Target Detection In Hyperspectral Images

Author: Ramesh Bhatta, CIS, RIT

Work:

The objective of this work was to apply a few target detection algorithms in Hyperspectral Images and perform their performance analysis. We used four algorithms namely: 1) Spectral Angle Mapper (**SAM**), 2) Spectral Matched Filter (**SMF**), 3) Adaptive Cosine/Coherence Estimator (**ACE**) and 4) Constraint Energy minimization (**CEM**) algorithms. The dataset used was "**SHARE 2010 HSI data**" provided by **Dr. Emmet Ientilucci** and the results of the tasks are explained in sections below:

1. **Original Dataset**

Fig 1: The Scene used for target detection

along with the truth masks.

2. Class Map

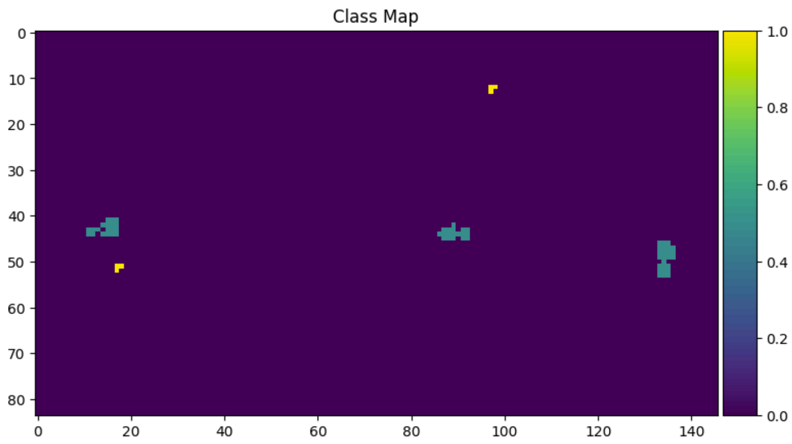

Fig 2: Truth map generated using ENVI (teal color is red panel target in open and yellow color is red panel in shadow)

3. Reference Panel Spectrum

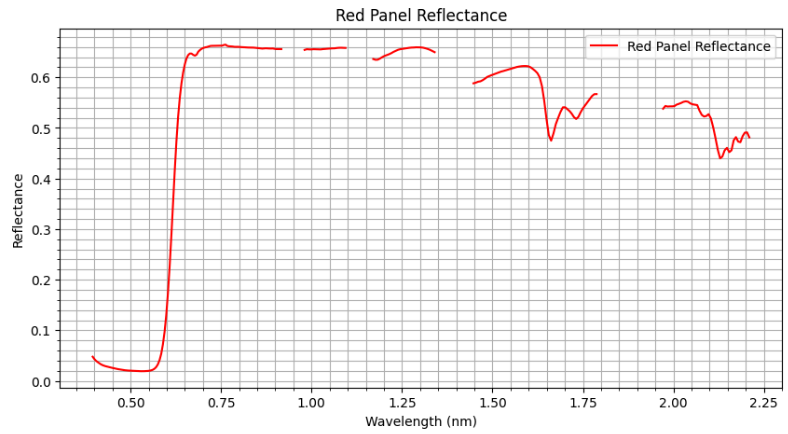

Fig 3: Red panel reference spectrum after bad band removal

TARGET DETECTION IN ORIGINAL SPACE:

We used four methods of target detection to detect the red panel target in the scene represented by our HIS data. The class map shown in figure 2 was generated using ENVI software and two classes of red panels were selected as truth mask. One class was the red panel in the open area like panels in lawn area and the roof and other class was the panels in shadow. The spectrum of red panel used as target signature is shown in figure 3. 5 regions with bad bands were removed and the remaining bands were used for the target detection purpose.

The results for each algorithm and the targets detected based on histogram thresholds are shown in the sections below. Almost all the algorithms were able to find the target in open areas while unable to find the target in the shadow region. Results presented below shows these implications:

a. Results For Spectral Angle Mapper Detector

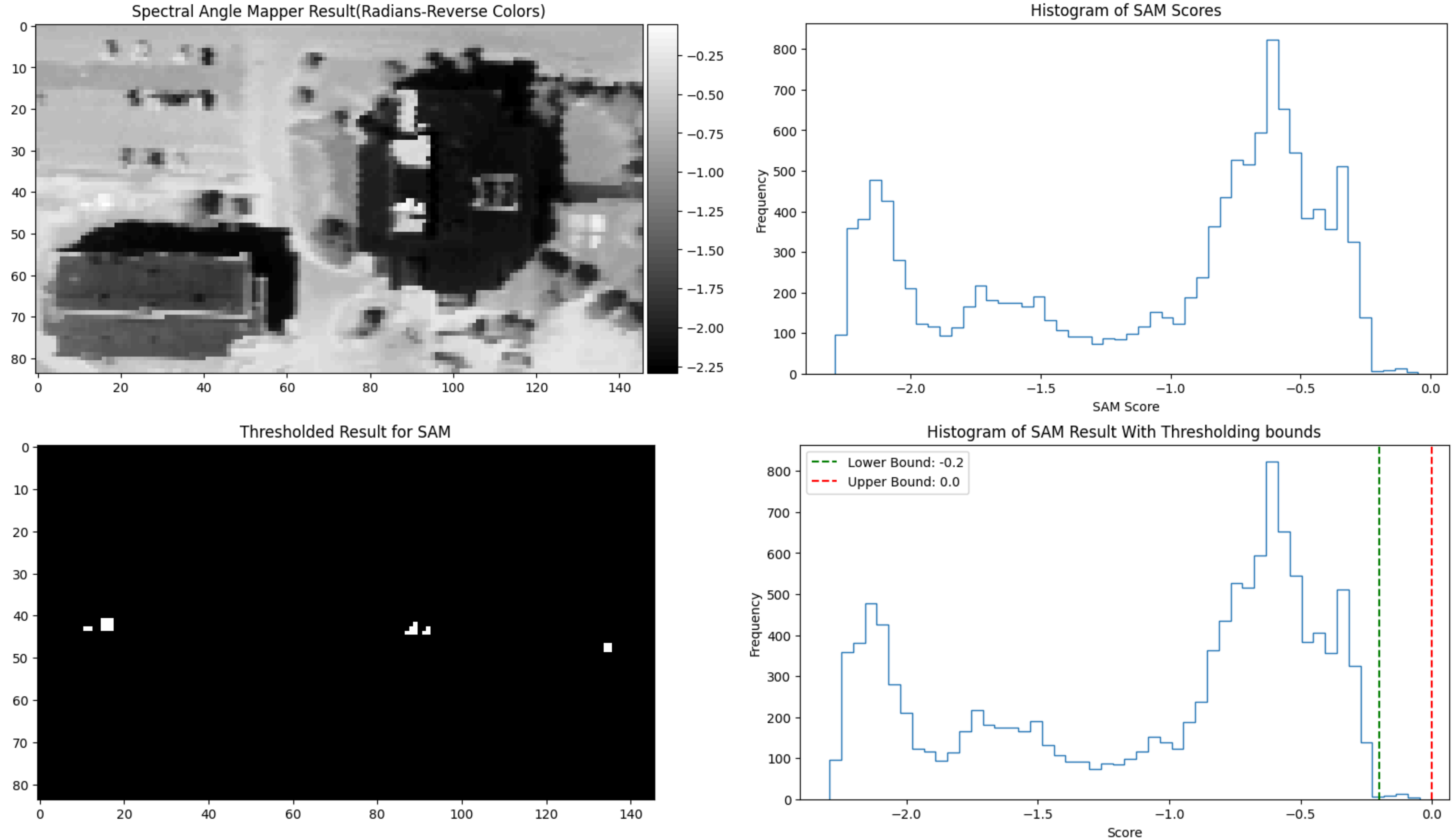

Fig 4: SAM Detector

b. Results For Spectral Matcher Filter:

