“Clustering Algorithms” Project

Christos Petrou

ID: DS2120014

Athens 2019

Introduction

The purpose of this project is the process of Hyperspectral images (HSIs) for clustering methods. In this assignment there are three algorithms that are used, the K-means, the fuzzy c-means and the possibilistic c-means. The hyperspectral image that is used is shown in Figure 1, it called "Salinas", which depicts an area of the Salinas valley in California, USA. It is a 150x150 spatial resolution HSI and consists of 204 spectral bands (from 0.2μm – 2.4μm) and its spatial resolution is 3.7m (that is, the HSI is a 150x150x204 cube). Thus, a total size of N = 22 500 sample pixels are used, stemming from eight ground-truth classes: "Corn," two types of "Broccoli," four types of "Lettuce" and "Grapes," denoted by different colors in the following Figure 1. Note that there is no available ground truth information for the dark blue pixels.

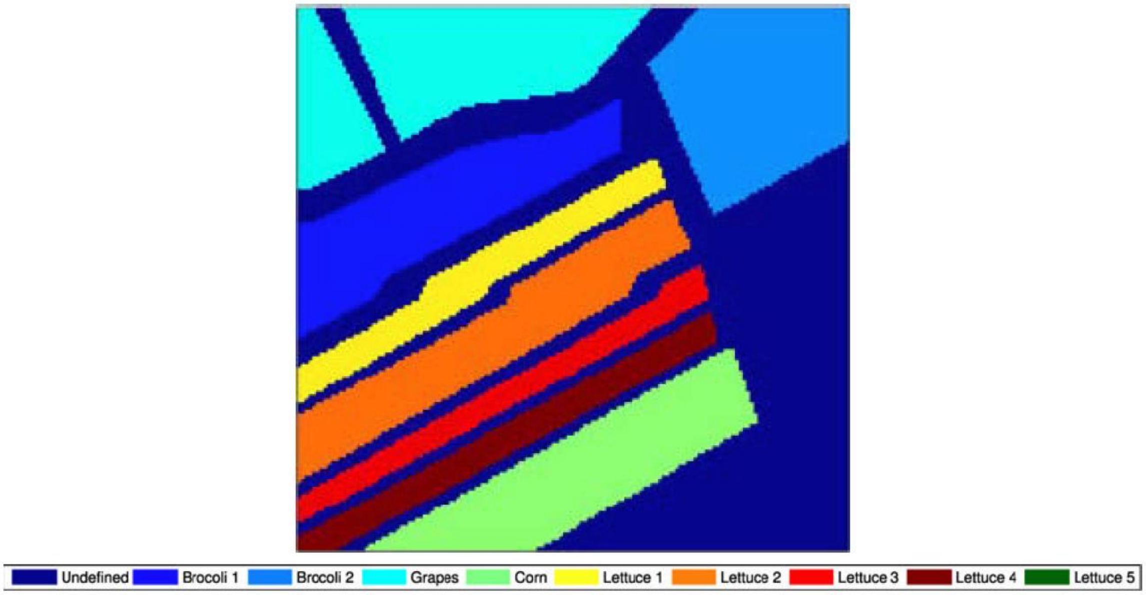

Figure 1 The Salinas HSI, and the areas that it depicts.

The code for the project was developed in MATLAB. The way this project is going to be presented is by analyzing each algorithm separate and, in the end, we are going to compare them together.

In general, the HSIs are difficult to graphicly be represented because they have a lot of feature, which means they cannot be depicted in the 2-D or 3-D. But there are many ways to approximate the multidimensional vectors in the 2-D or 3-D space. In this project Principal Components Analysis (PCA) was used, so that we can better visualize our data and also to make the algorithms depend in less values.

For every algorithm we do the clustering for both data, with PCA and without. In the “Salinas” HSI every pixel consists of 204 values, for the spectral information, also known as the spectral signature, as shown in Figure 2.

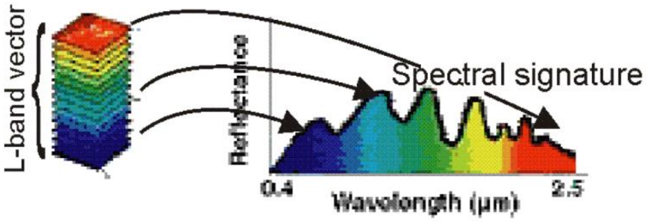

Figure 2 Spectral signature of a pixel from an HSI image.

When we perform PCA in the given data we can see in what percentage every principal component can estimate the variance of the raw data. In the “Salinas” HSI we get :

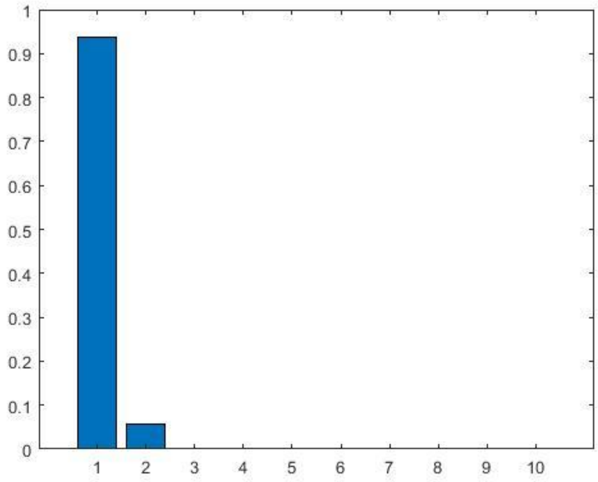

Figure 3 The PC percentages, the y-axis are the percentages, the x-axis is the principal components. Only the first 10 PC are shown.

