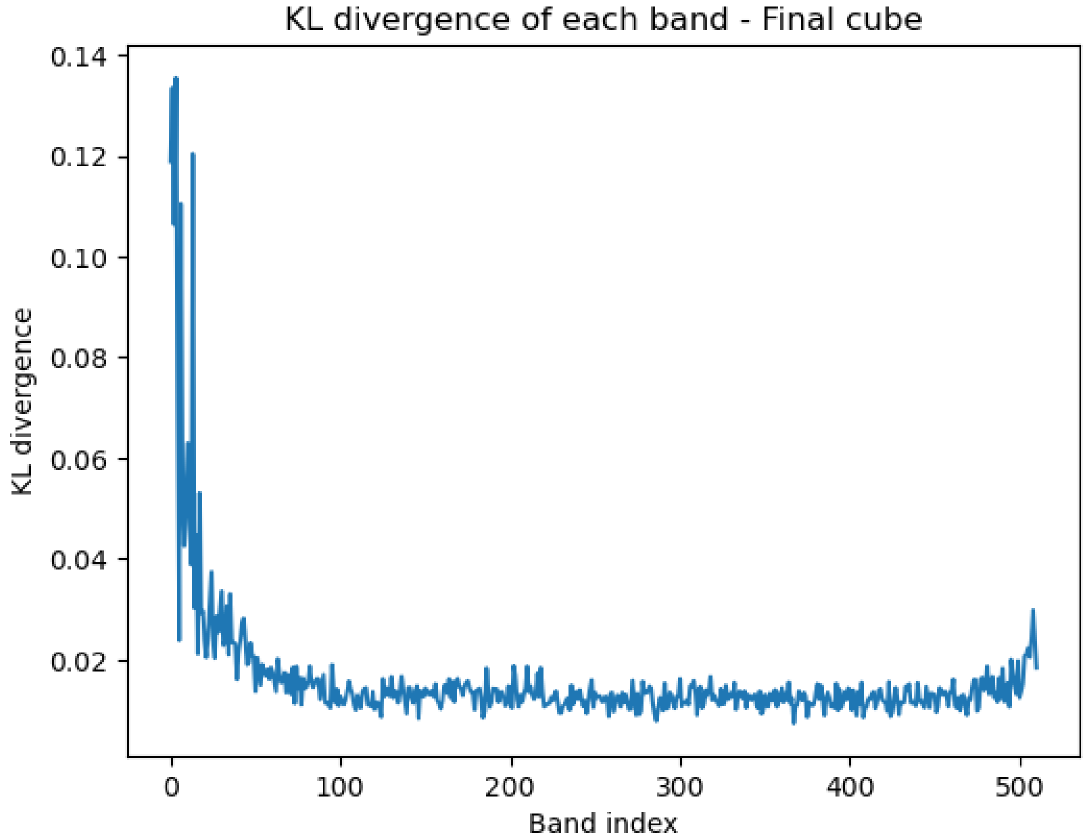
Yaniv Mazor 315635722
Ayalla Reuven 314077033



	Original	Compressed
$A_{0.001}$	0.00015855060589282686	0.003063583918159331
$A_{0.01}$	0.015341601284888082	0.035292378269950526
$A_{0.1}$	0.11765503889105476	0.0970508527846363

For low P_{FA} we saw some improvement, but for higher P_{FA} the original cube results was already better. The fact that this algorithm didn't satisfied significant improvement lead us to check the stationarity of the whole PCA cube.

As we have seen in the class, one of the methods for estimate the stationarity of data is to examine its KL divergence, in compare to Gaussian distribution (where the variance and the expected value are estimated from the data).



As we have expected, for the higher bands the data is almost Gaussian. For the first bands the divergence was higher, but it's actually pretty small divergence (comparing to what we have seen in the class). This confirming our claim- the algorithm didn't improved the ROC curve significantly because of the new data stationary.

Note that unlike the segmentation method, we was looking for the stationarity of the data. The noise in the new cube is white, as has been mentioned.

<u>ACE on PCA bands:</u>

As we have seen in the class, the ACE algorithm, take into account only direction of our data. The angle between the target and the data is examined:

