Project

Student name: *Sfrintzeri Konstantina, ds2200015*

Course: *Clustering algorithms* – Professor: *Dr. Koutroumpas*
Due date: *January 19th, 2021*

Hyperspectral Images processing.

For the purpose of this study we will use Hyperspectral images (HSIs) that depict a specific scene at several (L) narrow continuous spectral bands. This images can be represented by a MxNxL three-dimensional cube, where the first two dimensions correspond to the spatial information, while the third corresponds to the spectral information. In our case the dimensions are 150x150x204, so the specific scene is depicted by 204 spectral bands. The true labels of every sub-region is given so we will be able to extract further conclusions about the accuracy of every clustering algorithm that will be used.

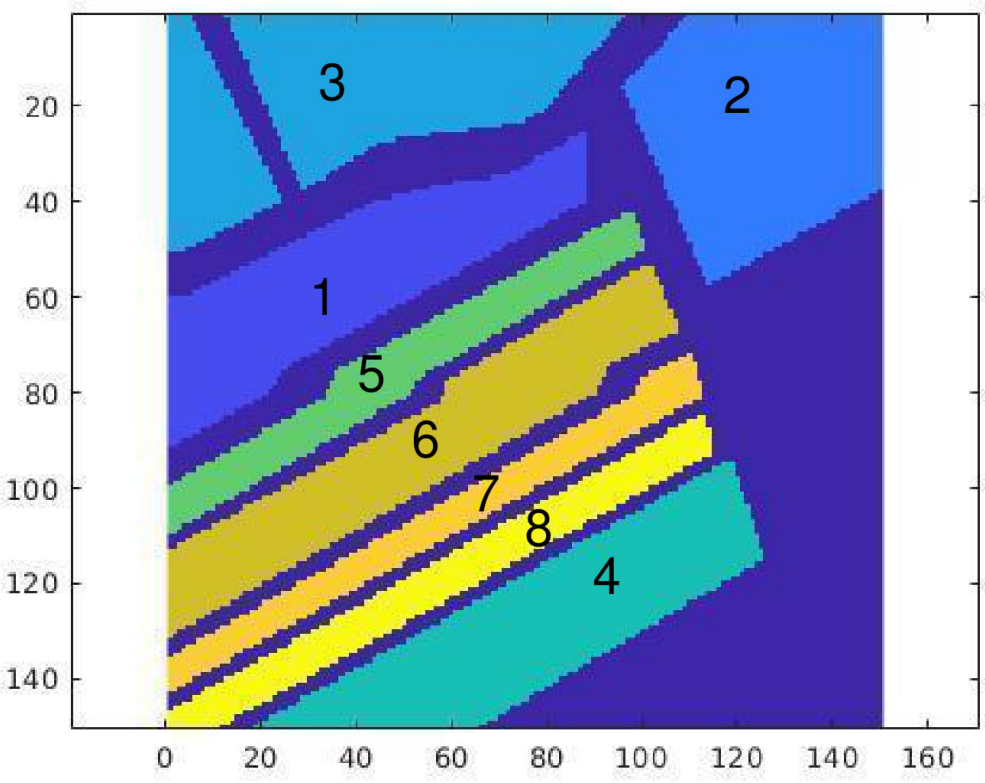


Figure 1: True separation of classes.

The numbers that are upon each class are the number that represent them in file Salinas Labels and from now on when we will use these numbers we will refer to the classes that they represent.

In order to investigate the extent to which it is possible to properly separate the classes based on the Hyperspectral images, we will apply pca to the data provided to us and examine the information that is given from the first 4 pcs.

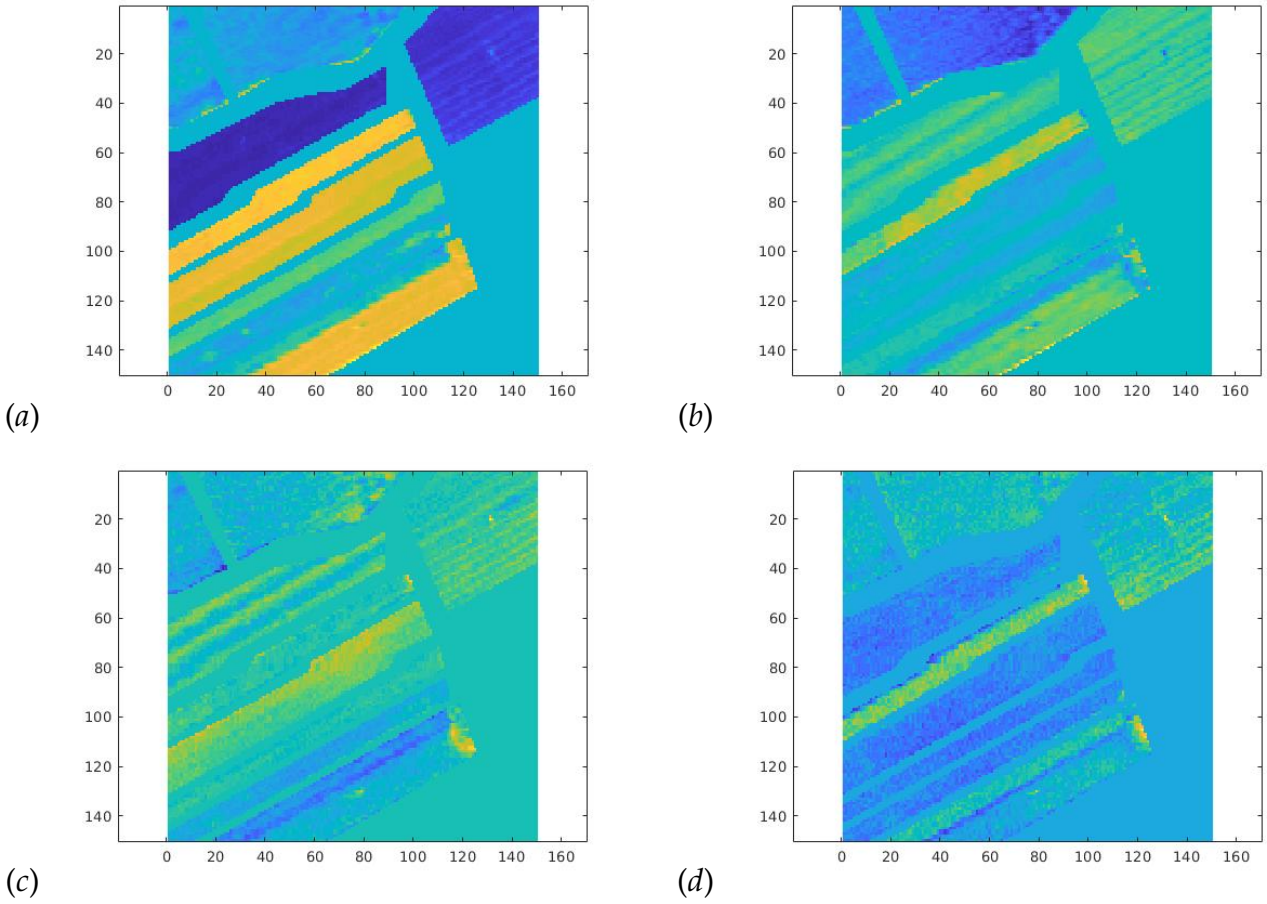


Figure 2: Pcs from principal component analysis

Case (a): This is the first pc that occurs from principal component analysis. In this case we can distinguish easily classes 1, 2, 5, 6, 7 and a part of class 4. Classes 3 and 8 are not easy distinguishable and class 2 seems to have quite similar color with class 1.

Case (b): This is the second pc that occurs from principal component analysis. In this case the most distinguishable classes are class 3, which was not distinguishable from pc 1, and class 6. Also class 4 seems is also distinguishable, but it is divided in 2, so we could mistakenly consider that in this region lay 2 classes.

Case (c): This is the third pc that occurs from principal component analysis. This pc gives less information compared to the previous ones. Once again class 6 stands out and the half part of class 4. The fact that class 4 doesn't appear unit will definitely create a problem when we will be implementing the clustering algorithms.

Case (d): This is the forth pc that occurs from principal component analysis. Finally in this pc classes 1, 5, 6, 7, 8 appear easy to discern so probably they will be easily be detected from the clustering algorithms.

With the use of principal component analysis we will reduse the dimensions of our problem so that it will be more fast and easy to handle. The plot below depicts the percentage of the initial information that is given by the pcs.

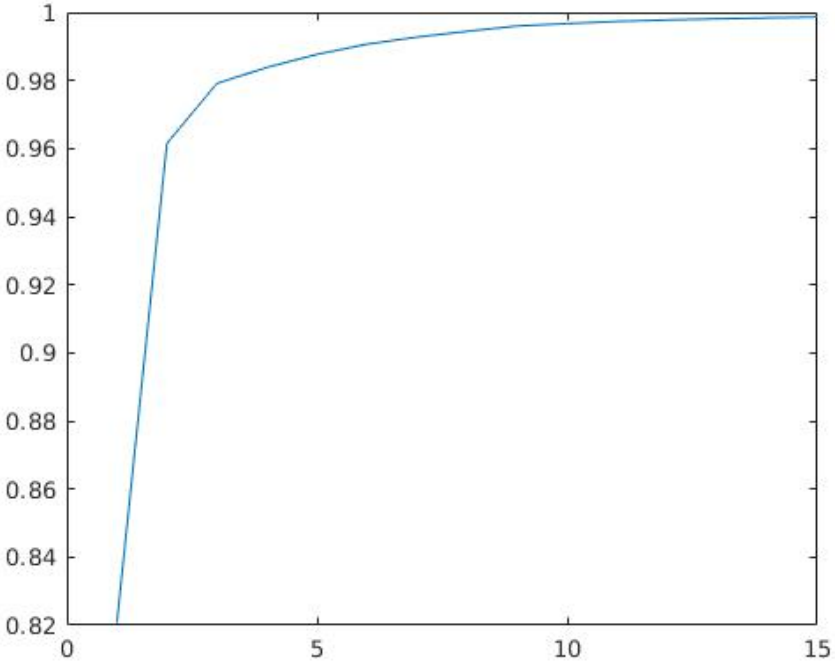


Figure 3: Information that is given by the pcs.

The information that is given from the first 9 pcs is the 99.6% of the initial information, so we will execute all the algorithms by using the 9 pcs.

Execution of the clustering algorithm

k-means

In k-means there are two main issues, the number of clusters must be known a priori and also the different initial partitions may lead k-means to produce different final clusterings, so the initialization of representatives plays a crucial role. In order to overcome these problems we will plot the cost function J versus the number of clusters, and the correct number of clusters will be given by the spot where a knee is formed in this plot. The best initialization for k-means is by selecting the k most distant points as initial representatives. Apart from that we will use as initial representatives the representatives that are computed from MBSAS algorithm and the we examine also the results that are given from random initialization.

k most distant points in the dataset as representatives

As we can see from this plot a knee is formed around value 6, bit even after this value the cost function seems to be decreased so we will also try values 7, 8 and 9.

6 Clusters

Case (a) is the map of clusters that are created from k-means and the numbers correspond to the real classes as they appear in *Figure 1*. *Case (b)* is the confusion matrix of real labels and the predicted labels. From this map we can easily discern that class 3, 6, 7 and 8 are correctly classified. Class 1 and 2 have been assigned to one cluster and