As we can see only the PC1 and PC2 are needed to explain the variance at 99,5% of the raw data. Also now we can plot our data after this transformation, as shown in the Figure 4. What we can see is that the data are forming what we could consider clusters, we are a given the initial clusters before the transformation so we know the clusters in this PC plot(Figure 5). As we can see there are clusters that compact and clusters that not compact at all.

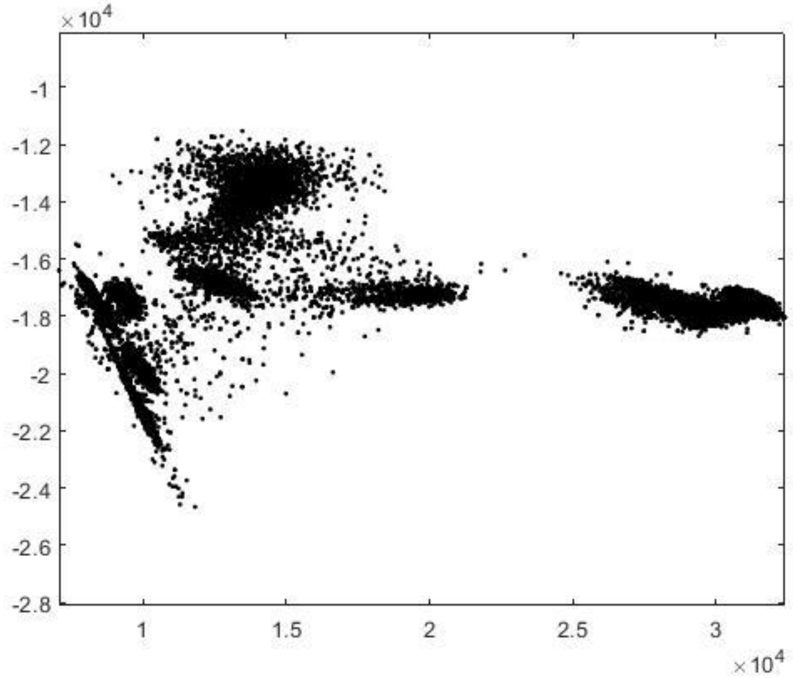

Figure 4 The data after the PCA. The x-axis is the PC1 and y-axis is the PC2.

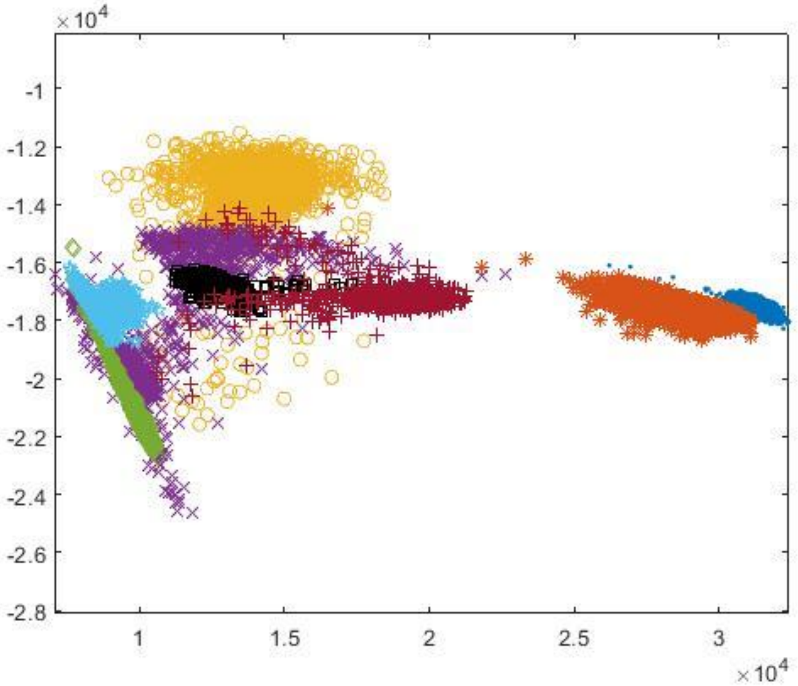

Figure 5 The data after PCA and in witch cluster they belong.

When we plot the first three PCs as shown of the points we get the Figure 6.