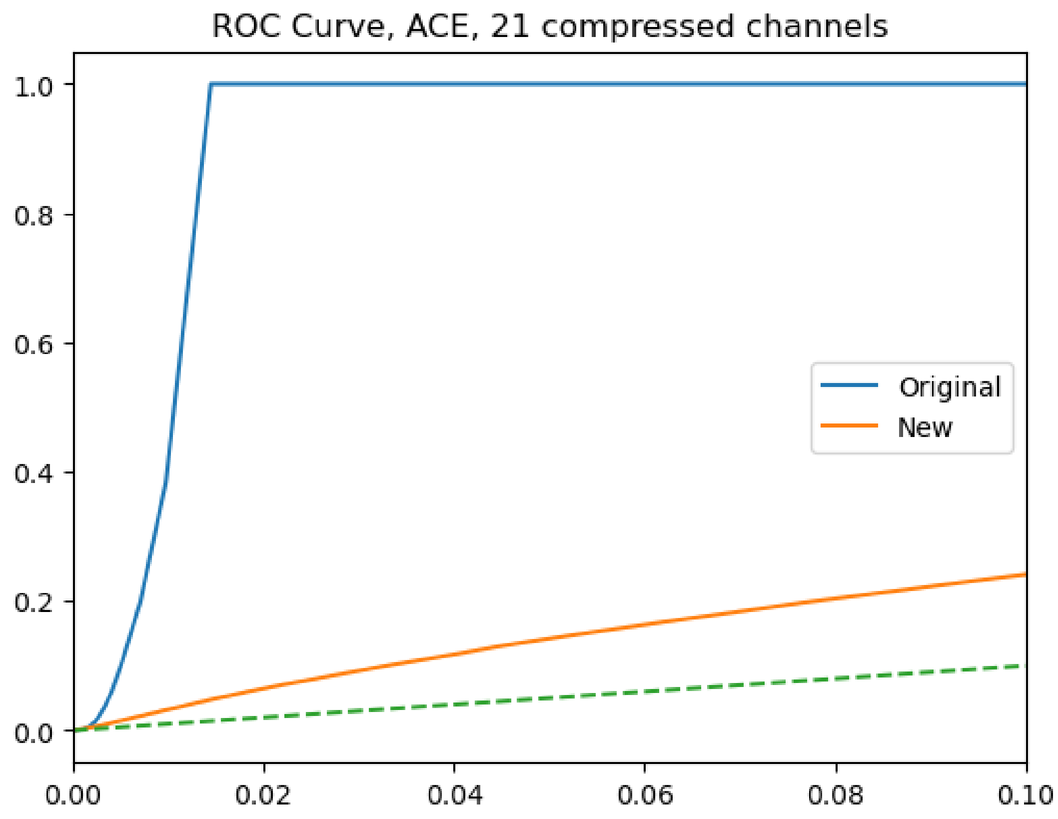
$$ACE(\overrightarrow{x_\iota}) = \frac{(\vec{t}^T \Phi_n^{-1}(\overrightarrow{x_\iota - m_{8_\iota}}))^2}{(\vec{t}^T \Phi_n^{-1}\vec{t})(\overrightarrow{x_\iota - m_{8_\iota}})^T \Phi_n^{-1}(\overrightarrow{x_\iota - m_{8_\iota}})} = cos\, cos\, \theta$$

Where θ is the angle between the target and the estimate noise.[2]

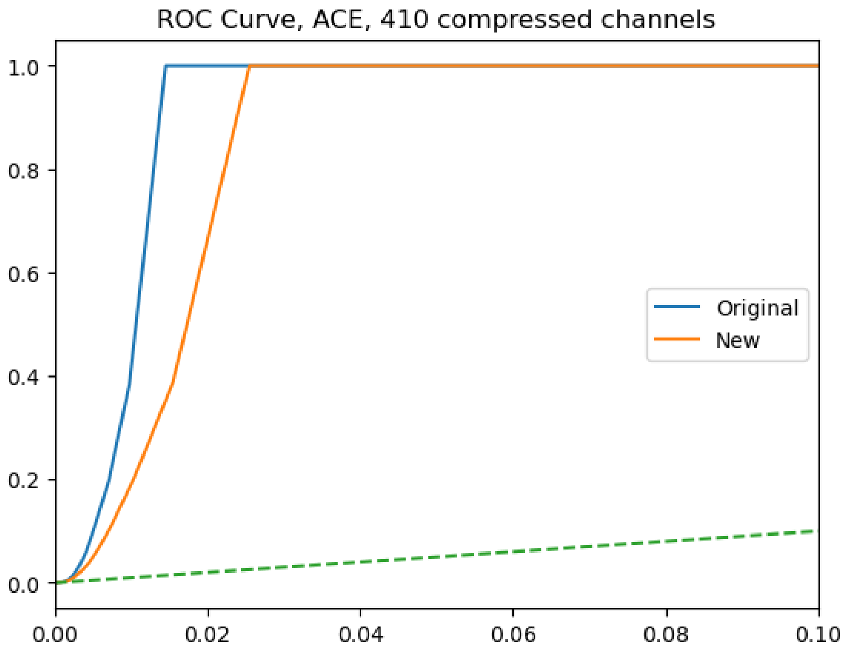
We saw that the compression doesn't work the same way on this algorithm, and we get worst results. This might be because of the fact that the ACE algorithm calculate direction (and not projection) of our data, and this operation require more information.

For 20 channels, the results was pretty bad, the better results was achieved only for 400 bands approximately.

The PCA was significantly better for other algorithms:



	Original	Compressed
$A_{0.001}$	0.00035839373733503004	0.0005021803082611645
$A_{0.01}$	0.20949788746200163	0.01063509305098879
$A_{0.1}$	0.8633573518388558	0.09220126736264536



	Original	Compressed
$A_{0.001}$	0.00035839373733503004	0.00021464230057414456
$A_{0.01}$	0.20949788746200163	0.03949414099303129
$A_{0.1}$	0.8633573518388558	0.8243871218193631

6. Conclusions and summary:

During our project, we explored the PCA algorithm and its influence on various filters for target detection. We aimed to enhance the algorithm by examining suitable filters and analysing each step of the process, including the level of whiting. In essence, we observed that employing MF for PCA bands can decrease the dimension of our system without compromising performance.

We believe that future research could concentrate on alternative filters that enable us to utilizeed the PCA projection. Additionally, the widely used M_8

algorithm for distinguishing noise statistics from data could be substituted with other methods to achieve superior outcomes.