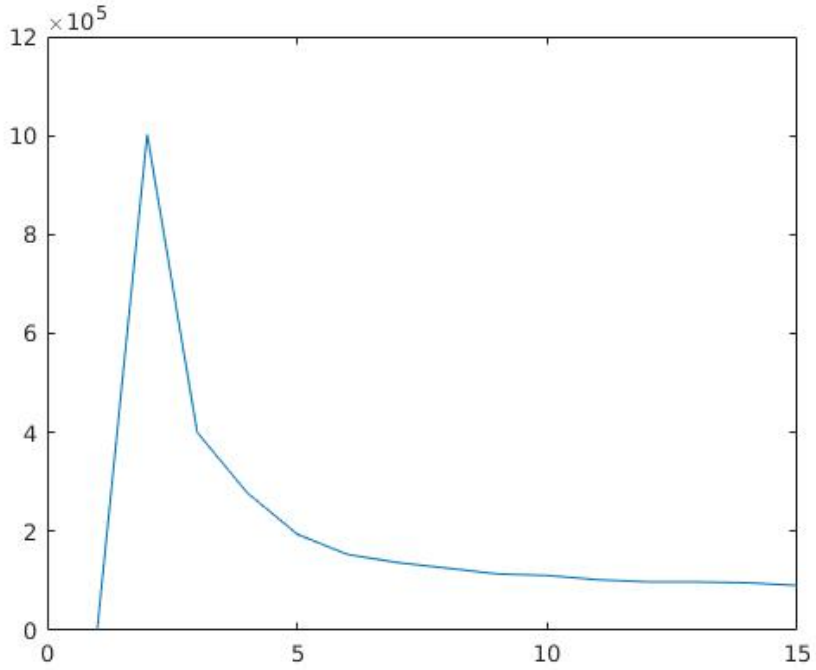
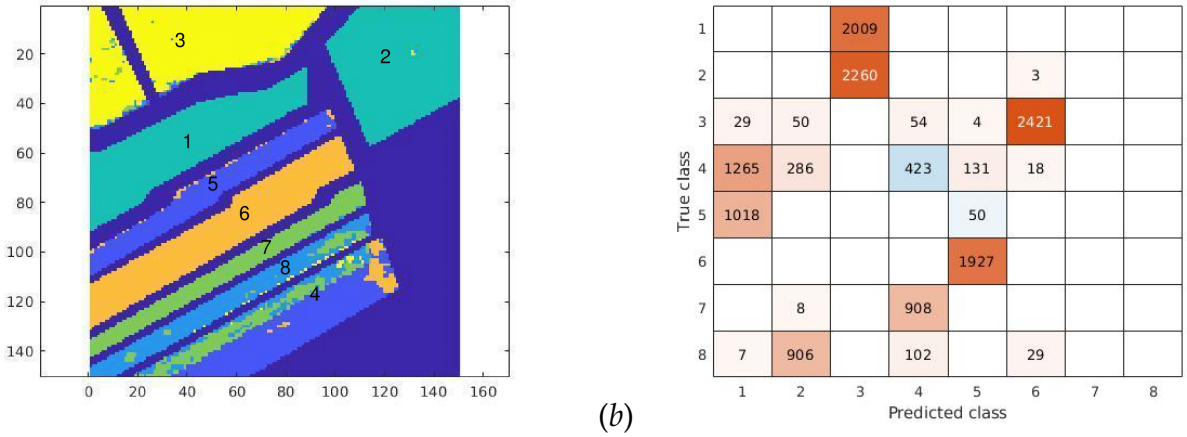


Figure 4: Cost function J versus number of clusters - Initialization with k most distant points

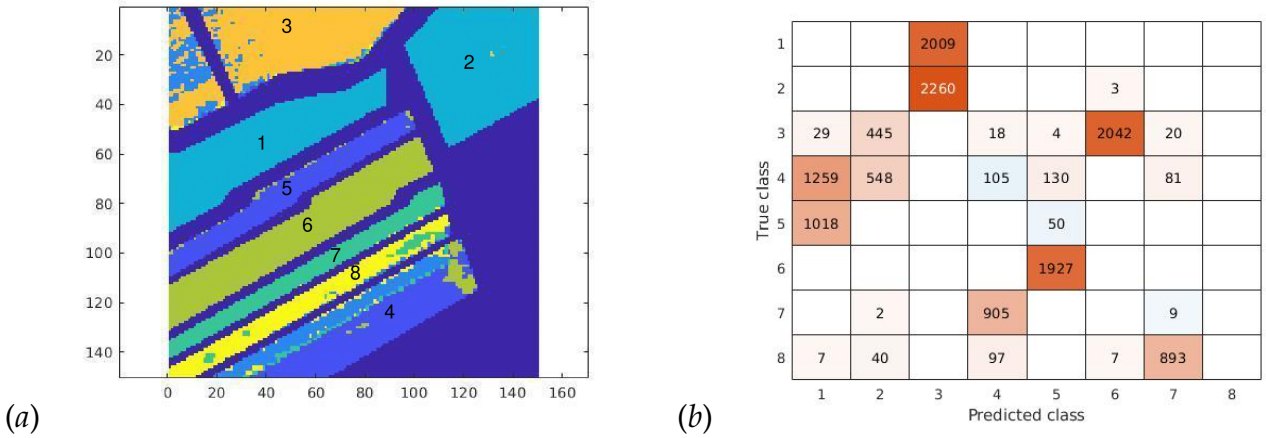


(*a*) (*b*)

Figure 5: Distribution of 6 clusters with k-means, initialization with k most distant points

class 4 is divided. the first half seems to be assigned in the cluster that contains class 5 and the second half is assigned in the cluster that contains class 7. The fact that class 1 and 2 are assigned into one class is reasonable because their spectral images are really close. Also the division on class 4 is something that we were expected because as we observed this case in the spectral images when we did the pca. For this case *accuracy = 69.56%*. From confusion matrix we can see that class 4 seems to have the biggest problem because it seems that it is scattered in 5 clusters.

7 Clusters



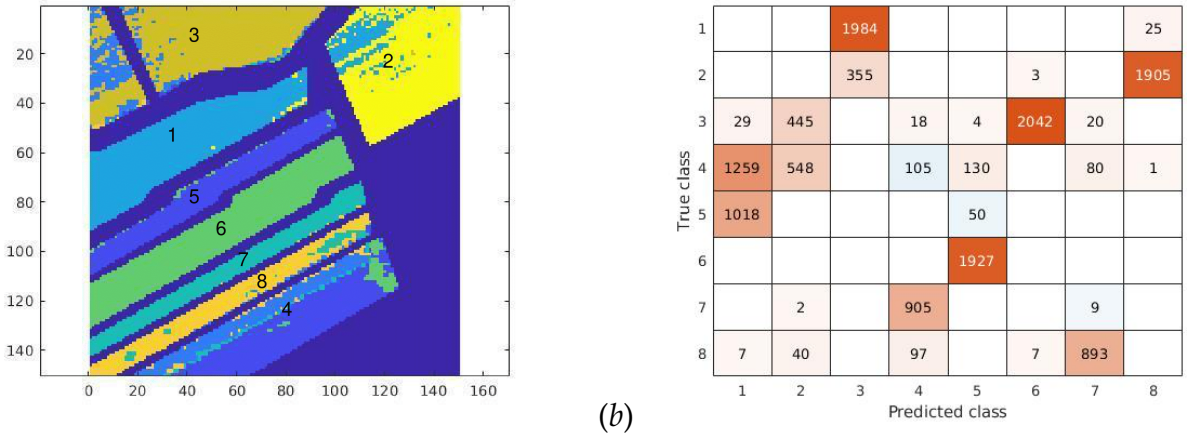
(*a*) (*b*)

Figure 6: Distribution of 7 clusters with k-means, initialization with k most distant points

Case (a) is the map of clusters that are created from k-means for 7 clusters and the numbers correspond to the real classes as they appear in *Figure 1*. *Case (b)* is the confusion matrix of real labels and the predicted labels. In this case the clusters that contain classes 6 and 7 are almost the same with the previous case, and are very close to the truth. Also the cluster that contains class 8 is really close to the truth but it appears that contains some noise too. Classes one and 2 are once again assigned together in the same cluster but they also don't contain any noise. This proves not only that they are close as far as it concerns the spectral images, but that they are also far from the other regions. Class 4 is again divided in two now the one half consist a unique cluster. The fact the the one half of class 4 is once assigned to the cluster that contains class 5 certifies that they are close spectrally. For this case *accuracy = 68.97%*. Despite the fact that the number of clusters in this case is closer to the true number of cluster, the accuracy is less and this is maybe caused due to the fact that the cluster that contains 3 or class 8 contains too much noise.

8 Clusters

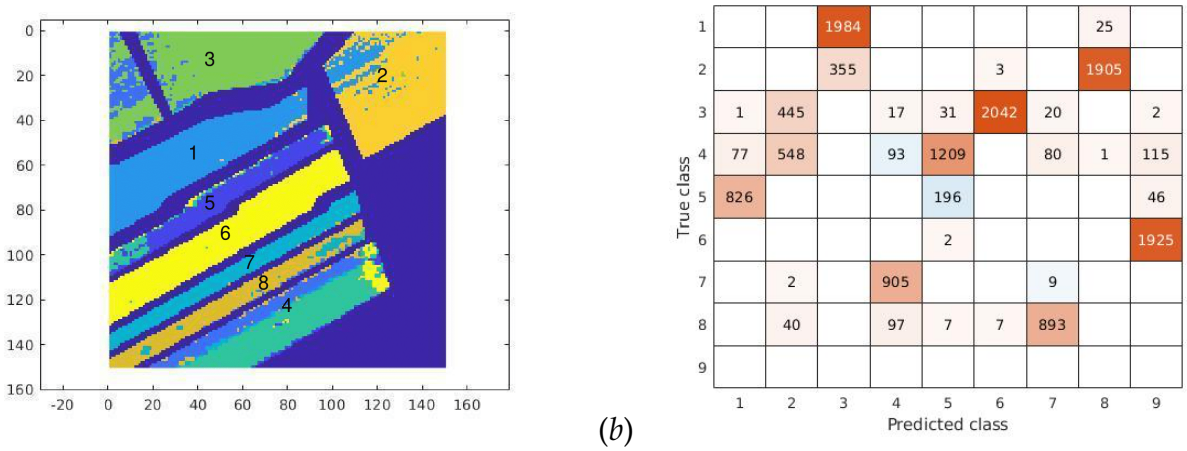
Case (a) is the map of clusters that are created from k-means for 8 clusters and the numbers correspond to the real classes as they appear in *Figure 1*. *Case (b)* is the confusion matrix of real labels and the predicted labels. This one appears to be the best case as *accuracy = 80.69%*. Now class 1 and class 2 are assigned to different clusters but as we can see from confusion matrix and the map too a significant amount of class 2 is assigned in the cluster that contains class 1. So the problem of distinguishing this two classes has not been eliminated. Class 6 is all assigned into one cluster as we can see from confusion matrix, as in row 6 there is only the 1927, which the the total number of pixels that correspond to class 6. However the cluster that contains class 6 contains some pixel "noises" too. Class 7 seem to be detected too. There is a problem again in class 4, as it is divided in two parts. Also as we can see from the confusion matrix, the biggest part with the 1259 pixels is assigned to the cluster that contains class 5, and the rest 548 pixels are those which create the eighth cluster, along with 445 pixels that belongs to class 3. So the clusters that are created in this case approach the truth but they are not perfect.



(*a*) (*b*)

Figure 7: Distribution of 8 clusters with k-means, initialization with k most distant points

9 Clusters



(*a*) (*b*)

Figure 8: Distribution of 9 clusters with k-means, initialization with k most distant points

Case (a) is the map of clusters that are created from k-means for 9 clusters and the numbers correspond to the real classes as they appear in *Figure 1*. *Case (b)* is the confusion matrix of real labels and the predicted labels. In this case as we expected the only important difference from the previous case is that the part of class 4 that was forming a cluster along with class 5 is now assigned to a new cluster. So the algorithm manages to find all the classes except from class 4, which is divided in two clusters. However, if we combine the clusters that contain class 4 together in one cluster then the final accuracy will be *accuracy = 87.98%*. The difference in the accuracy of this case with the accuracy of the previous case is quite significant. So the 9 clustering method of k-means gives the best results when the initial representatives are the k most distant points in the dataset.

10 Clusters

Case (a) is the map of clusters that are created from k-means for 10 clusters and the numbers correspond to the real classes as they appear in *Figure 1*. *Case (b)* is the

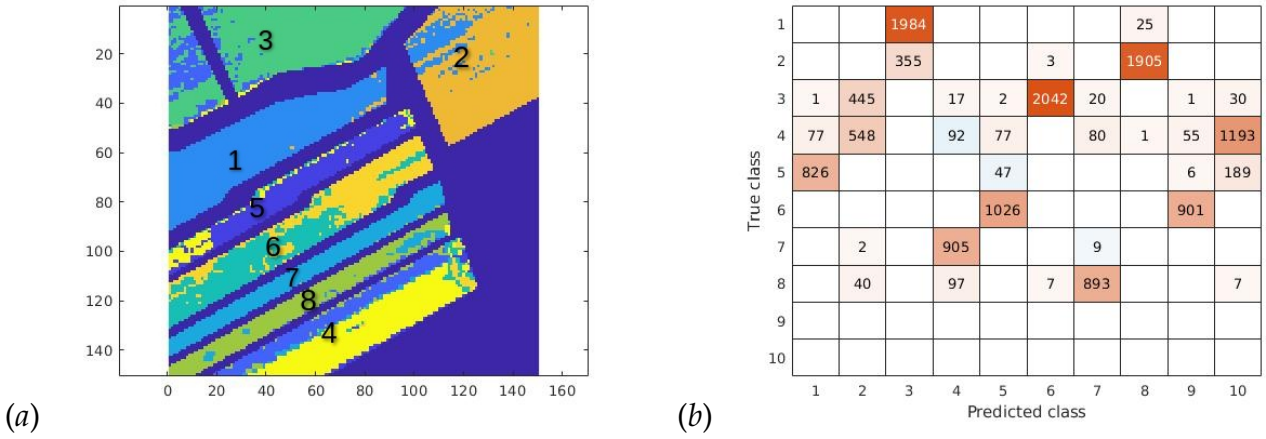


Figure 9: Distribution of 10 clusters with k-means, initialization with k most distant points

confusion matrix of real labels and the predicted labels. In this case, the only difference that exists compered to the previous case is that a new cluster is formed that contains the half part of class 6. So in order to compute the accuracy of this clustering case we will join the 2 clusters that contain the two halves of class 6 together. By doing this the final accuracy is *accuracy = 87.88%*. The results that we get from this method are slightly worst than the case of 9 clusters.