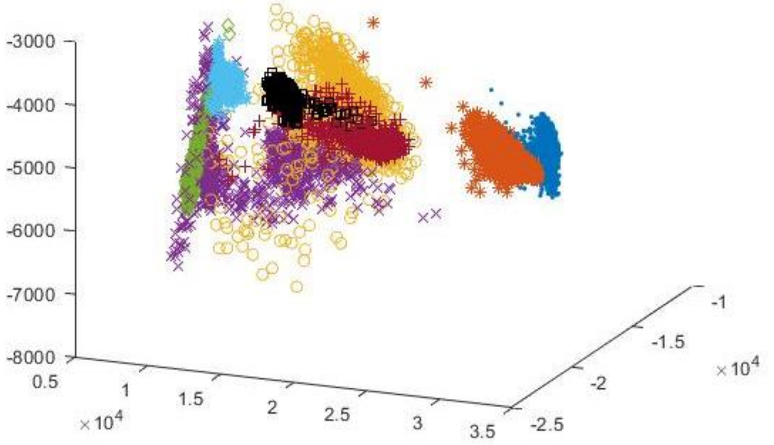

Figure 6 Data points using the first three PCs and their cluster.

But when convert these points back to the original image (Figure 7) we see that there are some regions that should be in the same cluster (so have the same color), that are not the same in values. (Figure 8). This may make the algorithms incapable of distinguishing the correct cluster.

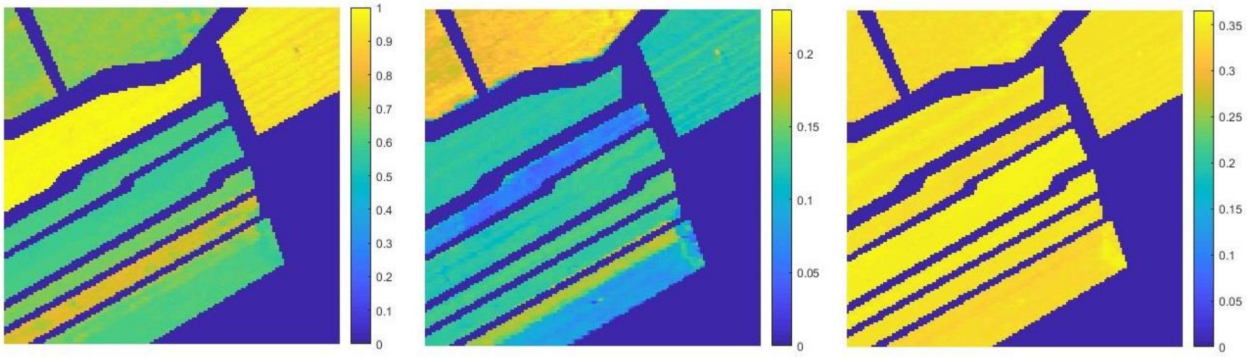

Figure 7 The three first PCs converted back to the original image

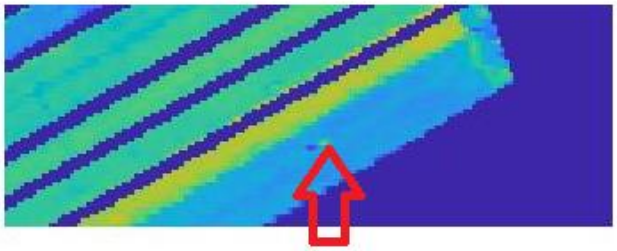

Figure 8 The difference in values for data from the same cluster may lead to false classification.

K-means

Using the k-means algorithm we can find the number of clusters that exist without needing to know that information in advance. From the "elbow point" that we can see in Figure 6, we can say that there are 8 clusters.

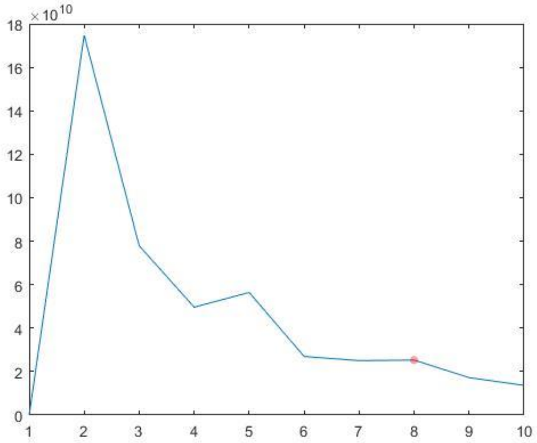

Figure 9 The elbow point. The y-axis is the cost function, the x-axis is the number of clusters.

- Raw data

The accuracy of the k-means algorithm really depends on the initialization of the representatives of the cluster. The mean value of the k-means is **59%-62**% (as shown in Figure 6). But there are some times that the accuracy goes up to **71%** but also it can drop to **40%**.