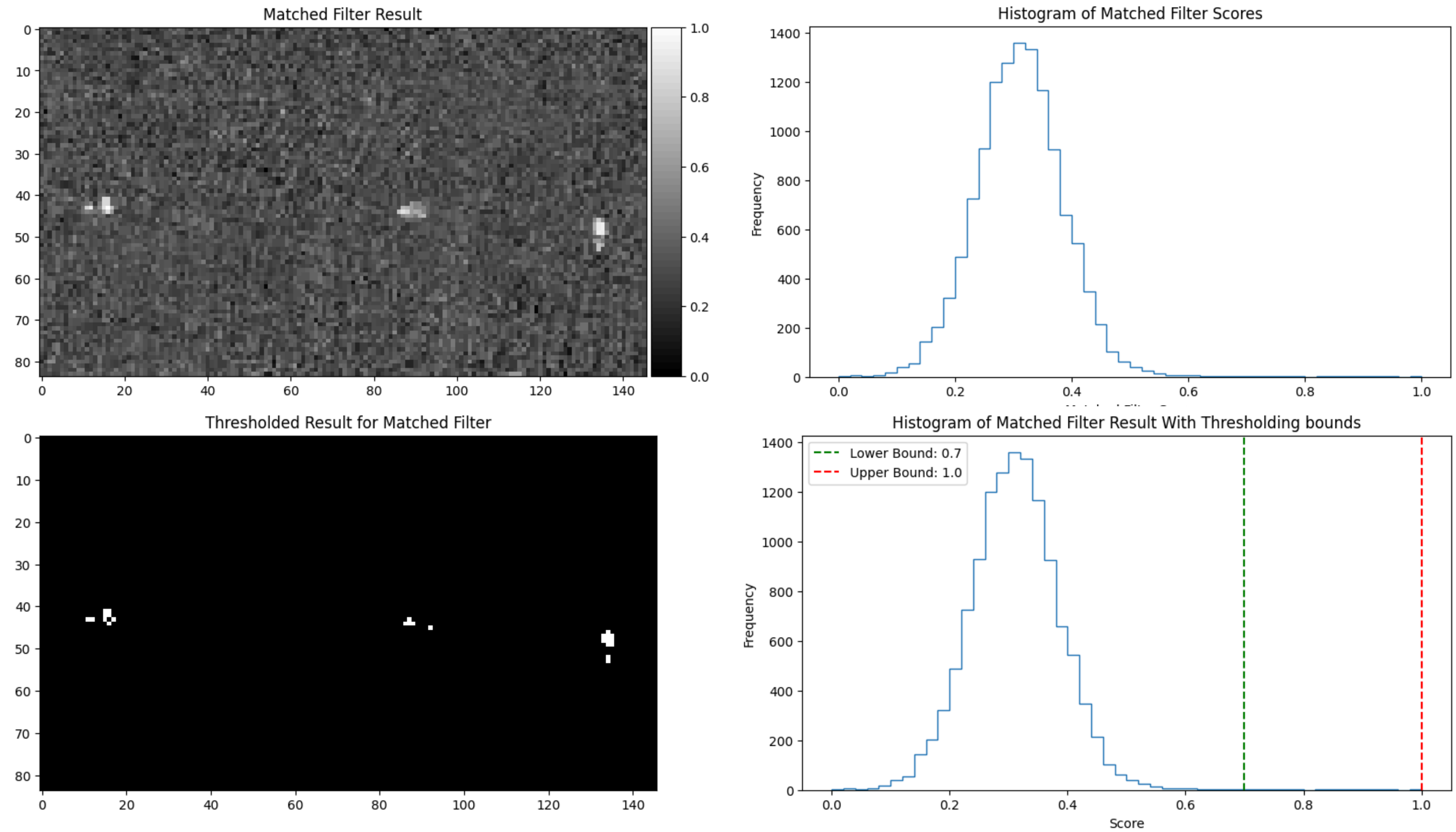

Fig 5: Spectral Match Filter

c. Results For Adaptive Coherence Filter:

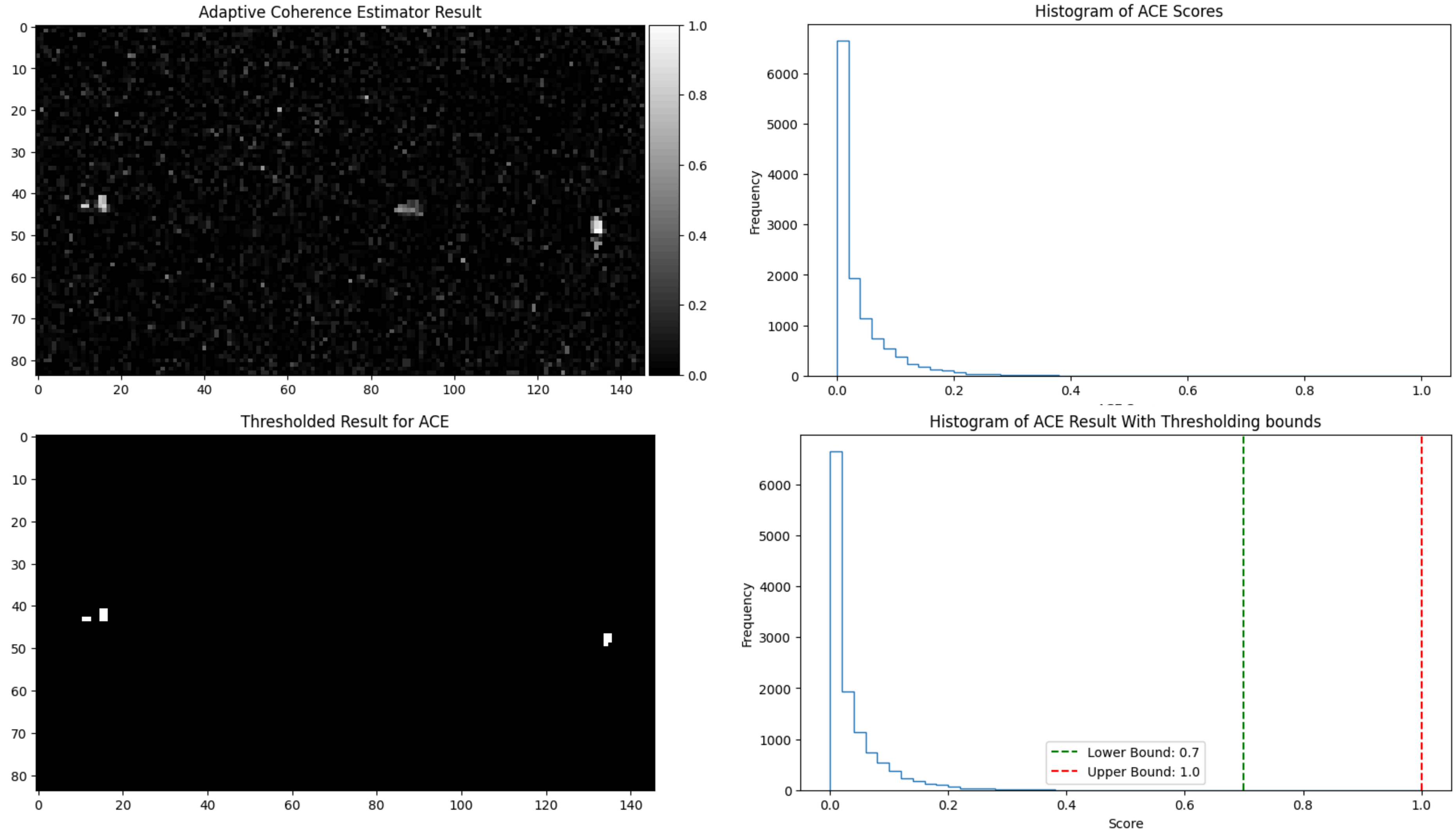

Fig 6: Adaptive Coherence Filter

d. Results For Constrained Energy Minimization detector:

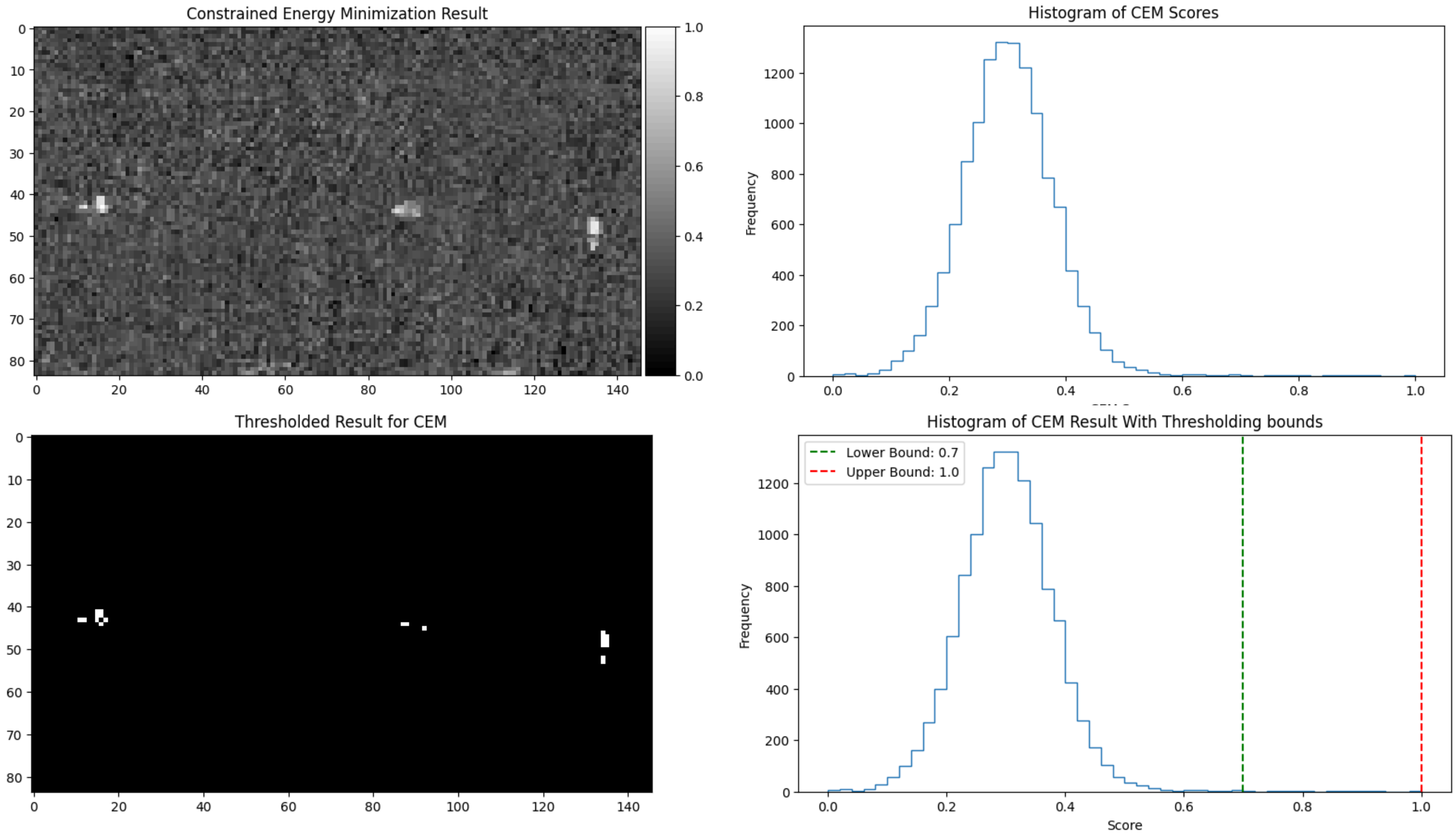

Fig 7: CEM Detector

e. Receiver Operating Characteristics Curve:

To access the performance of our detector the ROC curves were generated using the truth class map and our outputs from detectors. The following plot shows ROC curves both in linear and log scale:

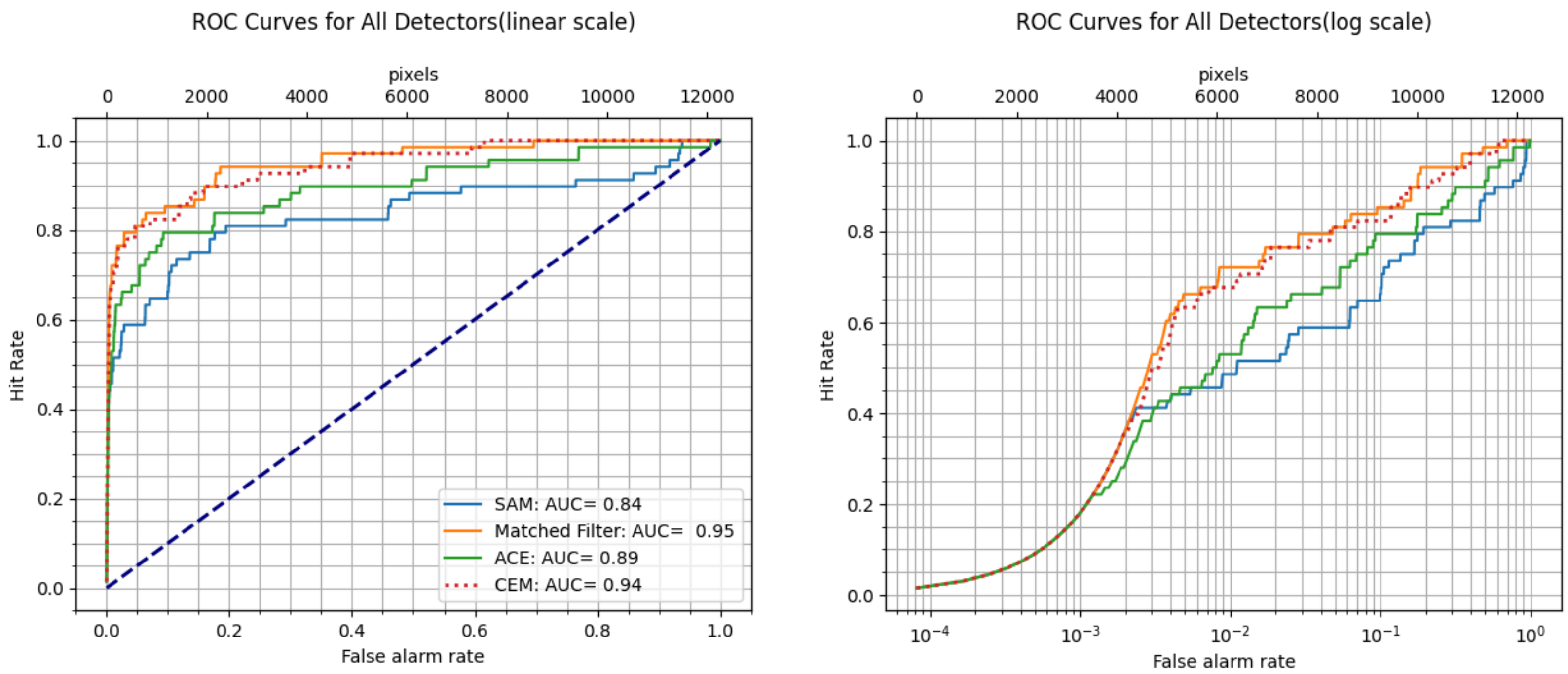

Fig 8: ROC curves for each detector both in linear and log scales

From the ROC curve we can say that the performance of "**Matched Filter**" and "**CEM**" algorithm is almost similar while the performance of "**SAM**" detector is below all of them with Area Under Curve (**AUC**) 84%.

TARGET DETECTION IN REDUCED DIMENSIONAL SPACE:

For dimensions reduction we used Principal Component Analysis (**PCA**) on the original dataset and used the total-variability plot which shows how much of the total-variation in original datasets is explained by some 'n' bands. The plot is shown below:

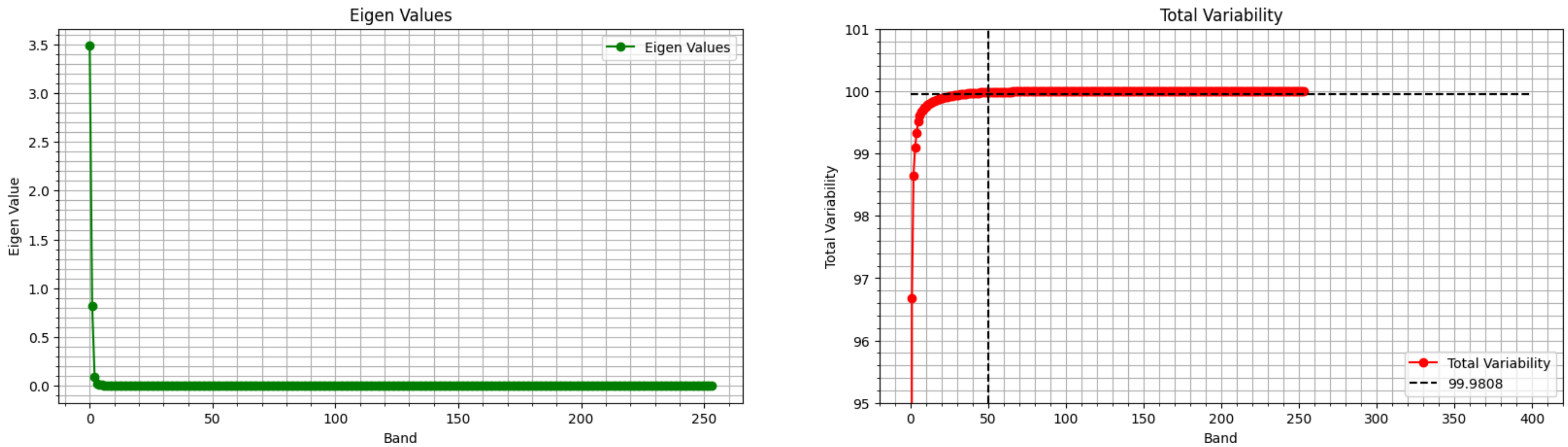

Fig 9: Eigen Value and Total variability plot

As we can see from the total variability plot, around 99.98% of variability in the original data is explained by just 50 bands in the PC space. So, we can reduce our dimensions to just 50 bands in this space and do the target detection here. We also projected our original spectrum into PC space using just the first 50 eigen vectors. So now we run the target detection using 50 bands in the PC space. The sample of 3 principal components in decorrelated space and the reference spectrum vector in the same space is shown in figure below:

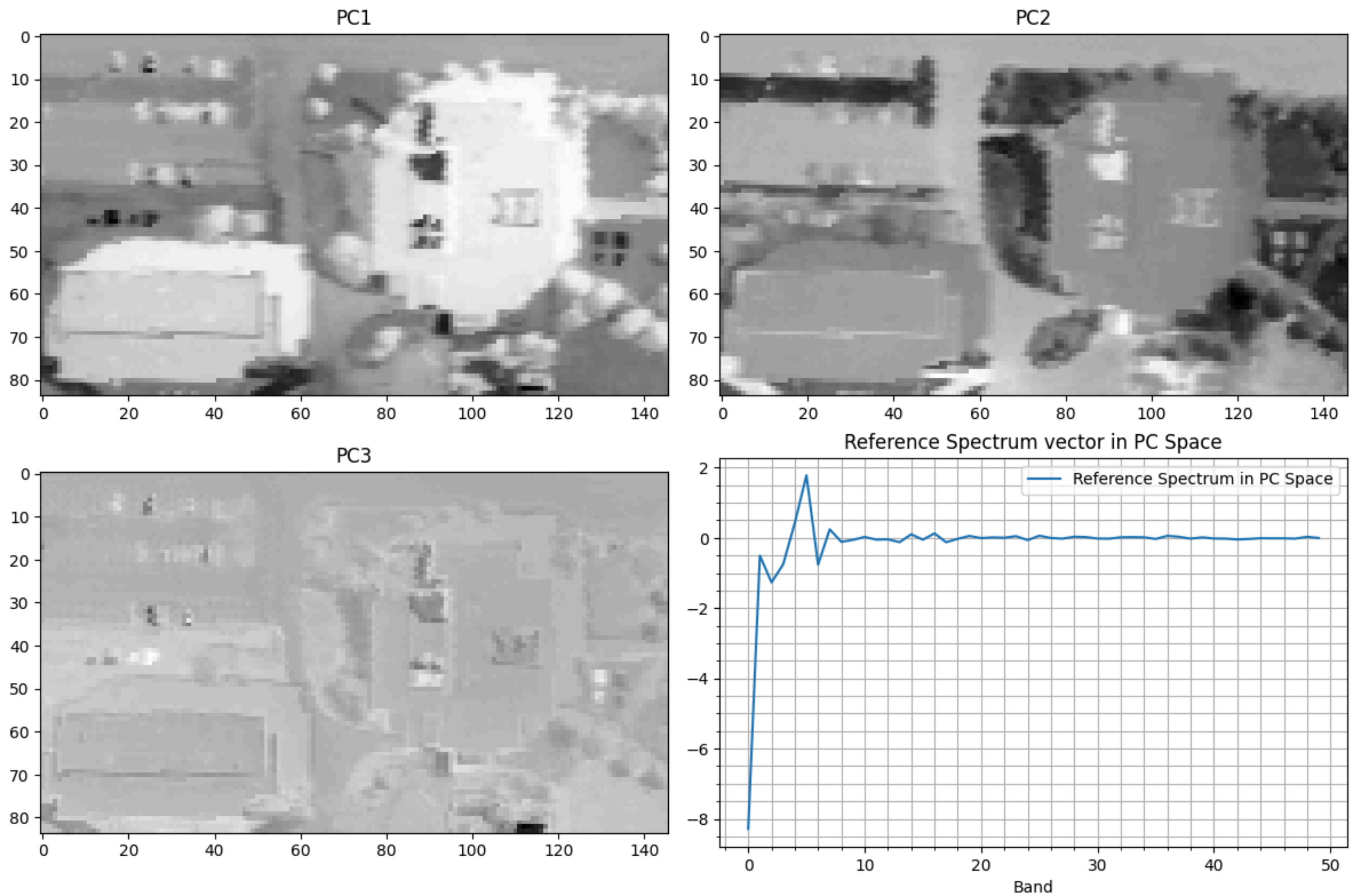

Fig 10: First 3 Principal components and the reference spectrum vector in PC space

Now we are set to apply target detection algorithms in PC space. The following sections show the results for each detection algorithms as before:

a. Results for Spectral Angle Mapper in PC space:

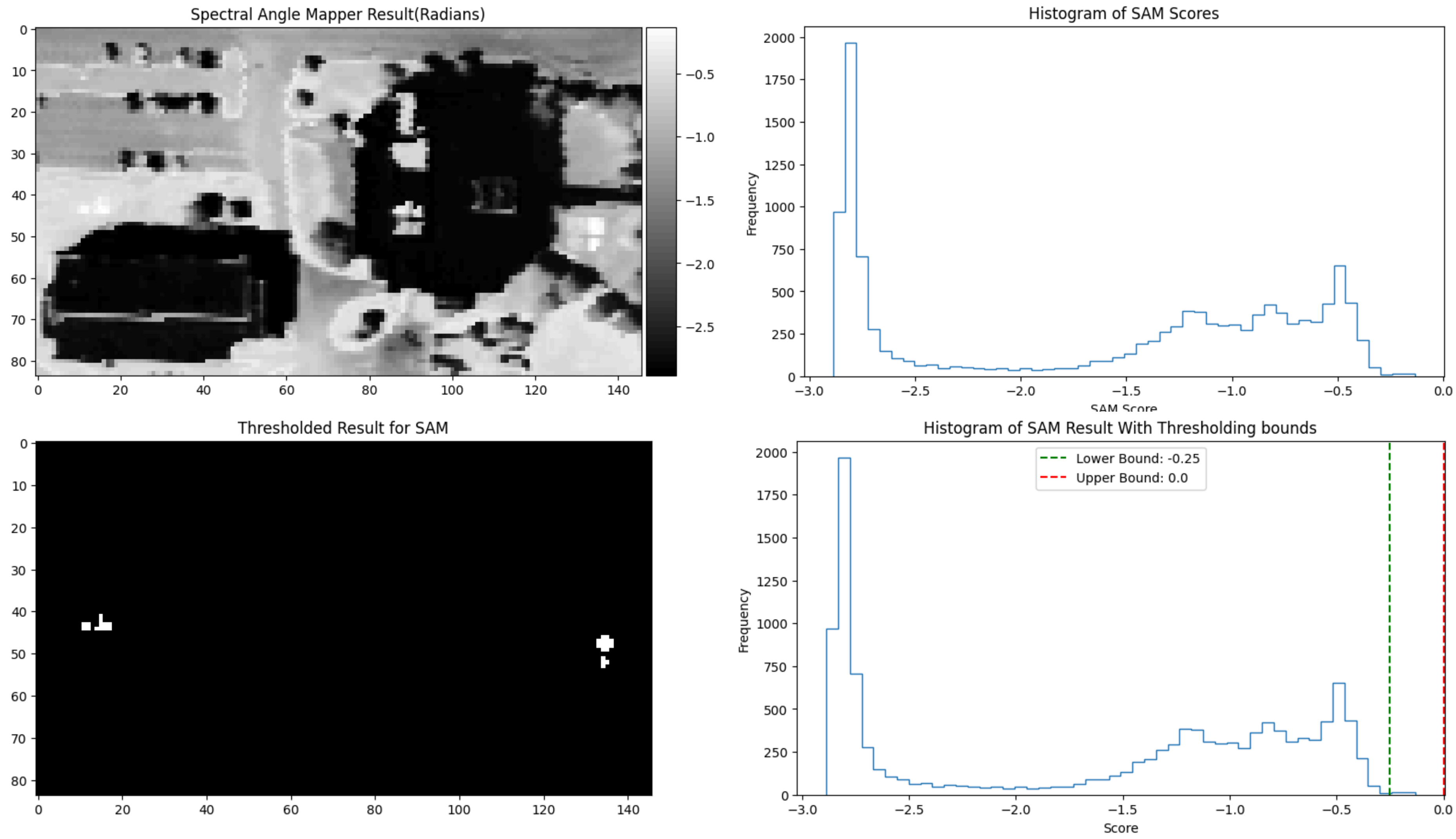

Fig 11: SAM in PC space

b. Results for Spectral Matched Filter in PC space:

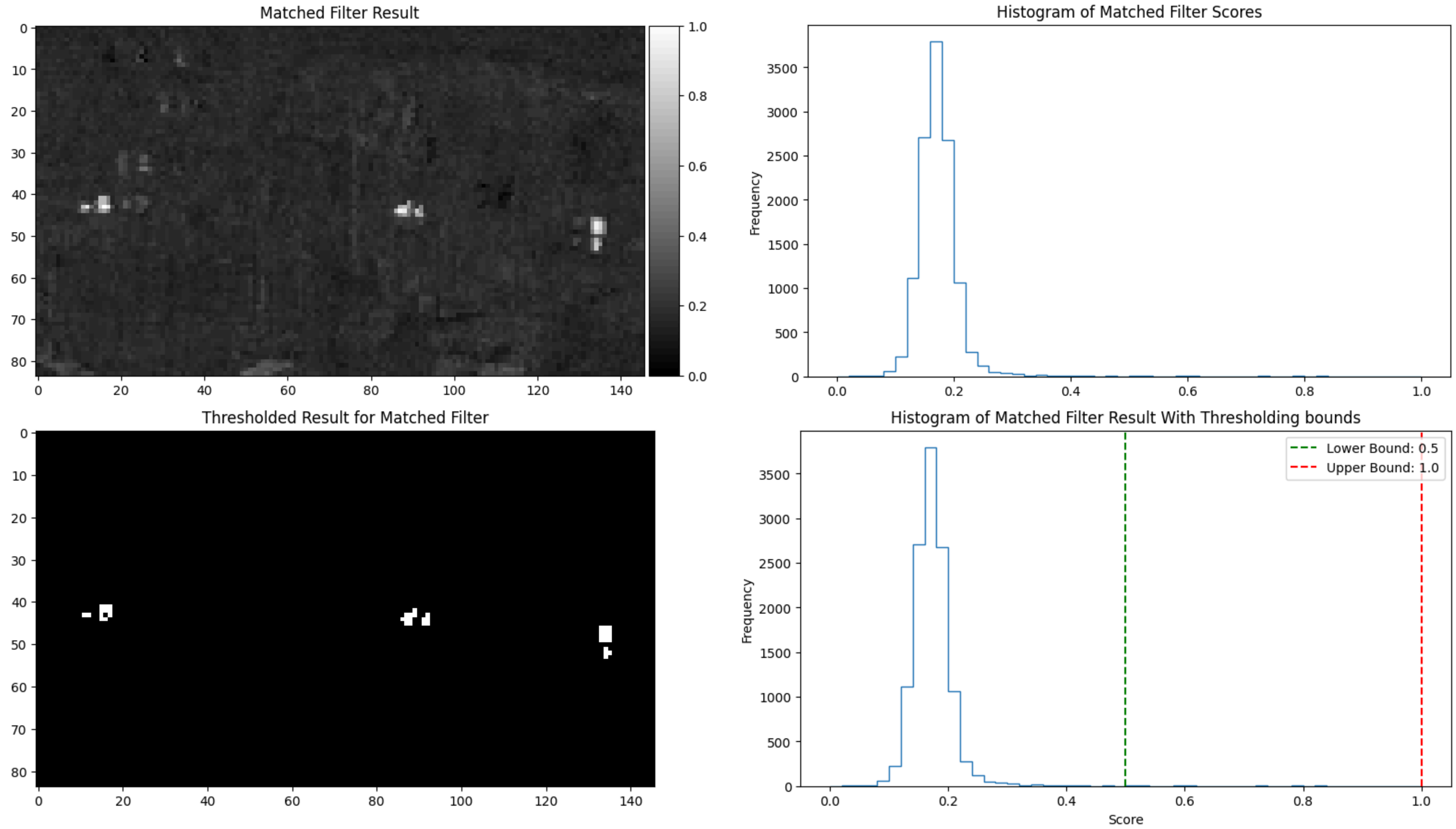

Fig 12: SMF in PC space

c. Results for Adaptive Coherence Estimator in PC space:

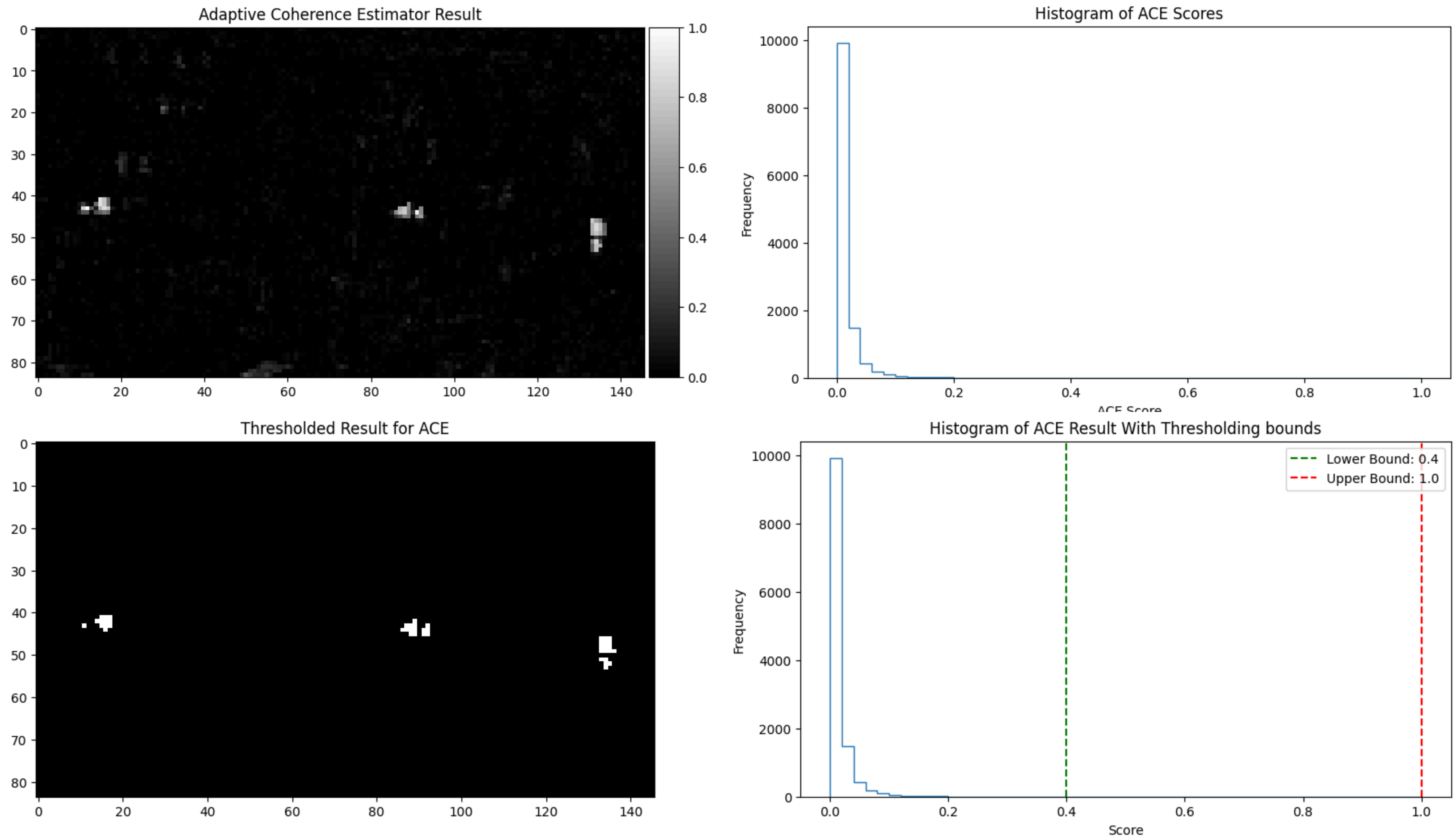

Fig 13: ACE in PC space

d. Results for Constrained Energy Minimization in PC space:

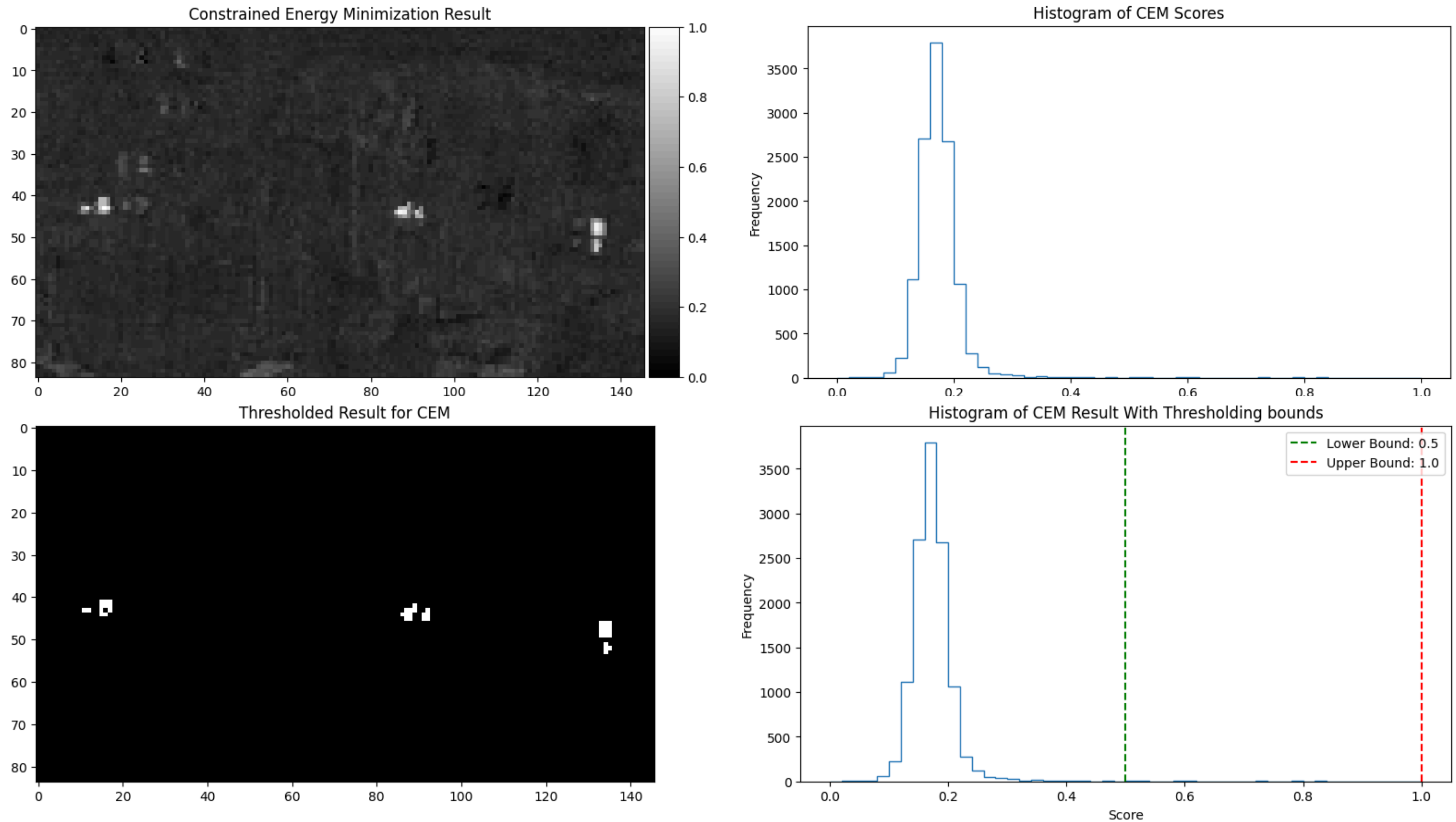

Fig 14: CEM in PC space

e. Receiver Operating Characteristics Curve:

Similar to earlier, we plotted the ROC curve for detection in PC space and obtained ROC as follows:

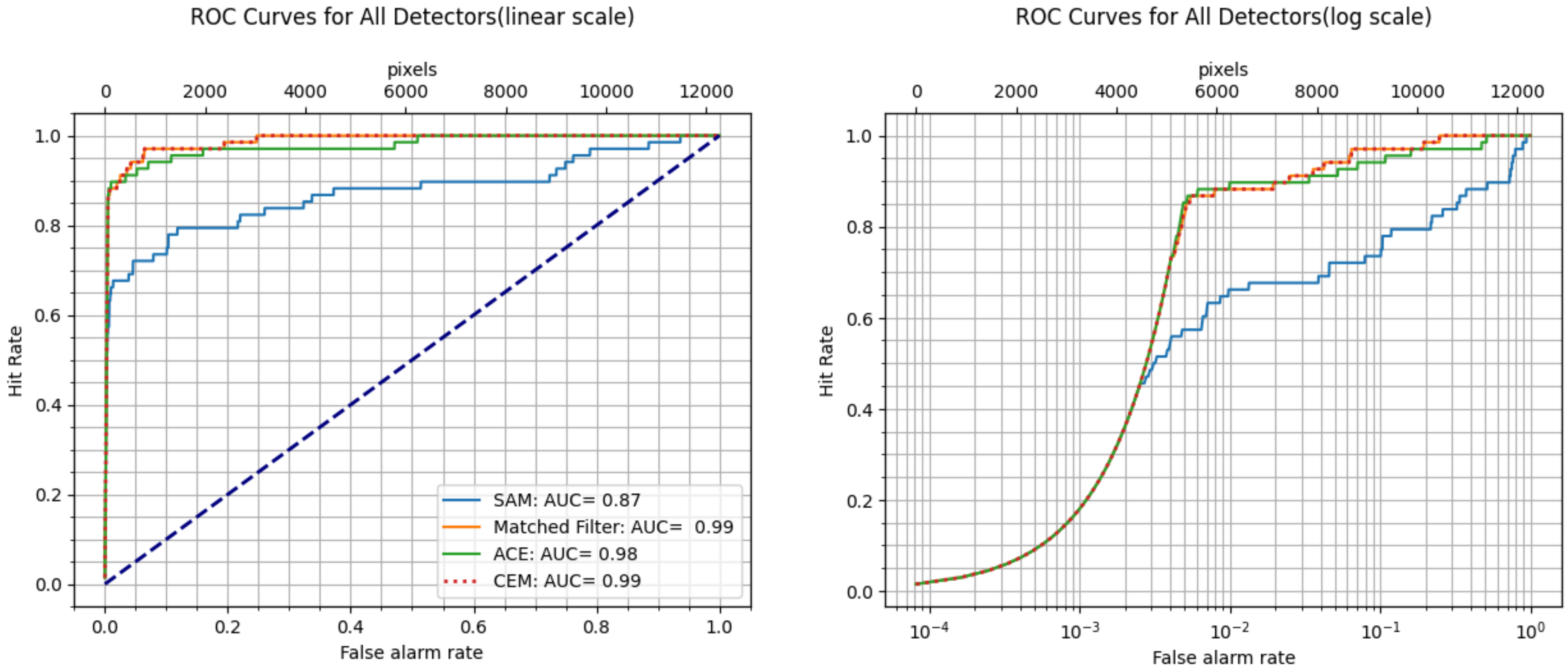

Fig 15: ROC for all detectors in both linear and log scale

(AUC is area under curve)

From the above ROC curve, we can see that detection in PC space is more efficient to find the targets than in original space. Comparing the ROC curves, we can see that the area under curve for "**Matched Filter**" and "**Constrained Energy Minimization**" is almost the same and higher than "**SAM**" and "**ACE**" algorithms in both spaces. SAM performs consistently below all other detectors in both spaces. Contrasting the detection ability between two spaces, the **AUC in reduced space is more** for all the detector than the original space which we can even see clearly in thresholded results for each detector. This may be because reducing dimension in PC space by neglecting higher order PC components have the effect of reducing noise components in the image thereby increasing the detection of targets. However, in both spaces we were not able to find the red panel in shadow target without introducing noise in the detection (lower thresholds).

Regarding the truth mask, we can say that the more accurate the truth mask is the more accurate performance analysis we can get. And the choice of truth mask using ENVI software is subjective which will affect more or less of our performance metrics i.e. ROC curve.

One more optional analysis of the effect of using SVD vs built in function to calculate the inverse during the application of detection algorithm is presented below:

Effect Of using SVD for Inverse:

During the calculation of Inverses of Covariance and Correlation metrices in detector algorithm I used the built in function in **Python NumPy** module. On using SVD for same calculation I didn’t find much change in result of detection in PC space. However, there was slight improvement in detection in original space. This can be related to the fact that **SVD** can provide more stable and accurate results when calculating inverses of matrix specially for ill-conditioned matrices. But in case where a little improvement in the AUCs and detections doesn’t affect the goal or application of target detection (which is mostly the case in real life) using built in function work like using the SVD. Maybe for critical detection tasks we can switch to SVD for inverse calculation. The difference in the detection performance in original space can be shown in figures below:

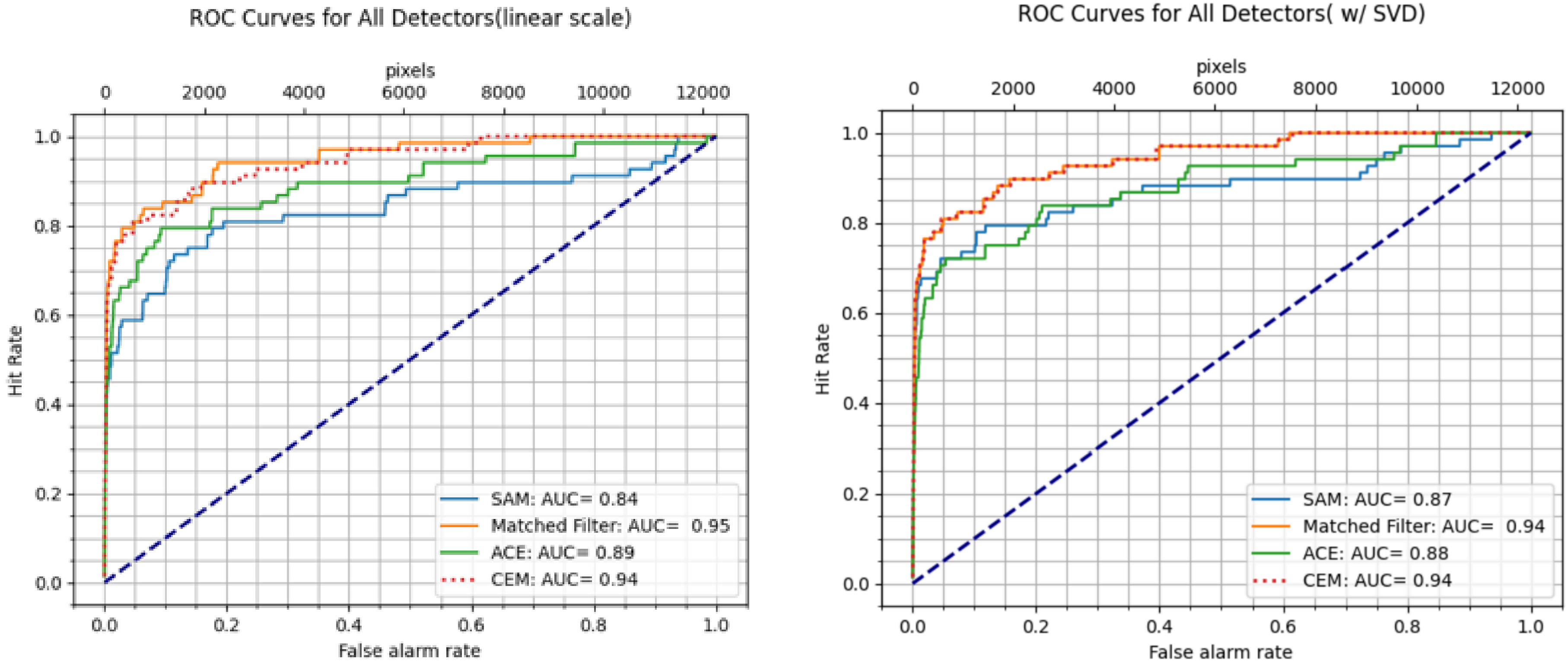

Fig 16: Performance by using SVD for inverse calculations