Random initialization

We will examine the performance of k-means by selecting random representatives. Below is the plots of cost function J versus the number of clusters.

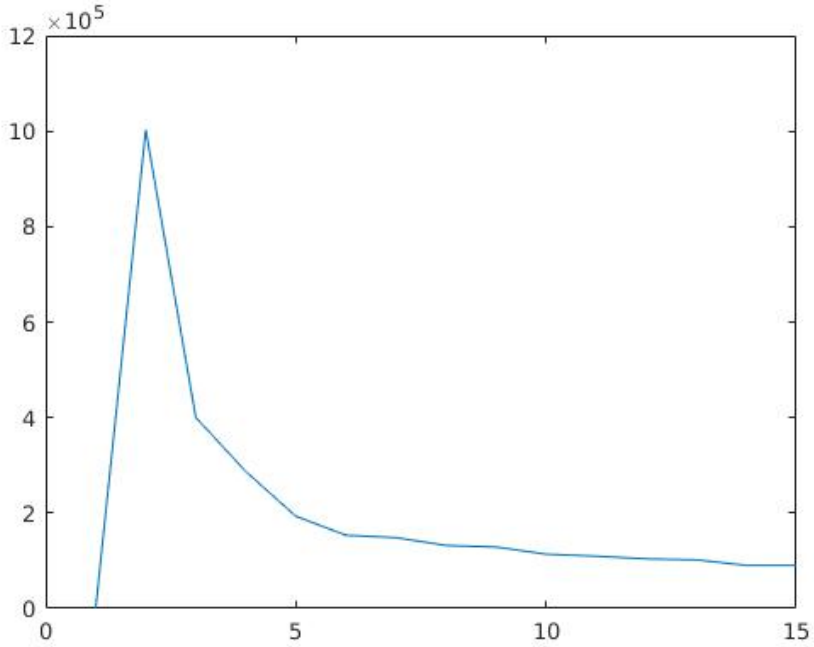
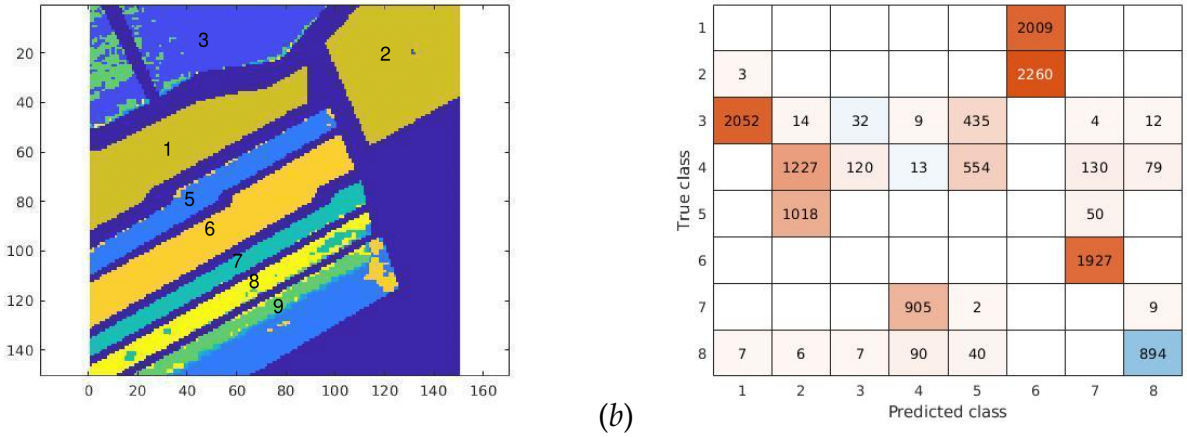


Figure 10: Cost function J versus number of clusters - Random initialization

The numbers in which the line appear to become for flatten is in 8 and 9 so these

are the cases that we will examine.

8 Clusters

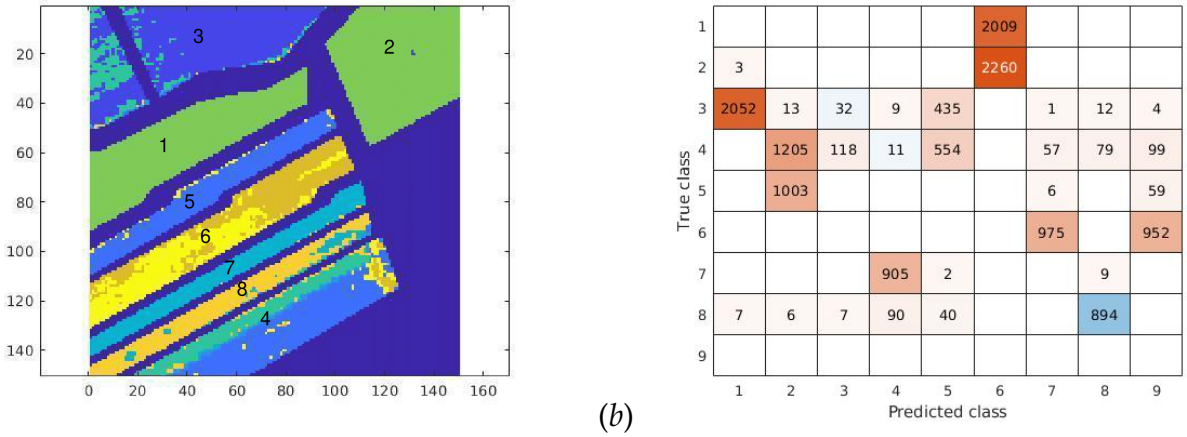


(*a*) (*b*)

Figure 11: Distribution of 8 clusters with k-means, initialization with k most distant points

Case (a) is the map of clusters that are created from k-means for 8 clusters and the numbers correspond to the real classes as they appear in *Figure 1*. *Case (b)* is the confusion matrix of real labels and the predicted labels. Despite the fact that the number of clusters that we demand from k-means to find is equal to the true number of classes the final result is not right as class 1 and 2 are assigned in one cluster and class 4 is divided in 2 clusters. Thus in this case *accuracy = 69.10%*.

9 Clusters



(*a*) (*b*)

Figure 12: Distribution of 9 clusters with k-means, initialization with k most distant points

Case (a) is the map of clusters that are created from k-means for 9 clusters and the numbers correspond to the real classes as they appear in *Figure 1*. *Case (b)* is the confusion matrix of real labels and the predicted labels. This case is quite similar with the previous one, as once again classes 1 and 2 are assigned in one cluster, class 4 is divided in 2 clusters and now due to that fact that we demanded k-means to find 9

clusters class 6 is divided in 2 clusters too. So this case is even worst that the previous one. This fact proves that selecting the most distant points in dataset as representatives is much more efficient that initializing them randomly.

Fuzzy

In fuzzy clustering every datapoint belongs to all clusters up to a certain degree. This fuzzyness becomes possibble due to variable q. The bigger the value q the biggest the fuzzyness of the clusters. This type of clustering algorithm is suitable for cases when the datapoints are too close to each other. The plots below depict the cost function J versus number of clusters for different values of q.

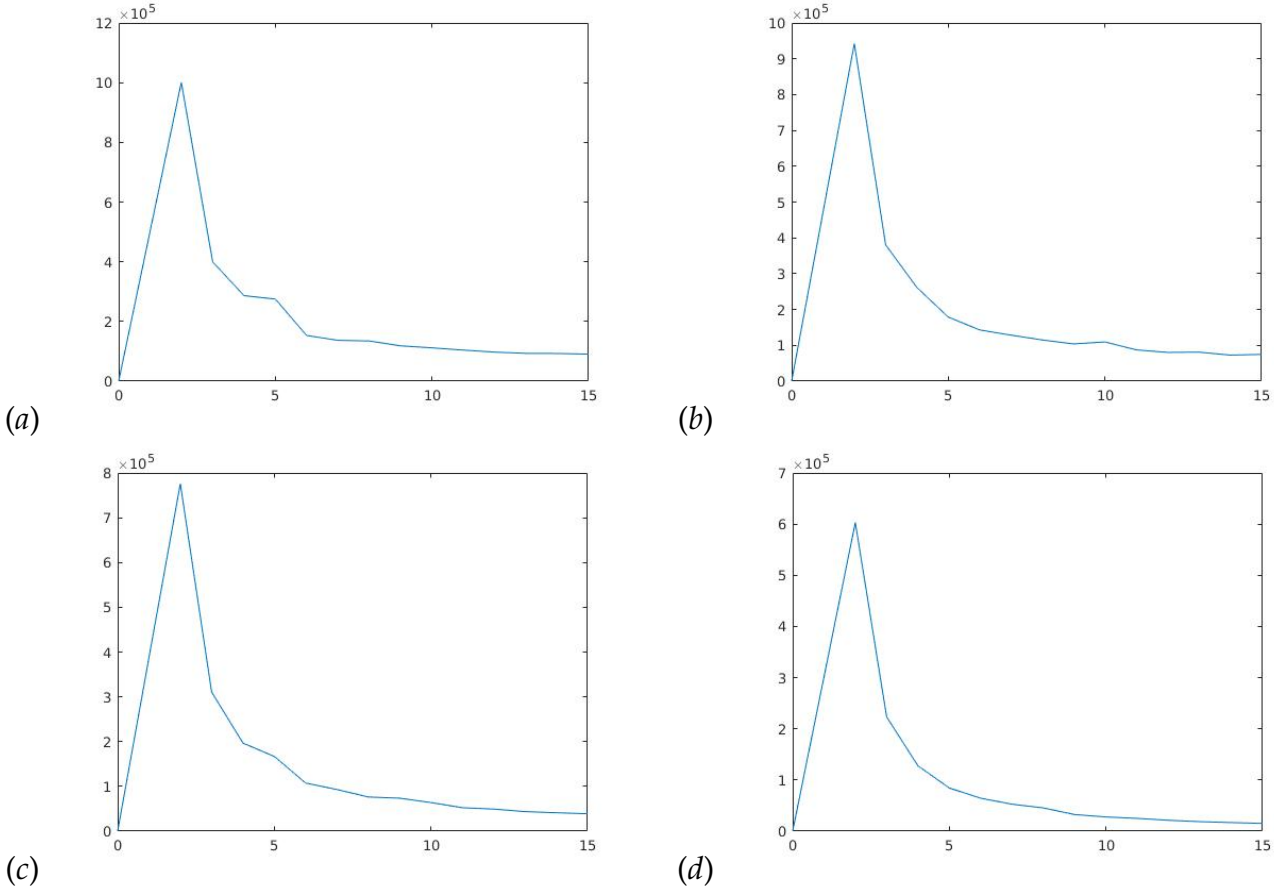


Figure 13: Cost function J versus number of clusters - Fuzzy

Case (a) corresponds to *q=1.1*, *case (b)* corresponds to *q=1.5*, *case (c)* corresponds to *q=2* and *case (b)* corresponds to *q=2*. In all 4 plot we can see that there are elbow points around numbers 7 and 8 so we will execute fuzzy with clusters close to these numbers.