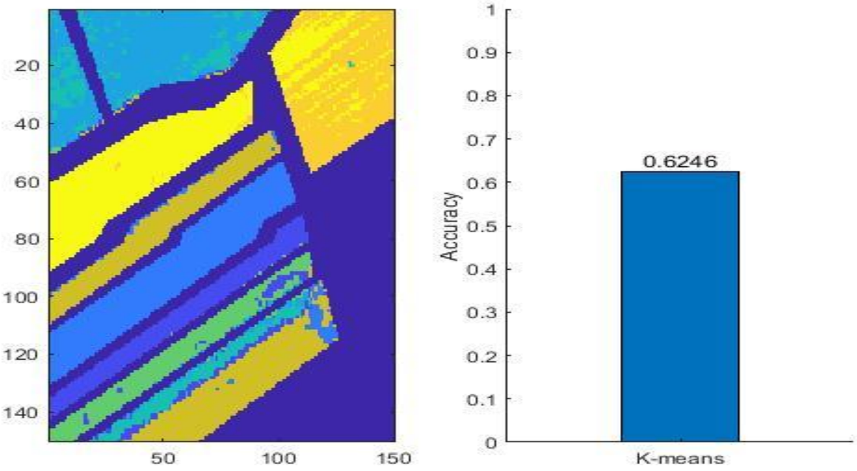

Figure 10 Accuracy of the k-means algorithm.

- PCA data

When we use the PCA data we can visualize the clusters and take a better picture on what is happening. Now the accuracy is more constant at **62-69**%, but sometimes times it goes up to **83%.**

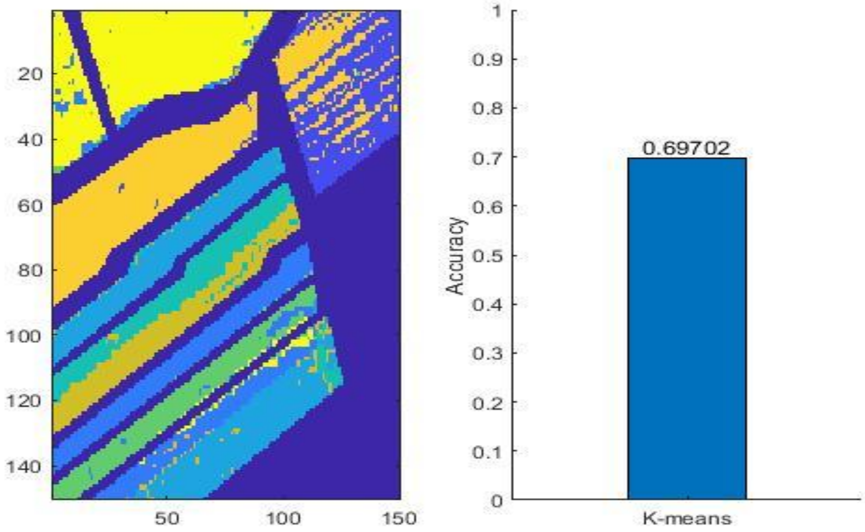

Figure 11 Accuracy of the K-means on PCA data.

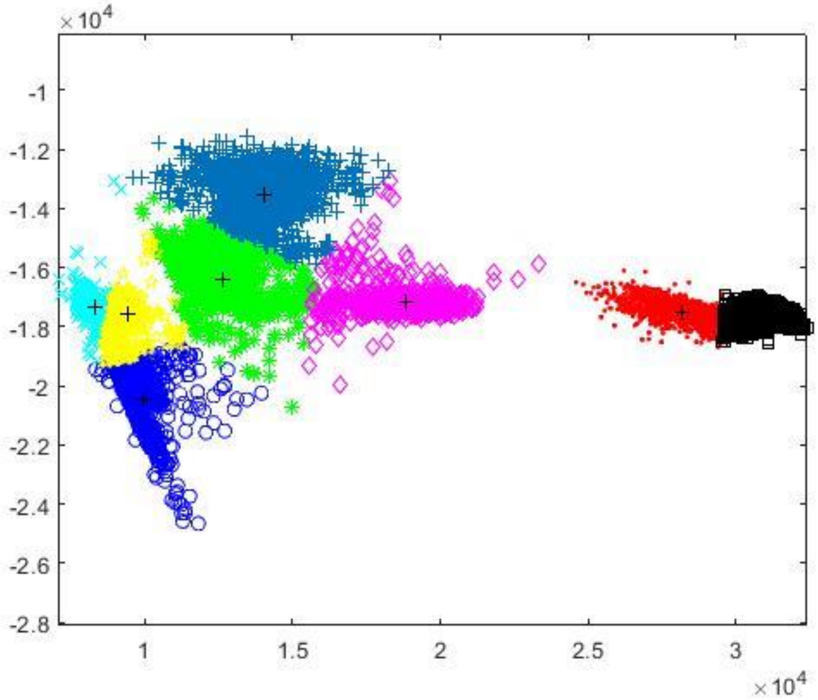

Figure 12 The formed clusters of the k-means algorithm on the PCA data.

In general the k-means algorithm can only group hard compact clusters and that is the reason it doesn't work that consistent in this HSI, as we can see in Figure 5 there are some compact clusters witch the k-means can distinguish but there are some that are not compact so the algorithm cannot work. Also the outcome of the clustering is heavily dependent from the initialization of the clusters.

Fuzzy c-means

We cannot find the number of clusters using the cost function of the fuzzy c-means as we can see in Figure 10. Because there is no “elbow point”. So we have to take as prior knowledge the number of clusters.