$$q = 1.1$$

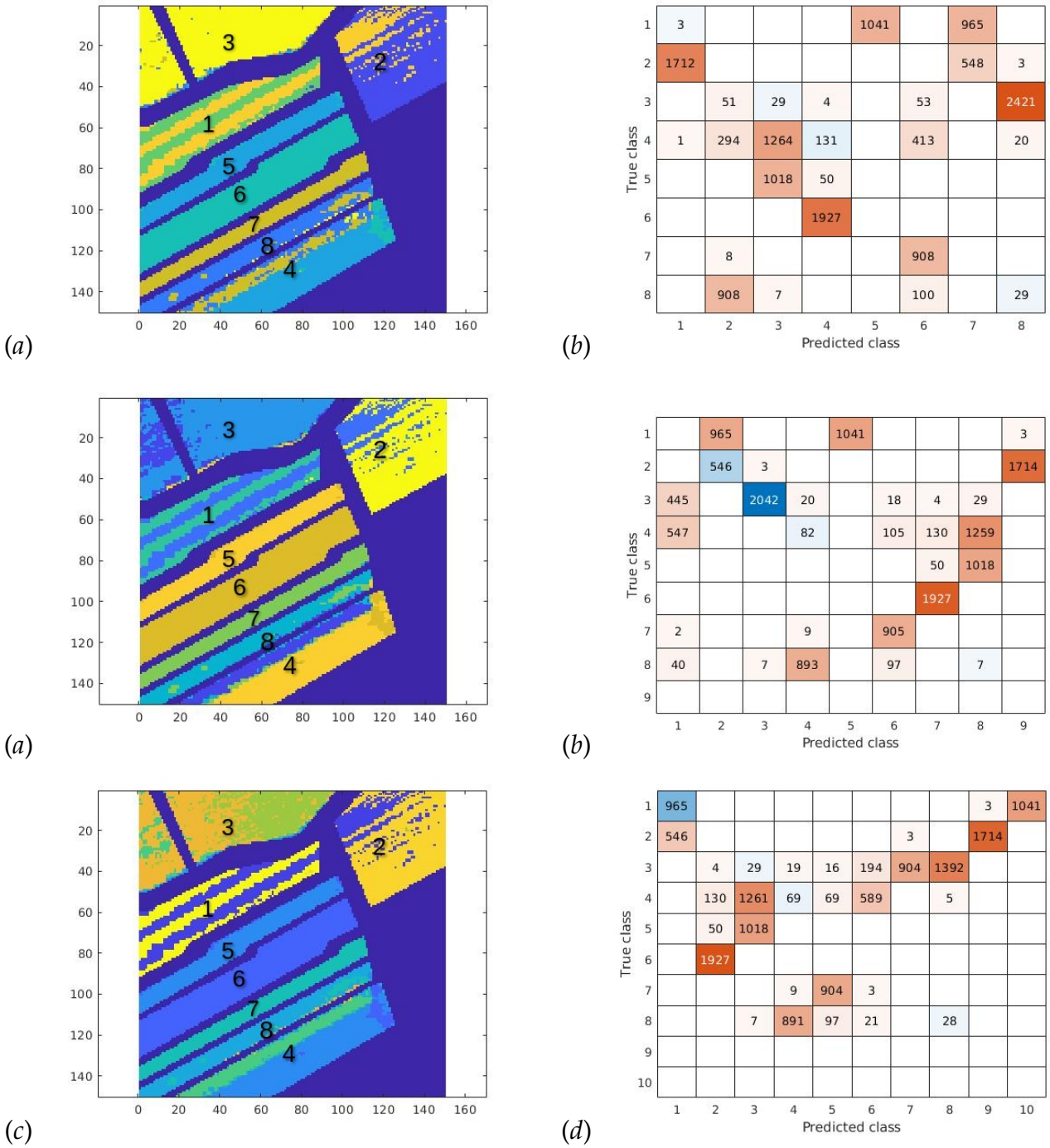
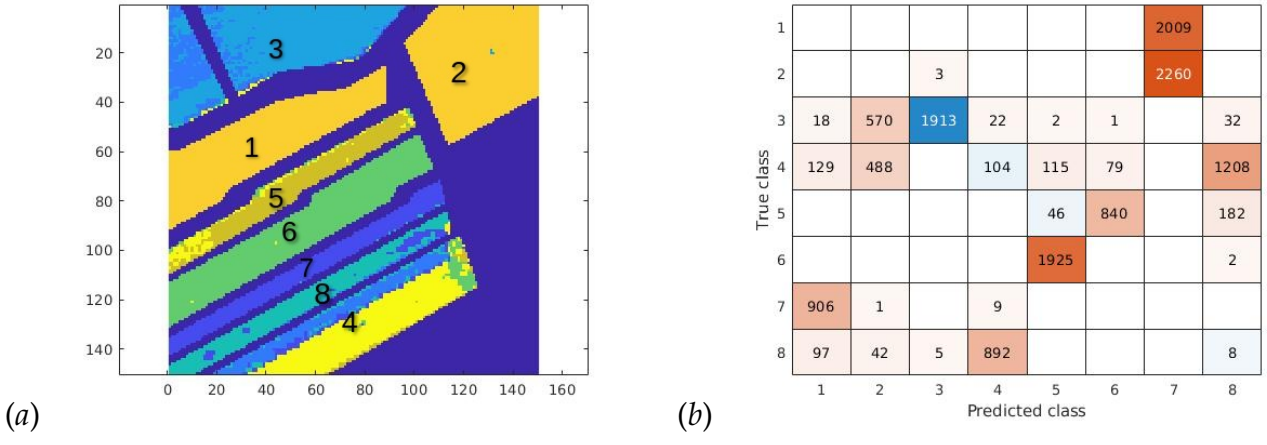


Figure 14: Distributions of datapoints in 8, 9 and 10 clusters and their confusion matrices, fuzzy $q = 1.1$

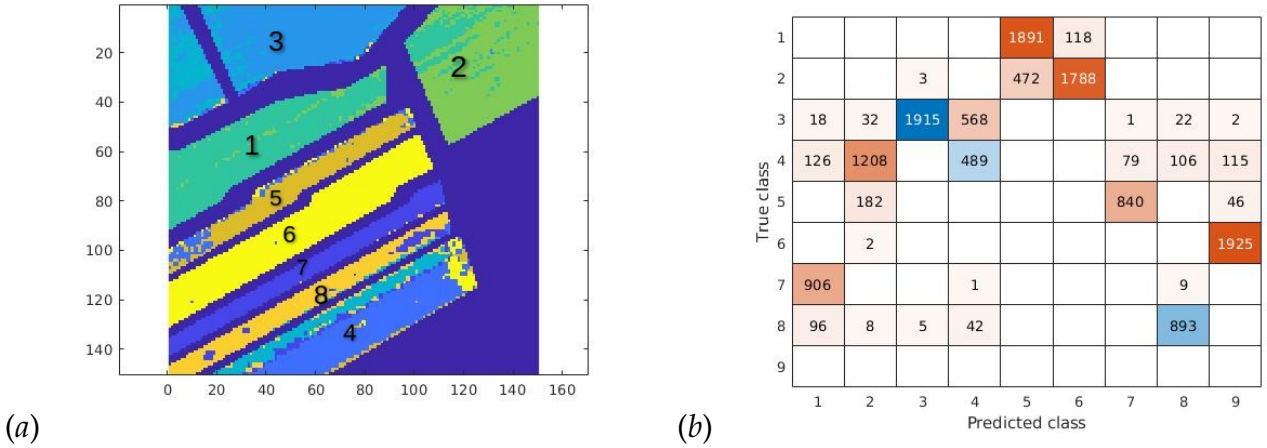
The first 2 images correspond to the case where fuzzy is executed for 8 clusters. This clustering has *accuracy* $= 73.2\%$. In this case class 1 and class 2 are not grouped together, but a new cluster has been formed that contains datapoints from class 1 and class 2. Also as we can see from the confusion matrix class 4 seems again to be problematic as it is distributed to many clusters. The case of 9 clusters is quite similar to the previous case, and the only difference seems to be the fact that the one half of class

4 now creates its own cluster. However the other half is again assigned to the cluster that contains class 5 so the previous problem remains. We executed fuzzy algorithm for 10 clusters with the hope that the one half of class 4 will be assigned to the 10nth cluster, so then we would be able to combine these 2 clusters, but instead of that, the algorithm divided class 3 in two clusters.

$$q = 1.5$$



(*a*) (*b*)

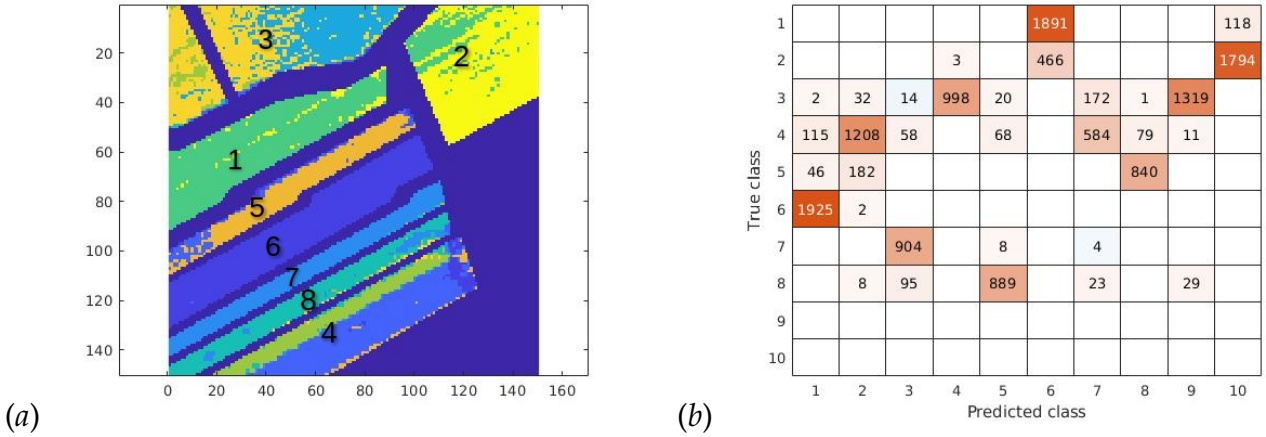


(*a*) (*b*)

Figure 15: Distributions of datapoints in 8 and 9 clusters and their confusion matrices, fuzzy $q = 1.5$

In case of 8 clusters the problem that we encountered while $q = 1.5$ with class 4 seems to be resolved as now the two halves belong to different clusters, but we could combine them together and get the real class. However a problem occurs, as class 1 and 2 are assigned in one cluster so the in this case $accuracy = 71.5\%$. To deal with this problem we executed the fuzzy algorithm with 9 clusters. In deed, class 1 and 2 are now contained in 2 clusters, so if we consider the clusters that contain the two halves of class 4 as one cluster we get $accuracy = 85.23\%$. It is slightly worst than the best case of k-means but it's definitely better that $q = 1.1$

10 Clusters - Best Fuzzy case



(*a*) (*b*)

Figure 16: Distributions of datapoints in 10 clusters and the confusion matrix, fuzzy $q = 1.5$

Case (a) is the map of clusters that are created from fuzzy for 10 clusters and the numbers correspond to the real classes as they appear in *Figure 1*. *Case (b)* is the confusion matrix of real labels and the predicted labels. This case gives the best results compared to the other cases of q as $accuracy = 88.81\%$. This accuracy is even better that the best case of k-means. The success of fuzzy is due to the fact that it can find the actual class of nearby points better compared to k-meas and that is its usefulness, as the points belong to all clusters just to a different degree.

$$q = 2$$

This case of 8 clusters is worst that the previous cases as not only class 4 in not identified as a unique cluster, but also class 3 is divided into 2 clusters. In this case $accuracy = 75.72$. In case of 9 clusters the we don't see a lot of improvement, as now class 1 and 2 are assigned in one cluster. Probably $q = 2$ is not a suitable value for this problem so we will search for $q < 2$ in order to find the best clustering algorithm.

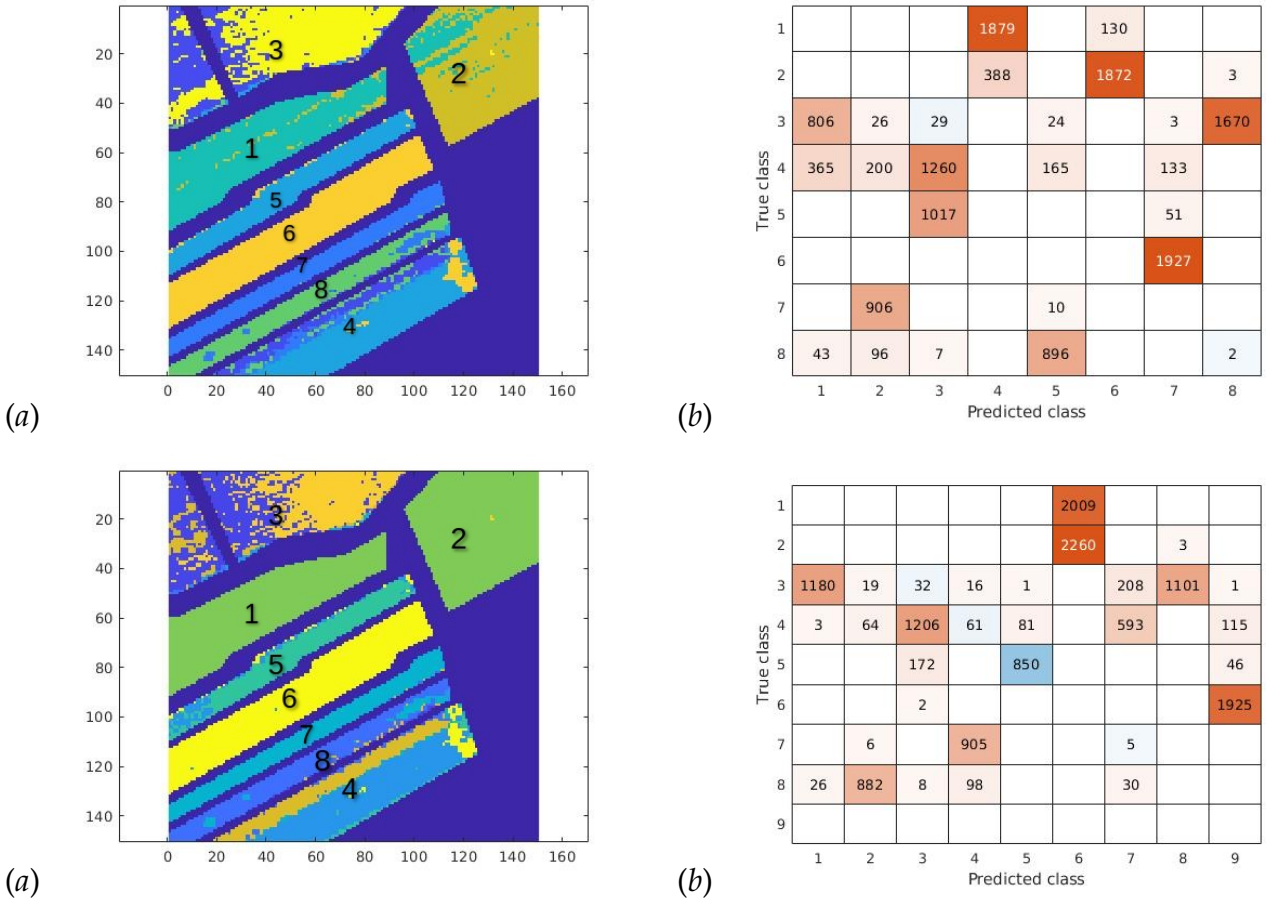


Figure 17: Distributions of datapoints in 8 and 9 clusters and their confusion matrices, fuzzy $q = 2$

Probabilistic

This clustering algorithm adopt a parametric mixture of distributions, in our case Gaussian mixture, and each one of then correspond to a cluster. So for every cluster it will be created a gaussian pdf. The only prerequisite knowledge that we need to have in order to execute this algorithm is the number of clusters. But first in order to comprehend the obtained results we will plot the classes with different colors in 2 dimensions by using only the first 2 pcs that pca analysis gives.

The numbers correspond to the real classes as they appear in *Figure 1*. Number 4 appears in two times because as we can see from the plot the yellow class is divided in two.

The first 2 images correspond to the case where the algorithm is executed for 8 clusters. This clustering has *accuracy* $= 64.4\%$. In this case class 1 and class 2 are grouped together and class 3 is divided in two. Also the one half of class 4 belongs to the cluster that contains class 5. The fact that class 1 and class 2 are group together is something that we were expecting because as we can see from the 2 dimensional representation these classes are really close. Moreover, almost all the datapoints of class 5 coexist with a part of the datapoints of class 4, and probably this the reason that half of the 4 class belongs to the same cluster with class 5. Also class 3 is quite sparse and this may cause the fact that a new cluster is created to represent its scattered and remote points. By executing the probabilistic clustering algorithm with 9 clusters

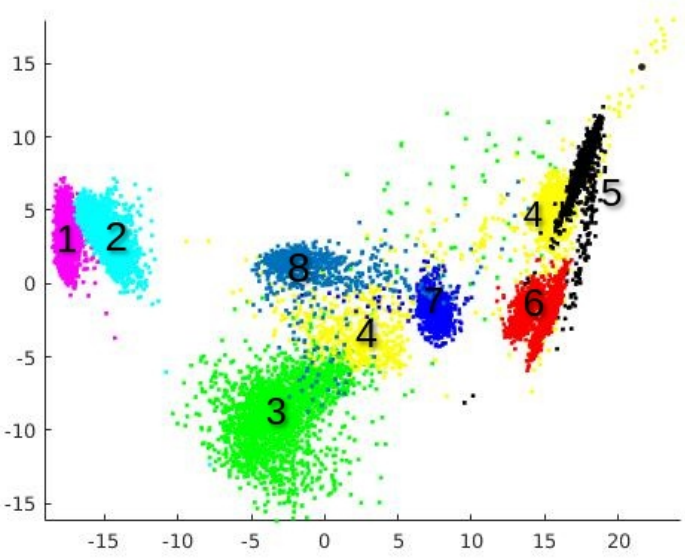
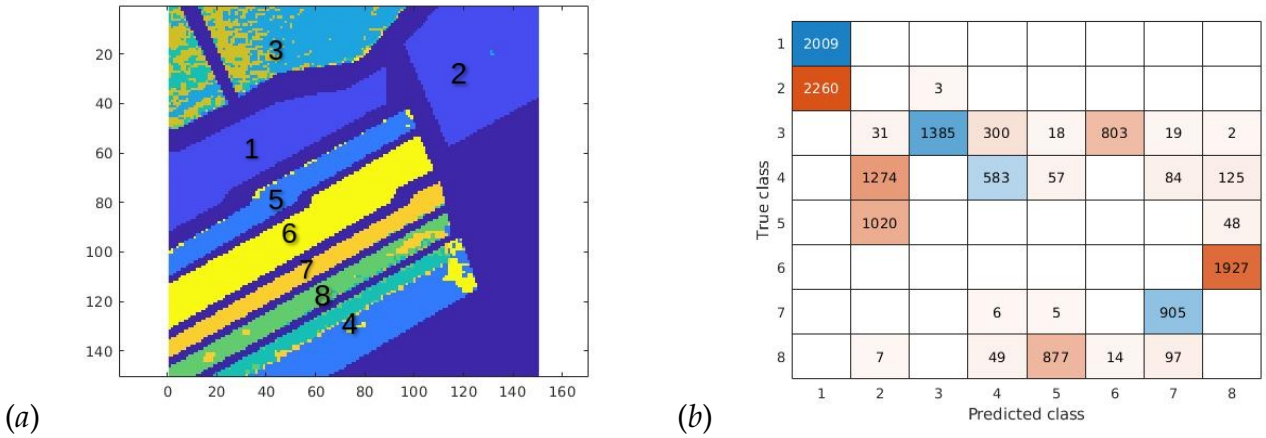


Figure 18: 2D representations of the 8 classes.



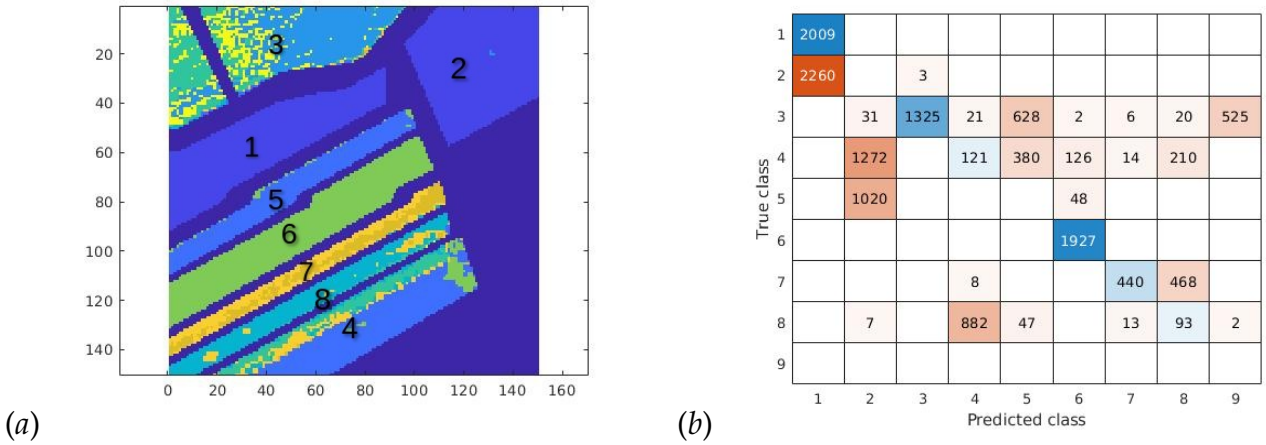


Figure 19: Distributions of datapoints in 8 and 9 clusters and their confusion matrices, probabilistic clustering

instead of 8 we get a 9th cluster that by dividing class 3 in three parts, so we are getting more far from the truth.

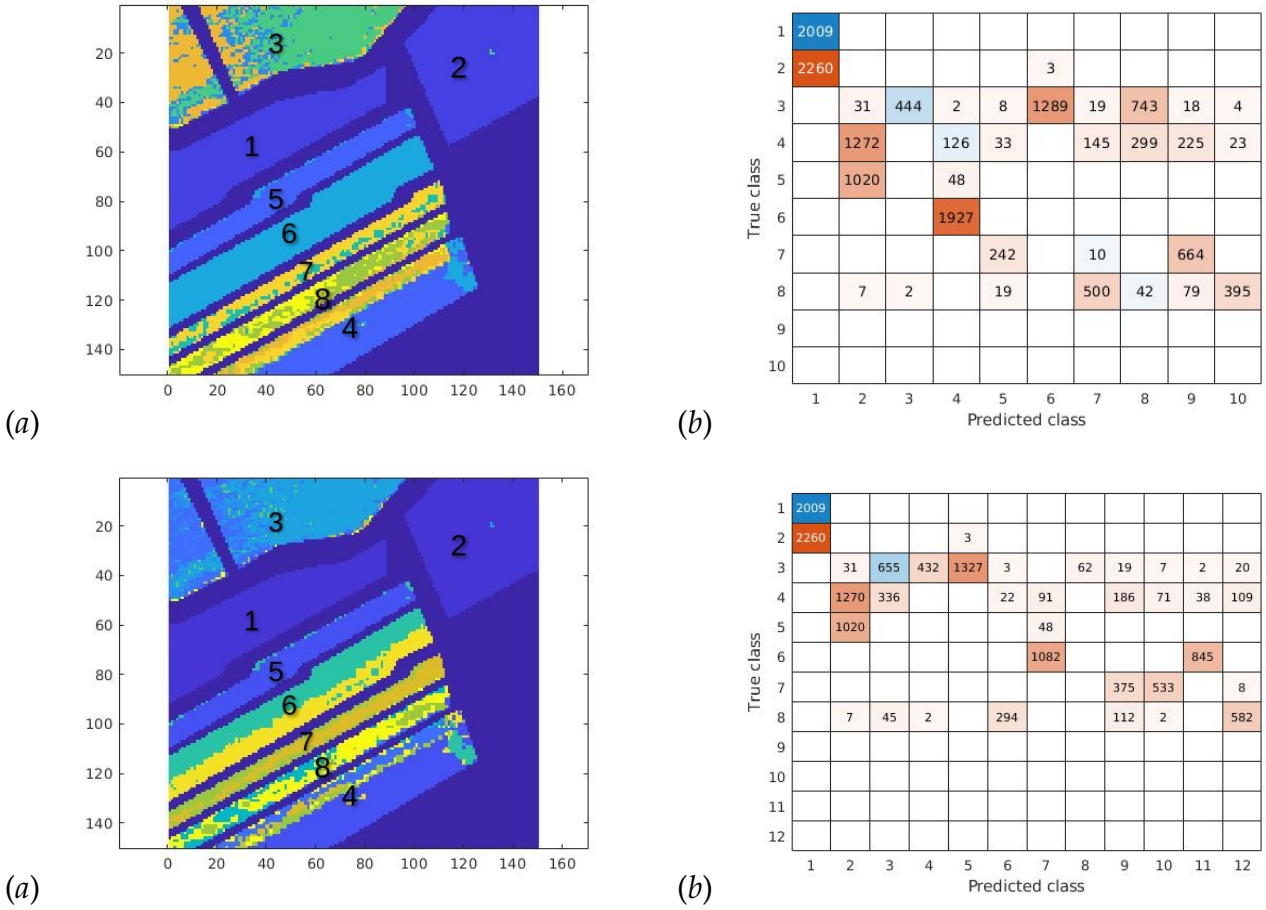


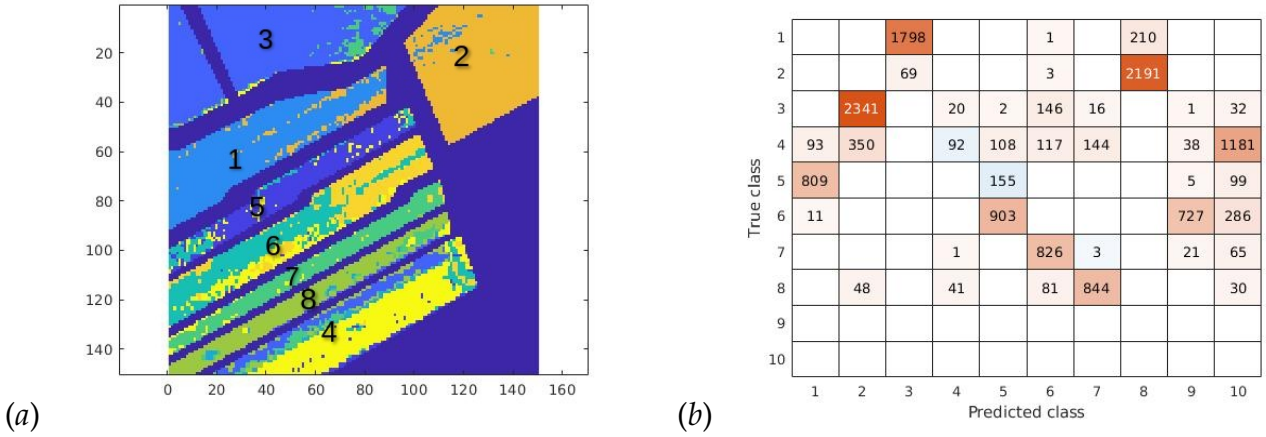
Figure 20: Distributions of datapoints in 10 and 13 clusters and their confusion matrices, probabilistic clustering

The first 2 images correspond to the case where the algorithm is executed for 10 clusters and the second two correspond to the case where the algorithm is executed for 13 clusters. We executed the algorithm with the logic that many clusters would be created which would be pieces of the classes, so then we would join these clusters and get the real classes. But this plan failed as instead of dividing classes 1 and 2 and classes 5 an 4 in separated clusters the algorithm divided the correct classes into more clusters. So we could conclude that the probabilistic algorithm is not suitable for this dataset.