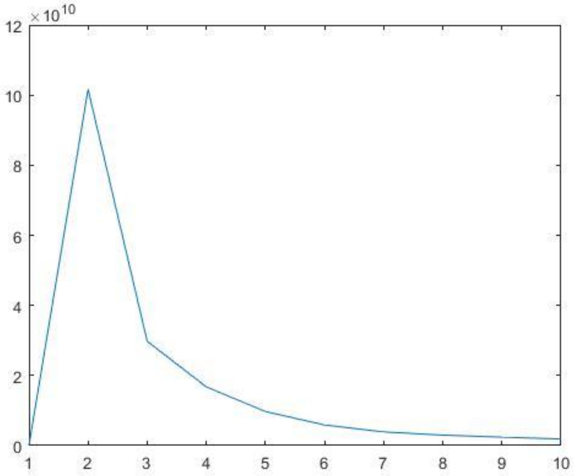

Figure 13 There is no elbow point. The y-axis is the cost function, the x-axis is the number of clusters.

- Raw data

The initialization of the representative here is also done by assigning a random point from the dataset to it. The accuracy of the algorithm is **60%** (as shown in Figure 10), but sometimes it can go up to **70%.**

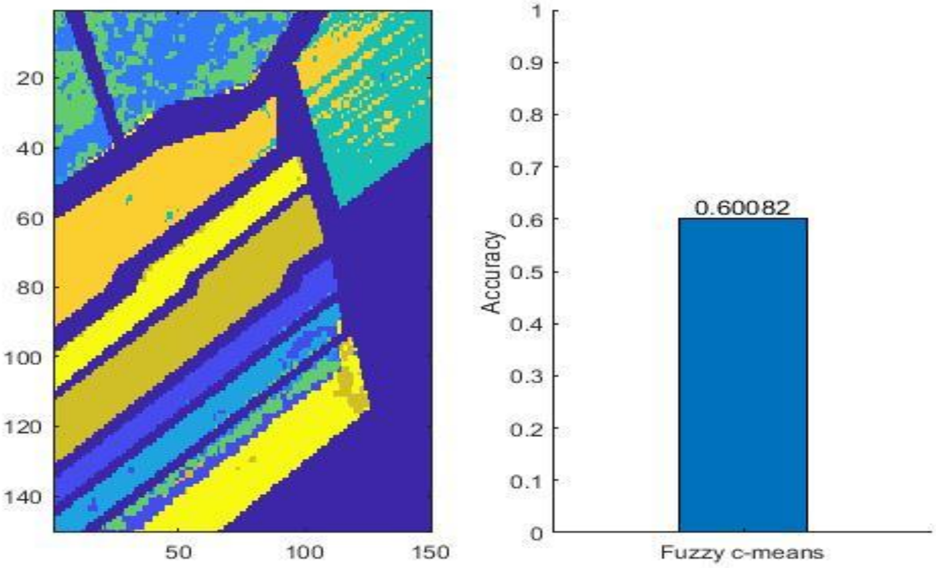

Figure 14 Accuracy of the fuzzy c-means algorithm.

- PCA data

What we notice here is that the accuracy does not change much, no matter the representatives initialization. The accuracy is around **58%-62%** (such an instant can be seen in the Figure 12).

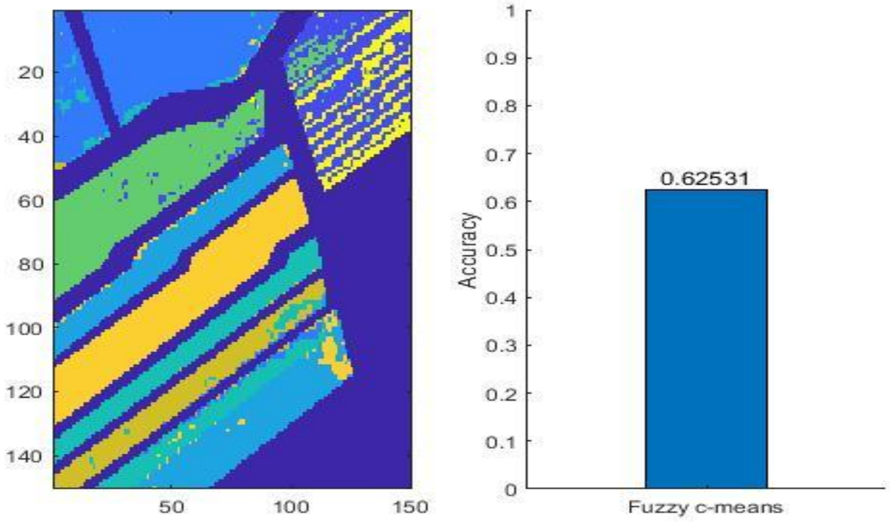

Figure 15 Accuracy of the fuzzy c-means on PCA data.