Possibilistic

In order to run the possibilistic clustering algorithm we have to determine eta, which we accomplished by examining the variance of the clusters that were created from the fuzzy algorithm. The range of the etas that we tried was 0.4 to 3.4.

The initialization method that was used was vectors of X that are "most distant" from each other. Also we used $q = 1.5$ because this value of q had the best results in fuzzy.



(*a*) (*b*)

Figure 21: Distributions of datapoints in 10 clusters and the confusion matrix, possibilistic $q = 1.5$

The previous algorithms but also the display of data in the two-dimensional space led us to the conclusion that it would be better to look for 9 me 10 clusters. The case that gave the best results was that of the 10 clusters. As we can see from the plot and the confusion matrix class 6 is divided in 2 clusters, which we will consider them as one when we will measuring the accuracy and also class 4 is divided in two, which also we will consider it as one. By doing these actions we get $accuracy = 84.14\%$, which is worst than the best case of fuzzy, but probably this is due to the fact of not choosing the best values for the eta. Probably possibilistic may be able to outperform with a proper tuning, but it is a time consuming and laborious process and this is the big disadvantage of this algorithm.

Agglomerative algorithms

Single Link

In single-link hierarchical clustering, we merge in each step the two clusters whose two closest members have the smallest distance or in other words the two clusters with the smallest minimum pairwise distance. We executed single-link and there is the dendrogram that occurs:

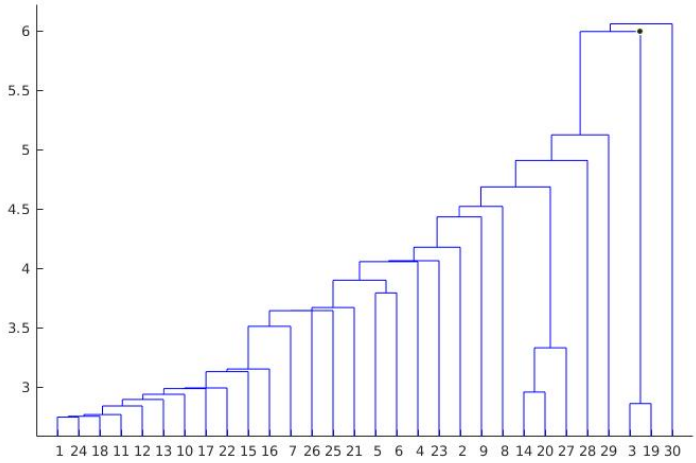


Figure 22: Dendrogram of single-link

We examine the distribution in the data when single-link has created 8 clusters and when it has created 20 clusters. Below are the distribution of data on the map and the confusion matrix for both cases.

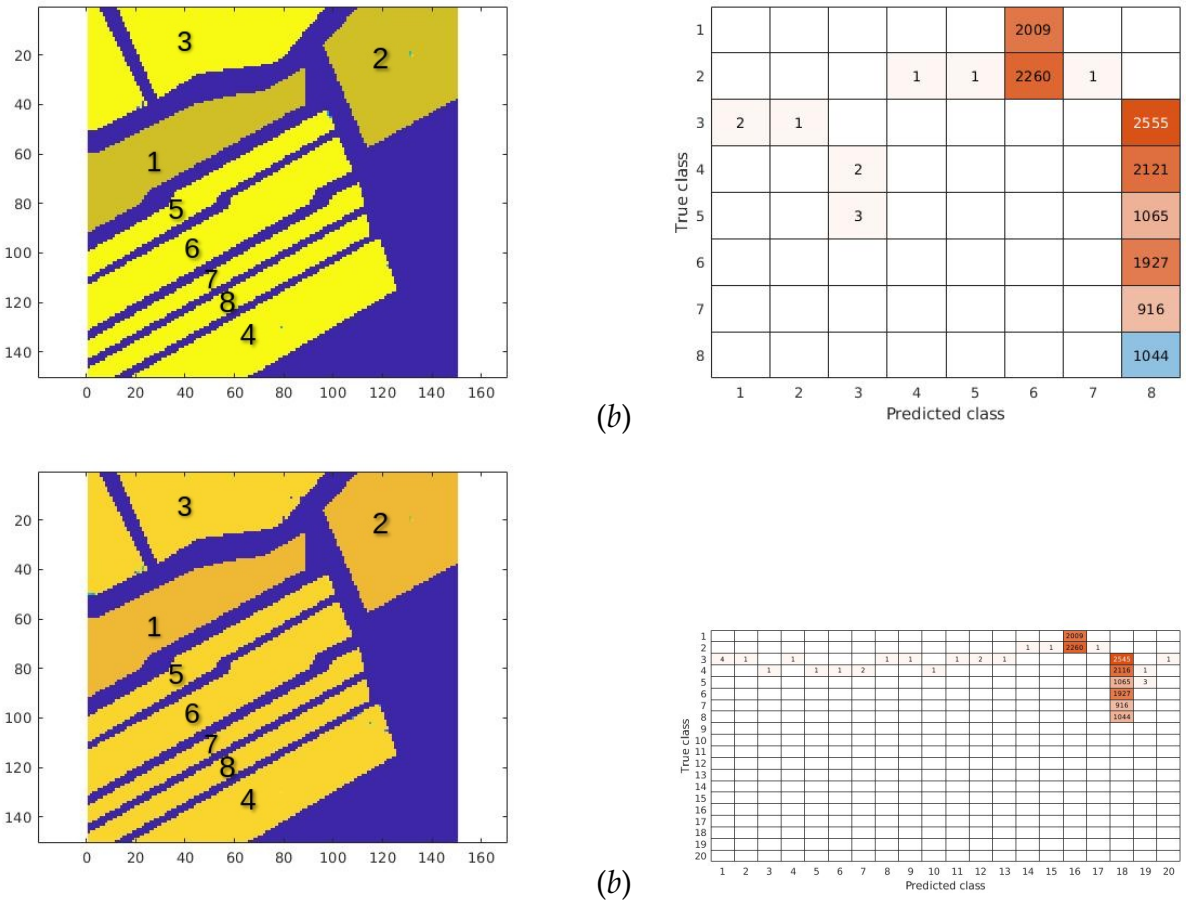


Figure 23: Distributions of datapoints in 8 and 20 clusters and their confusion matrices, single-link

As we can see from the map and the confusion matrix single link not only can't classify the datapoints correctly but it can find only two meaningful clusters, because the rest of the clusters contain too few points. The reason that we executed the algorithm for 20 cluster is to figure out if this problem of not finding more that 2 clusters goes away but apparently it can't be solved. Single-link fails completely because the classes are not elongated.

Complete Link

In complete-link hierarchical clustering, we merge in each step the two clusters whose merger has the smallest diameter or in other words the two clusters with the smallest maximum pairwise distance. We executed single-link and there is the dendrogram that occurs:

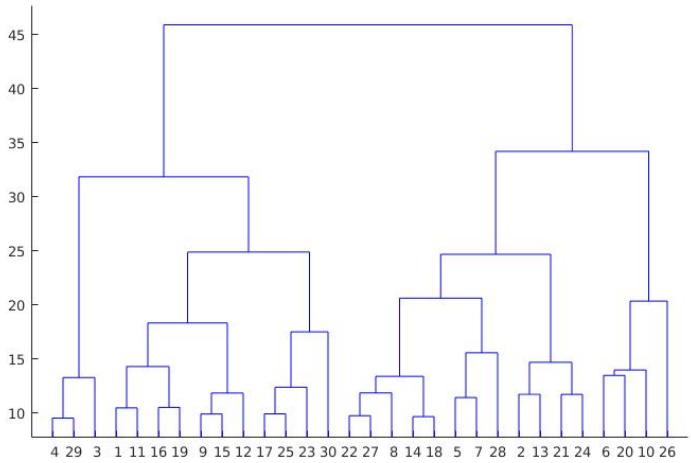


Figure 24: Dendrogram of complete-link

We examine the distribution in the data when complete-link has created 8 clusters and when it has created 12 clusters. Below are the distribution of data on the map and the confusion matrix for both cases. Complete-link is better that single-link because the classes tend to be more compact than elongated.

As we can see from the map and the confusion matrix complete link can detect most of the classes, so it's much better that single-link but it is not very accurate as $accuracy = 57.24\%$ Class 4 is once again divided in two and class 1 and 2 are assigned in the same cluster. We examined if these problems goes away when the clusters are 13 but as we can from the plot and the confusion matrix instead of creating clusters that contain class 1, class 2 and class 4 alone, the algorithm divides class 3 in at least 3 significant clusters. So complete link isn't the best clustering algorithm for our data.

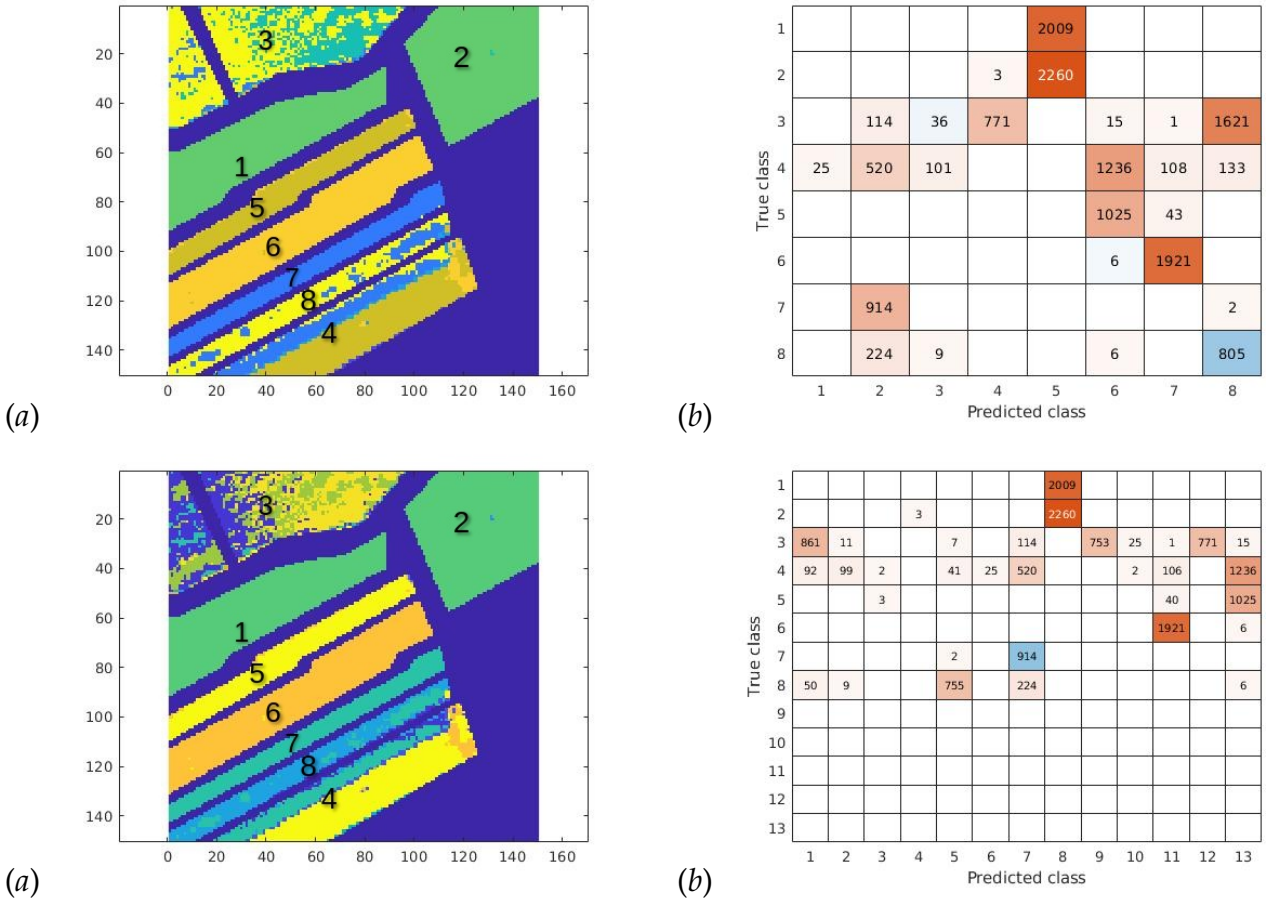


Figure 25: Distributions of datapoints in 8 and 13 clusters and their confusion matrices, complete-link

Weighted Pair Group Method with Arithmetic Mean

The WPGMA algorithm constructs a rooted tree that reflects the structure present in a pairwise distance matrix. At each step, the nearest two clusters, say i and j, are combined into a higher-level cluster $i \cup j$. Then, its distance to another cluster k is simply the arithmetic mean of the average distances between members of k and i and k and j:

$$d_{(i\cup j),k} = \frac{d_{i,k} + d_{j,k}}{2}$$

We executed WPGMA and there is the dendrogram that occurs:

We examine the distribution in the data when WPGMA has created 8 clusters and when it has created 14 clusters. Below are the distribution of data on the map and the confusion matrix for both cases.

WPGMA seems to be more accurate than complete-link as it has $accuracy = 69.69\%$. However class 4 is again divided in two and class 1 and 2 are assigned in the same cluster. This problem that we encountered in previous case to doesn't seem to be easy to solve as it pops up in almost all the clustering algorithms. We examined the 14 clusters that are given from WPGMA with the hope that the problem was eliminated but class 1 and 2 are again together and 4 separeted in two.

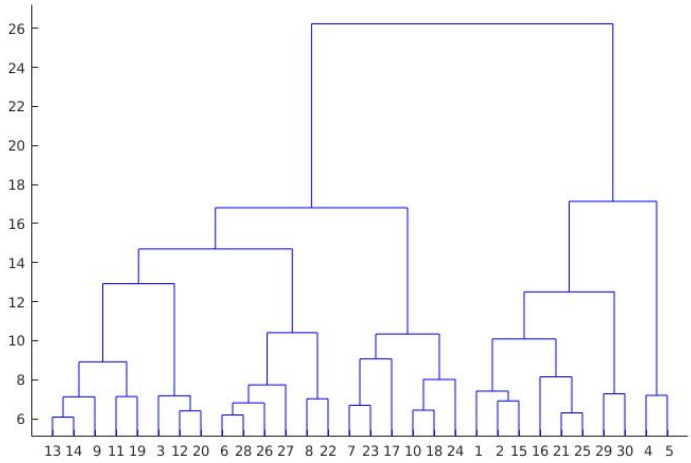
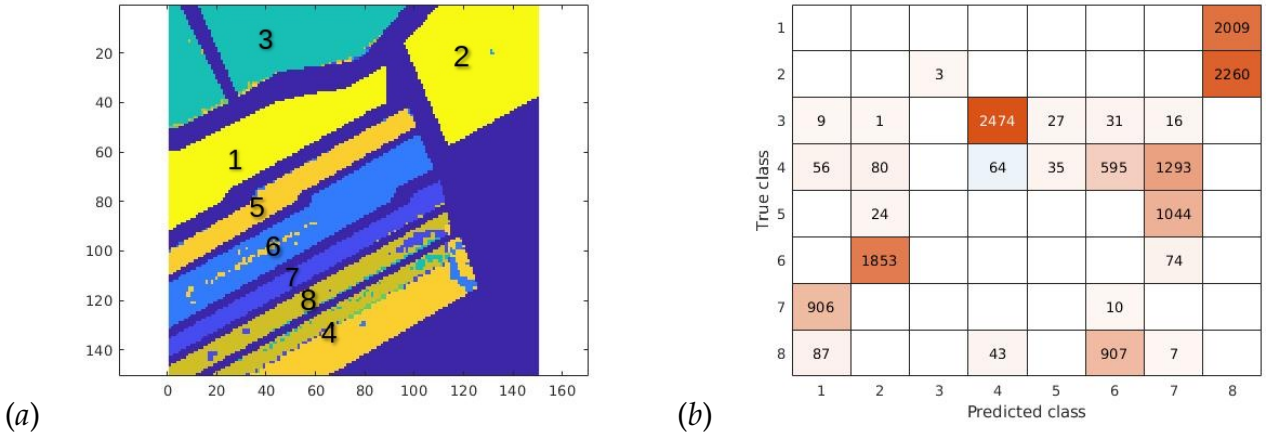


Figure 26: Dendrogram of WPGMA



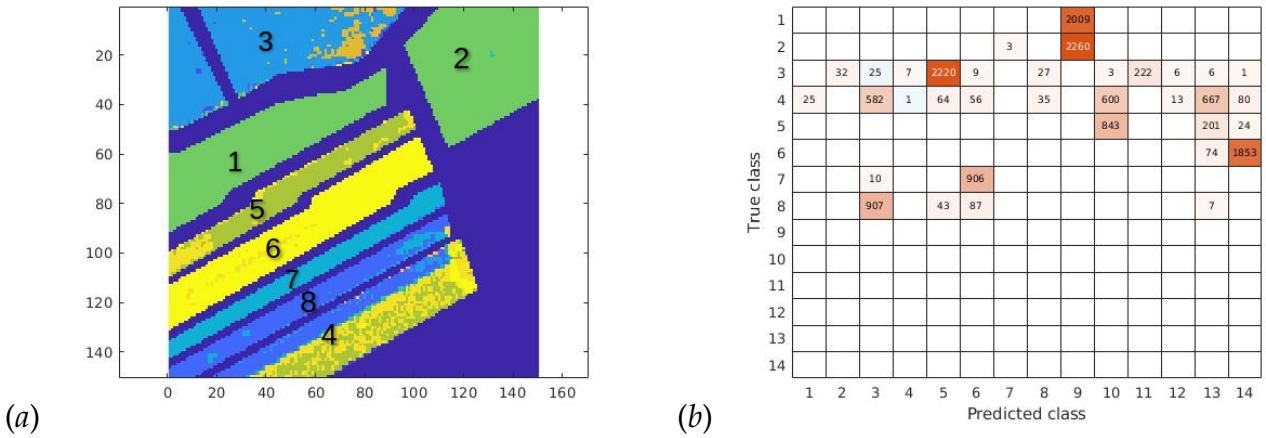


Figure 27: Distributions of datapoints in 8 and 14 clusters and their confusion matrices, WPGMA

Unweighted pair group method with arithmetic mean

The UPGMA algorithm is quite similar to the Weighted Pair Group Method with

Arithmetic Mean with the difference that:

$$d_{(i\cup j),k} = \frac{n_i}{n_i + n_j} d_{i,k} + \frac{n_j}{n_i + n_j} d_{j,k}$$

We executed UPGMA and there is the dendrogram that occurs:

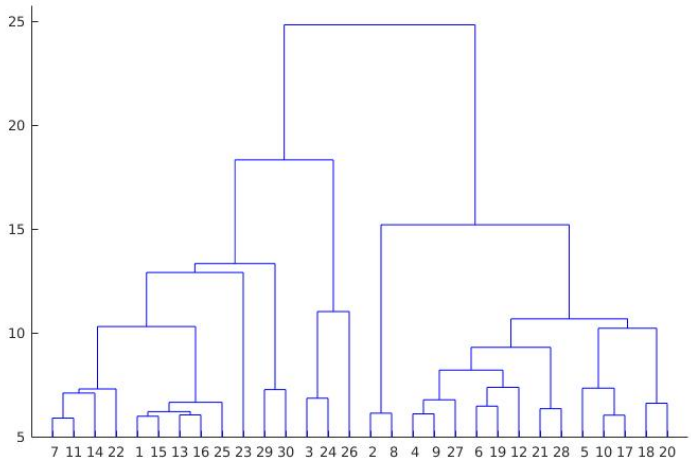


Figure 28: Dendrogram of UPGMA

We examine the distribution in the data when UPGMA has created 8 clusters and when it has created 14 clusters. Below are the distribution of data on the map and the confusion matrix for both cases.

UPGMA seems to be less accurate than WPGMA as it has $accuracy = 57.13\%$. Despite the fact the clusters that are created in both cases (for 8 and 23 clusters) are well defined and doesn't seem to have noise the final clusters the occur are not correct.

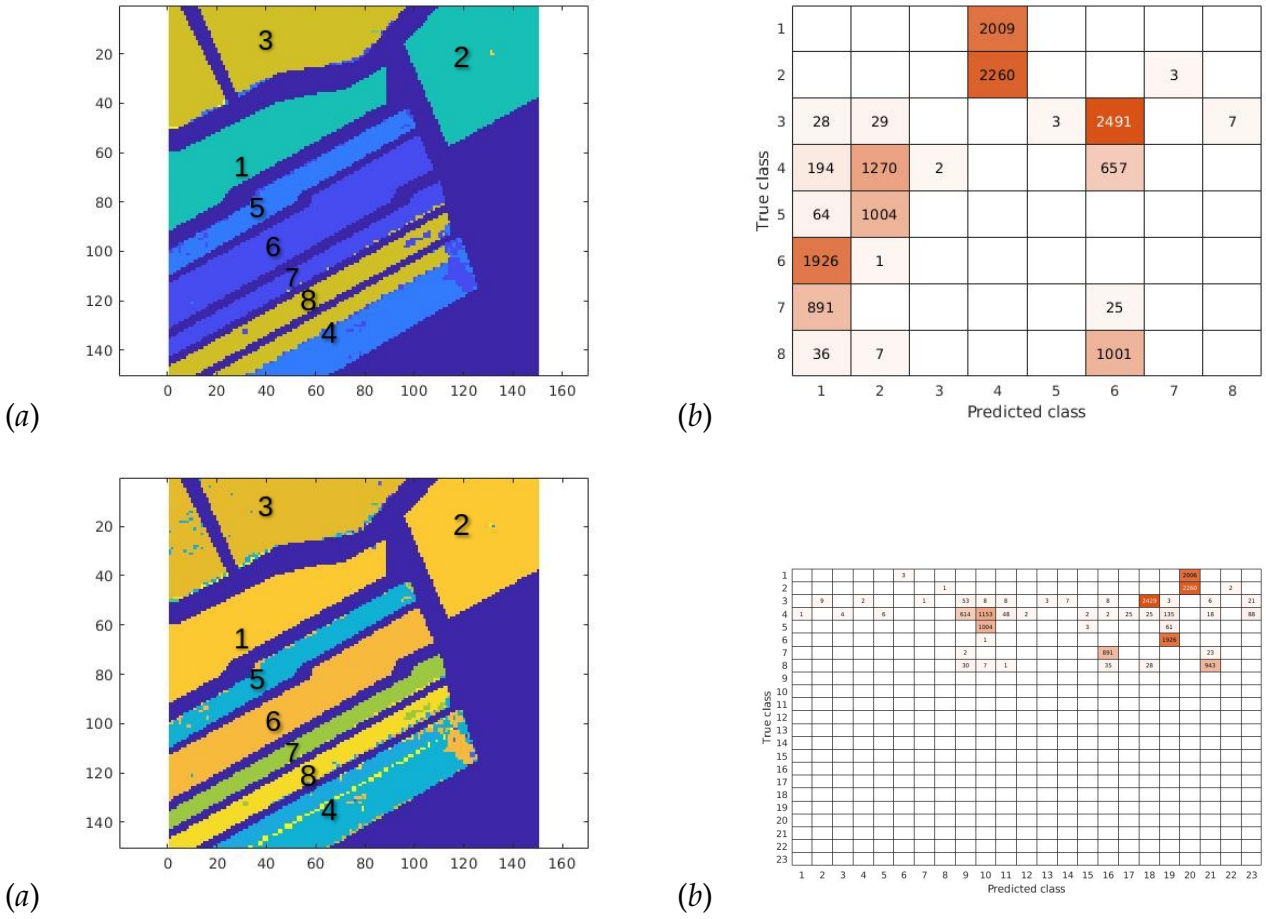


Figure 29: Distributions of datapoints in 8 and 23 clusters and their confusion matrices, UPGMA

Weighted pair group method centroid

The distance between two clusters is defined as the distance between the centroid for cluster 1 and the centroid for cluster 2. We executed WPGMC and there is the dendrogram that occurs:

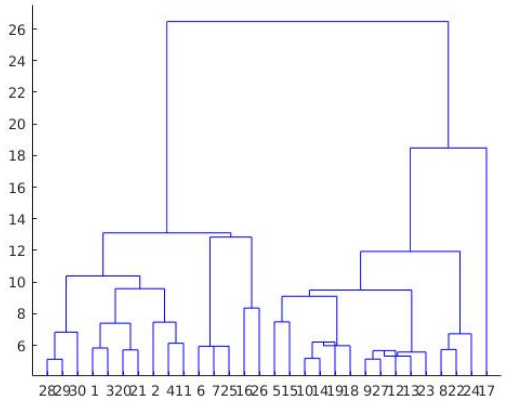


Figure 30: Dendrogram of WPGMC

We examine the distribution in the data when Ward has created 8 clusters and when

it has created 12 clusters. Below are the distribution of data on the map and the confusion matrix for both cases.