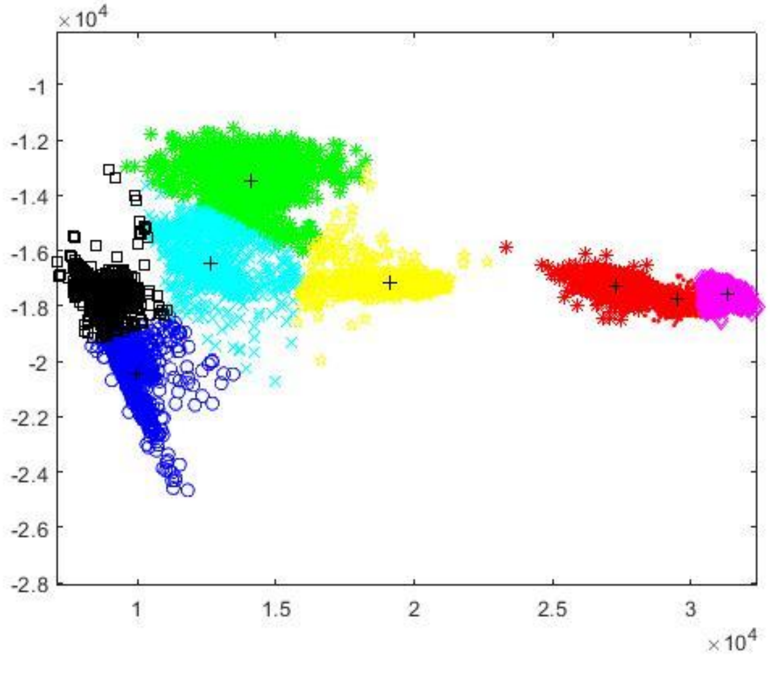

Figure 16 The formed clusters of the fuzzy c-means algorithm on the PCA data.

In conclusion this algorithm is much more stable on its accuracy in contrast with the k-means. The problem here is that we must know the number of clusters beforehand. Also, the fuzzy c-means is good at distinguishing compact clusters and that is the reason for the low accuracy.

Possibilistic c-means

For this algorithm we also need to know the number of clusters beforehand. As there is no way to understand the number of cluster through the PCMA algorithm. What we see is that the algorithm is greatly improved from the PCA data transformation.

- Raw data

In Figure 14 we see that that the PCMA accuracy is poor. The accuracy is around **38%-43%**.

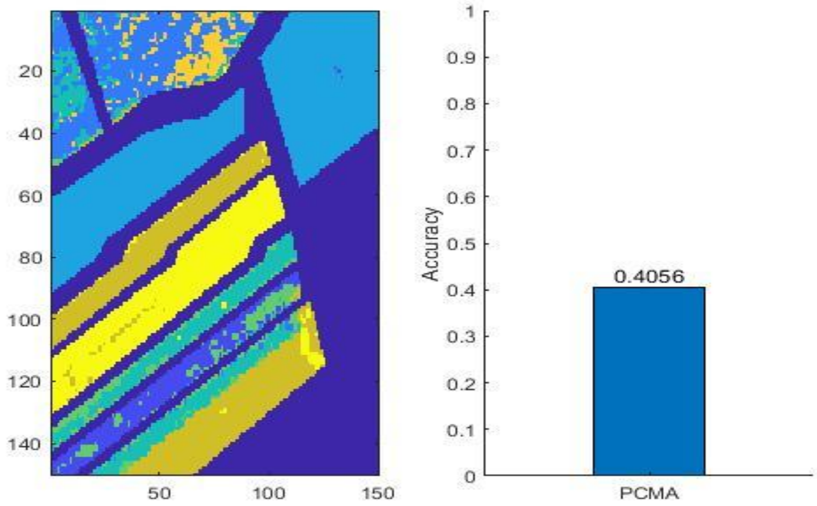

Figure 17 Accuracy of the PCM algorithm.

- PCA data

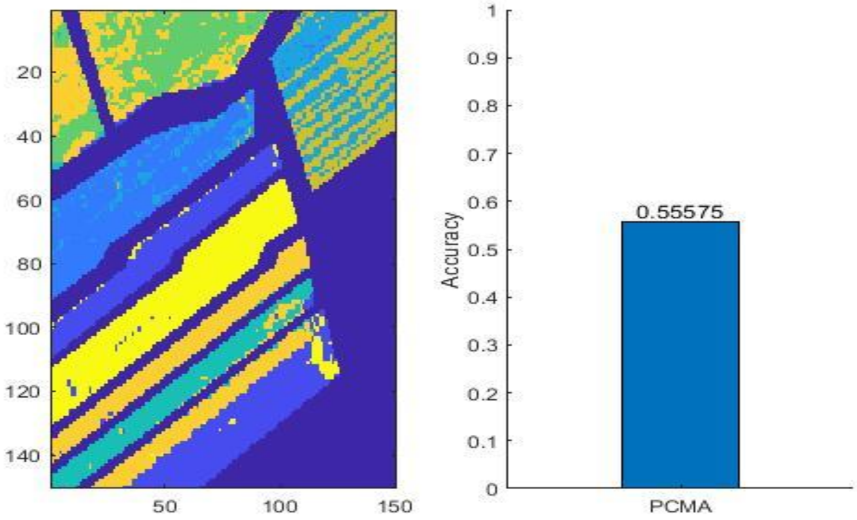

Figure 18 Accuracy of the PCMA on PCA data.