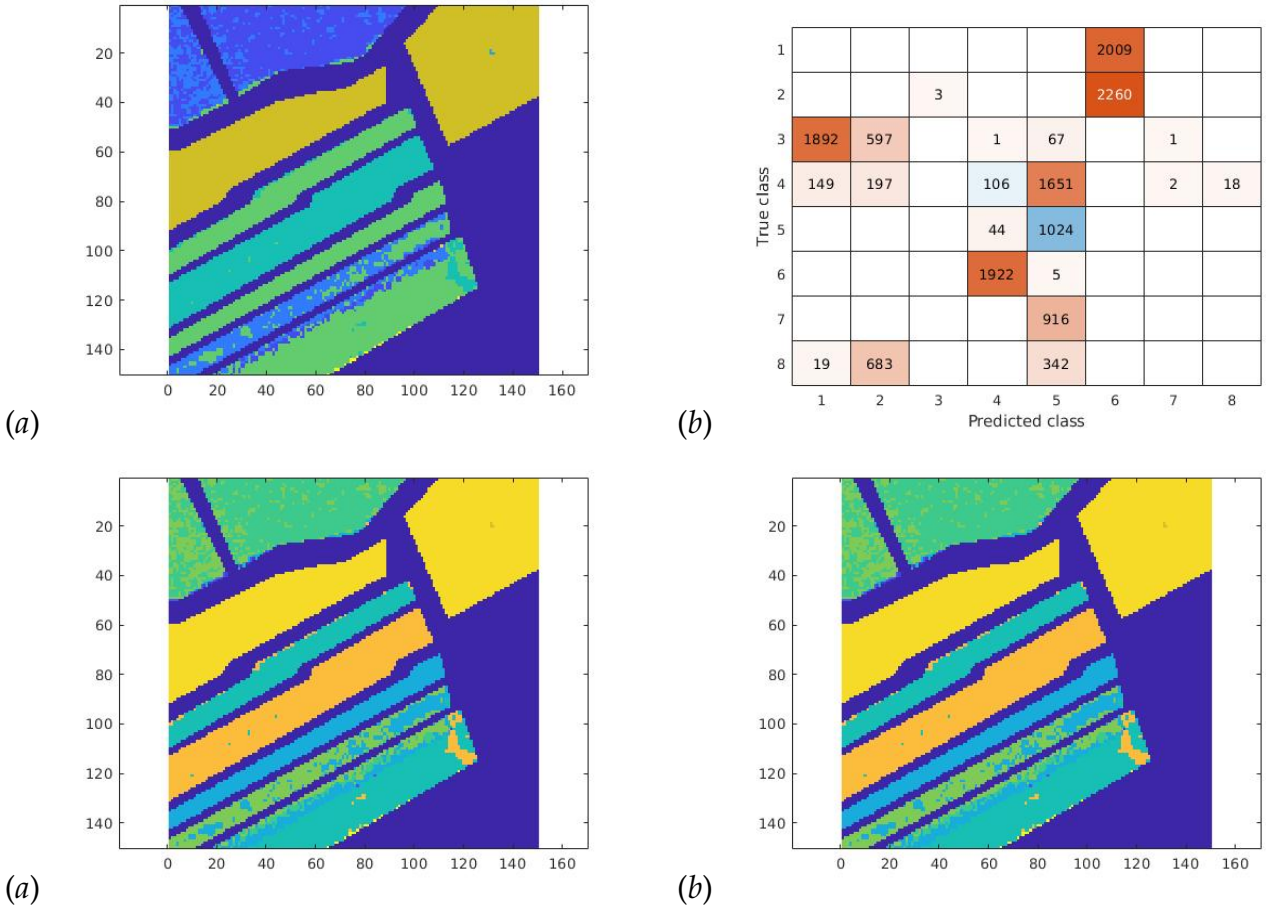


Figure 31: Distributions of datapoints in 8 and 12 clusters and their confusion matrices, WPGMC

In this case we get $accuracy = 60.45\%$ for the 8 clusters, which is better from the previous case. However the clustering is not accurate neither in the case of 8 clusters nor in the case of 12.

Ward or minimum variance algorithm agglomerative

Like other clustering methods, Ward's method starts with n clusters, each containing a single object. These n clusters are combined to make one cluster containing all objects and at each step, the process makes a new cluster that minimizes variance. We executed Ward and there is the dendrogram that occurs:

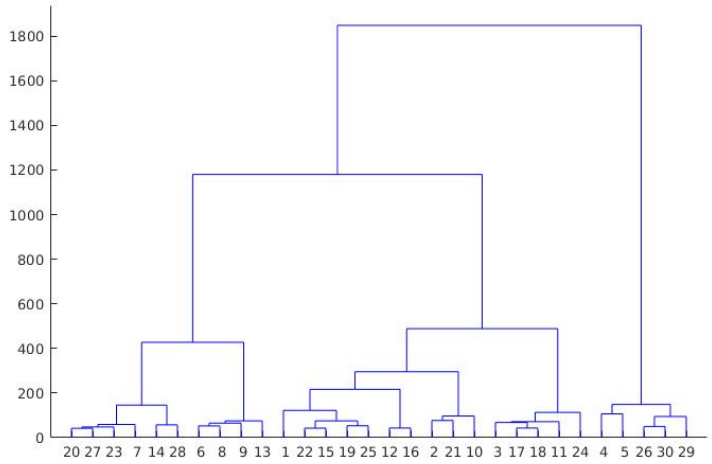


Figure 32: Dendrogram of Ward

We examine the distribution in the data when Ward has created 8 clusters and when it has created 9 clusters. Below are the distribution of data on the map and the confusion matrix for both cases.

Ward seems to be less the most accurate of all agglomerative algorithms as it has $accuracy = 82.29\%$. In case of 8 clusters almost all clusters correspond to one true class except from class 4 which is once again divided. However, when that clusters are 9 we could combine the two clusters that contain the two halves of class 4, so in the end we will have 8 clusters, which will be quite close to the true classes. If we implement this method then we get $accuracy = 89.21$ which is much higher that the previous case.

10 Clusters - Best case

In this case class 3 and class 4 are divided into two clusters each but if we consider them in both cases as one then we will have 8 final clusters and the $accuracy = 92.96\%$, which is the best accuracy achieved over all the clustering algorithms that we tried. Ward's method approach also does well in separating clusters if there is noise between clusters and maybe this led to such a good accuracy. The fact that Ward's method is the best proves that the clusters are globular, which was also proved by the complete failure of single-link.

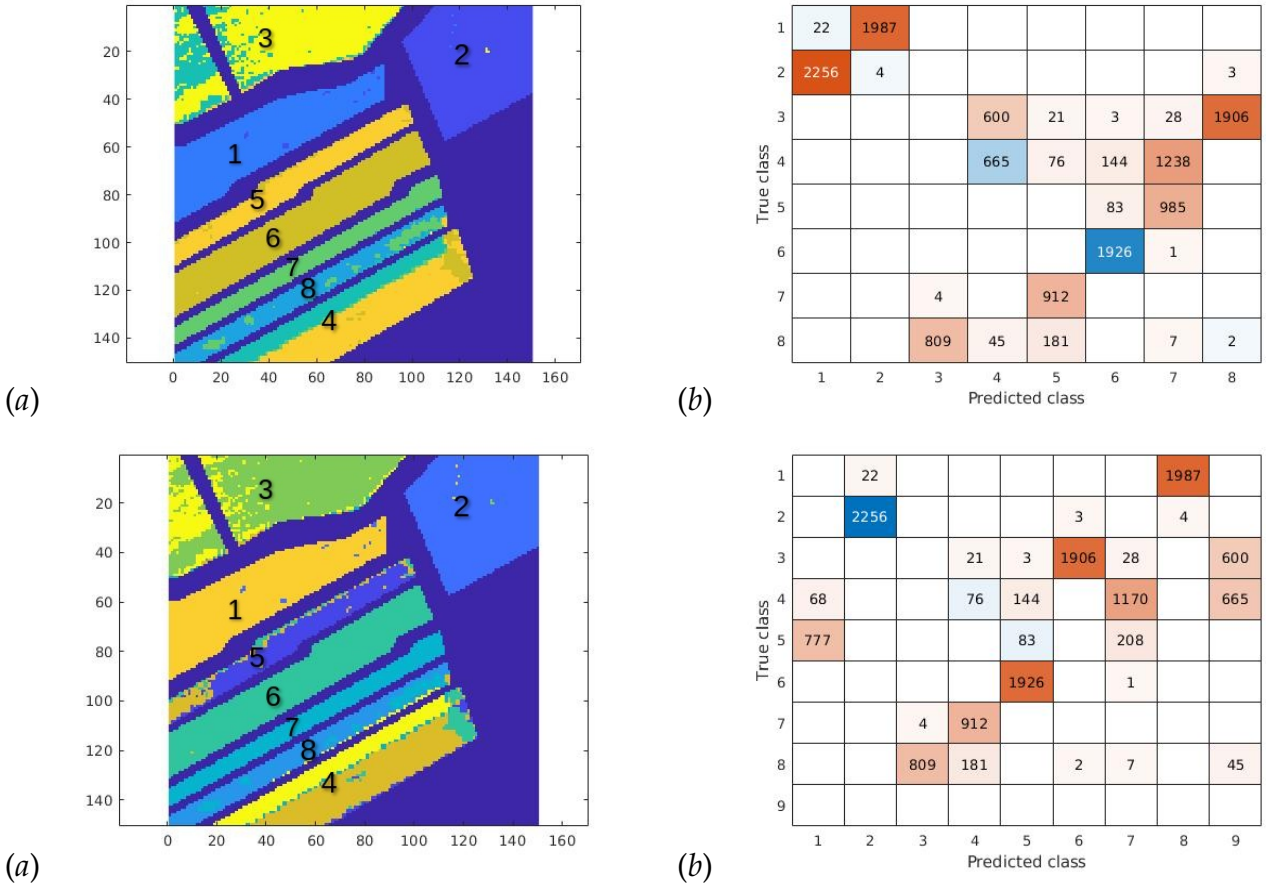


Figure 33: Distributions of datapoints in 8 and 9 clusters and their confusion matrices, Ward

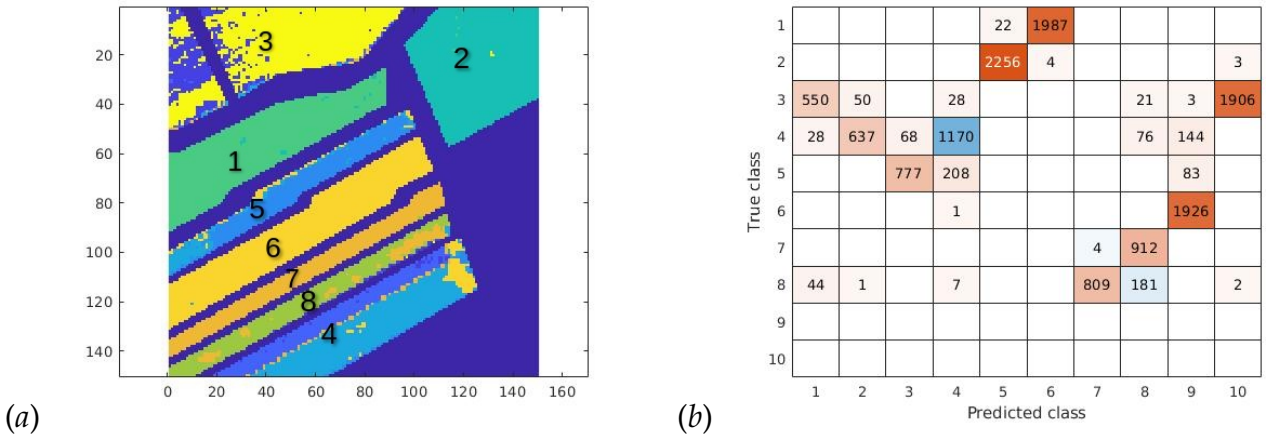


Figure 34: Distributions of datapoints in 10 clusters and the confusion matrix, Ward

Conclusions

Hierarchical clustering may have some benefits over k-means such as not having to pre-specify the number of clusters and the fact that it can produce a nice hierarchical illustration of the clusters. However, from a practical perspective, hierarchical clustering analysis still involves a number of decisions that can have large impacts on the interpretation of the results. First, you still need to make a decision on linkage method.

Each linkage method has different systematic tendencies in the way it groups observations and can result in significantly different results. In our case Ward's method proved to be the best linkage method as our classes are made from roughly the same number of observations and there are not outliers. Also, although we do not need to pre-specify the number of clusters, we often still need to decide where to cut the dendrogram in order to obtain the final clusters to use.

In conclusion the best agglomerative algorithm appears to be the Ward algorithm as it outperforms the others. However in general the CFO algorithms gave good results too, as we saw in case of fuzzy with 10 clusters and k-means with 9 clusters. Probably possibilistic is able to give even better results if we perform a most careful tuning. It is worth noting that the CFO algorithms were more noisy. Finally, the CFO algorithms gave more mediocre results in contrast to the agglomerative ones which gave quite bad results, but in the end both categories of algorithms managed to solve the problem efficiently.