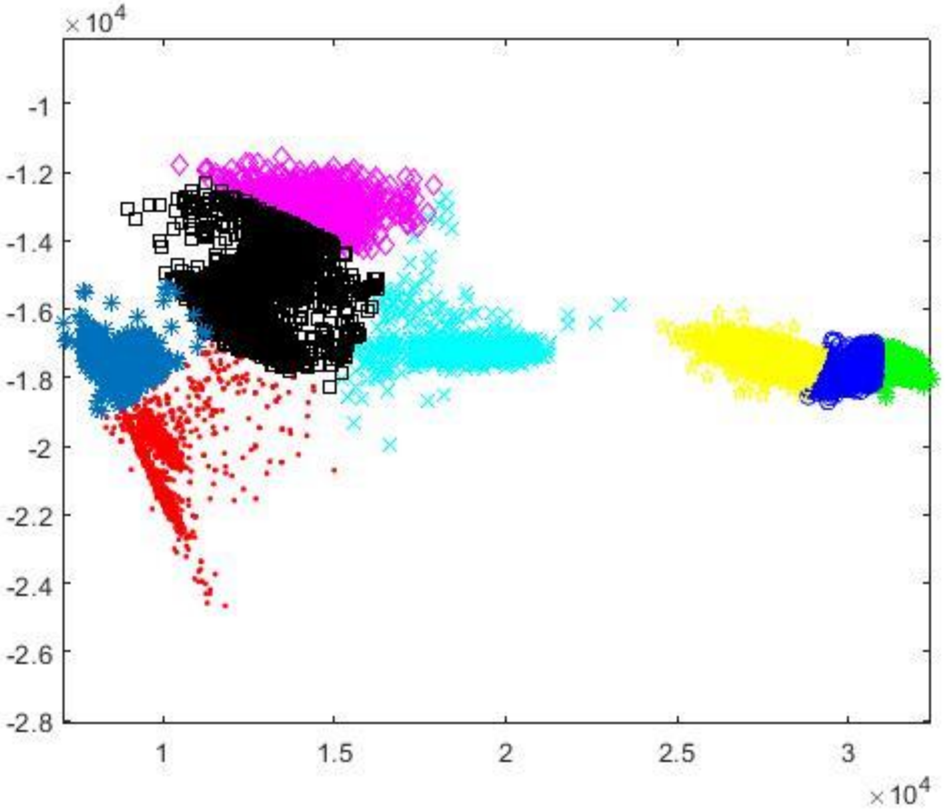

Figure 19 The formed clusters of the PCM algorithm on the PCA data.

The PCMA greatly improves from the PCMA transformation, the accuracy is more consistent around **60%-65%.**

Probabilistic

For this algorithm we need the number of clusters beforehand. This algorithm is good at recognizing compact clusters we different densities. But the biggest drawback of this algorithm is that is the slowest of all the tested algorithms, because of the big computational resources it needs. The number of iterations used is 50.

- Raw data
  When we run the algorithm we the raw data the algorithm cannot terminate because of the big matrices that are used.

- PCA data
  After we feed the PCA data the algorithm is still slow, but it terminates. The results are shown in Figure 20 and Figure 21. The accuracy is steady around **60%.**

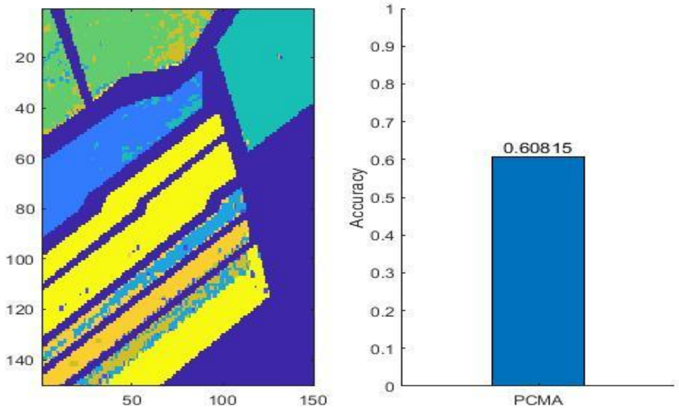

Figure 20 Accuracy of the Probabilistic algorithm on PCA data.

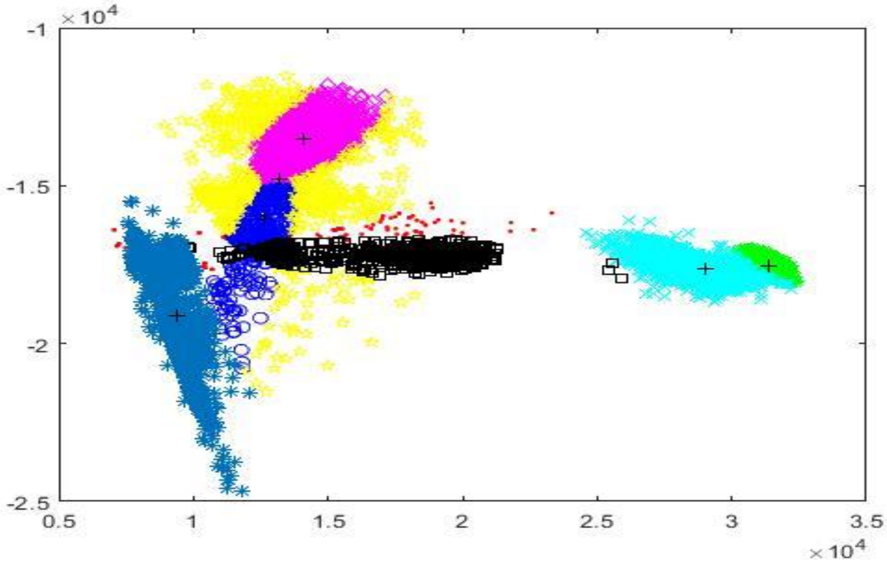

Figure 21The formed clusters of the Probabilistic algorithm on the PCA data.

Testing the algorithms all together

For the testing of all the algorithms, the PCA data were used also the initialization of the representatives is the same for all the cases. The most_dist_repre code was used, which was given through this course, for the initialization of the representatives.

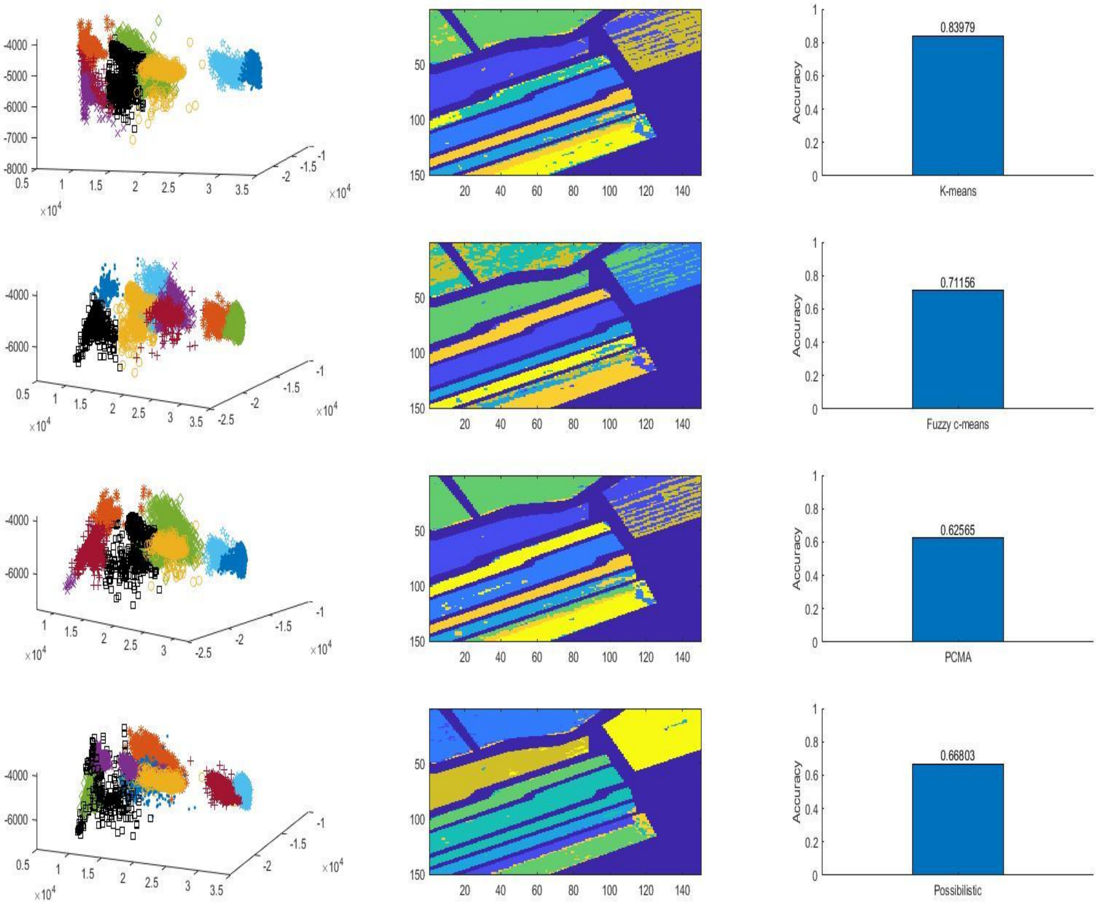

Figure 22 All the algorithms using the same initiation data.

The algorithm with the best accuracy was K-means with 83% accuracy. What we see is that the area that was mentioned in the introduction. None of the algorithms was able to distinguish the real cluster, but was expected for the values of the PCA.

In conclusion

What we see, is that each algorithm performs different on the specific image. There were many problems that occurred doing this project. First was the multidimensional nature of the HSIs. That was solved by performing Principal Component Analysis on the input data. This way we could plot the clusters and the solving time was reduced.

The next big problem was that the initialization of the representatives of clusters was important for the algorithms to find the right clusters. At first, they were initialized in zero but to output was given, the only thing that was done was to initialize the representatives randomly on data point. There are to improve this fact by applying the initialization depending the algorithmic.