Lab 9

Julia Janicki

Introduction

My goal for this project is to map a general land cover in the area of Alexandria in Egypt using supervised classification, specifically the Maximum Likelihood and Support Vector Machine algorithms.

In supervised classification, the identity and location of some of the land cover types are known beforehand and sites are labeled as specific classes to create a training dataset. The sites are evaluated for separability and a supervised classification algorithm is applied where programs can determine pattern of training class and each pixel in full dataset is compared to training patterns and labeled on similarity.

The acquisition date of this image is 2003/08/05, the path/row of the footprint is 2042/2218, the resolution of the image is 30 m and the sensor is unknown.

Method

Supervised classification

For this project, supervised classification algorithms were used to classify land cover types for the image. In supervised classification, the identity and location of some of the land cover types are known beforehand and the analyst will label sites that represent homogeneous areas of the different land cover types to create a training dataset. The spectral areas of the training dataset are used to train the classification algorithm for color mapping of the rest of the image by comparing similarity between individual pixels and the training set. There are many supervised classification algorithms the analyst can choose from.

Supervised classification can be much more accurate than unsupervised classification, but this depends greatly on the training dataset, the skill and experience of the analyst processing the image, and the spectral distinctness of the classes. Thus if the training data is not good and classes have overlapping spectral characters, the resulting image might not be ideal. Supervised classification requires more attention while creating the training data therefore it generally requires more times and money compared to unsupervised classification.

Maximum Likelihood Algorithm

Maximum likelihood is a supervised classification algorithm that assumes the class samples are normally distributed, it evaluates the mean, variance and covariance of the training data then computes the probability for each unknown pixel belonging to a particular class. A pixel with the maximum likelihood or highest probability is classified into the corresponding class.

Advantages of Maximum Likelihood include that it is the most accurate of the classifiers if the data has a normal distribution. Disadvantages include when the distribution of the population does not follow the normal distribution the maximum likelihood does not work well, it is computationally intense, and it tends to overclassify signatures with relatively large values in the covariance matrix.

Support Vector Machine Algorithm

The SVM algorithm is derived from statistical learning theory and produces an optimal hyperplane by determining the location of decision boundaries that produce the optimal separation between classes. The optimal hyperplane maximizes the distance between itself and the planes representing the two classes. The support vectors are the data points that lie at the edge of each individual class hyperplane and are closest to the optimal hyperplane. SVMs can handle non-linear boundaries between classes by using kernel functions.

Advantages of SVM include that it often produces higher classification accuracy than the traditional statistical methods and it requires a small training dataset. In particular, it is often more accurate compared to Maximum Likelihood, since data acquired from remotely sensed imagery often have unknown distributions and methods such as Maximum Likelihood assume a normal distribution. Disadvantages of SVM include that it has extensive memory requirements and if the training dataset is too large it takes a long time to run, and it is are very sensitive to the choice of the kernel parameters.

Potential Issues

For supervised classification, if the training data is not good and classes have overlapping spectral characters, the resulting image will not be great. This can be addressed by creating a better training dataset by drawing more precise polygons to ensure there are no mixed pixels and drawing more polygons.

For the Maximum Likelihood algorithm, if the statistics for each class in each band are not normally distributed then the resulting image may also not be very good. This can be addressed by using a different supervised classification algorithm that is non-parametric.

For SVM, potential issues that may come up include that the algorithm may take a long time to run if there are many classes and a larger training dataset, and different kernel functions may produce very different results. This can be addressed by keeping the number of classes low and drawing more precise region of interests.

Classification scheme and Training Sites

The satellite image was first analyzed in order to determine which land cover class types to use to classifying the image, and some adjustments were made along the way. The final land cover class types used are: deep water, shallow water, contaminated water, very contaminated red water, desert, urban area and roads, dry bare field, wet bare field, dark brown field, sparsely vegetated field, moderately vegetated field, and densely vegetated field.

After having decided the initial set of land cover type classes to use, in ENVI I created many Region of Interests for each land cover class type by drawing polygons. For each land cover type I created from five to 20 polygons depending on how complicated and how much spectral variation the land cover type is.

For the first map I ran the Maximum Likelihood algorithm on the image 18 times, each time altering the polygon number or shapes, adding or deleting classes as well as changing the probability threshold until I produced a decent image.

For the second map I ran the SVM algorithm on the image and I used the same ROIs used to produce the final map from the Maximum Likelihood algorithm.

For the Maximum Likelihood algorithm, creating more polygons for certain classes, redrawing some of the ROIs to make them more specific, making sure that each aggregate class have spectral subclasses, and adding or deleting land cover class types were all something that worked for me in certain cases.

An example that worked after adding more specific land cover type classes is water. Since water has a lot of variation in its spectral character as there are contaminated water and vegetated water, on top of drawing more polygons for the water class I also added more specific land cover type classes for water.

An example that worked after reducing land cover type classes is merging dense urban, sparse urban and roads into urban. In the beginning instead of having a general urban + road class, I had three separate classes: dense urban, sparse urban, and road. However, since the algorithm cannot distinguish the three specific classes I ended up merging them into one urban land cover class type.

Redrawing ROIs and making them more specific to avoid having mixed pixels in the class also worked. For example, instead of having a big ROI that included urban and a narrow river that crossed the urban area, after I refined my ROIs and made sure the urban class only included pixels from urban areas and not the river, the output turned out a lot better.

For the SVM algorithm, refining the polygons worked, removing some of the water ROIs also worked, but creating an extra land cover type class for "vegetated water" did not help.

There were many water ROIs that included areas that had a lot of vegetation and mud, when the SVM algorithm ran many of the dense vegetation areas were classified as water. To resolve this problem I deleted these ROIs so the water class only includes areas that are purely water, which worked really well. But some of the vegetated water were still classified as water so I created an extra land cover type class for "vegetated water", but this made it worse as a lot of the dense vegetation were classified as vegetated water instead, so I ended up not using this class.

Accuracy assessment

Having produced the final maps, it is essential to conduct an accuracy assessment in order to understand how close the maps are to reality (map accuracy), the deviation of maps from known locations (spatial accuracy) and the level of detail that is mapped (thematic precision). We need to consider accuracy assessments since there are many sources of errors throughout the whole remote sensing process, from data acquisition to classifying images. In this particular case, I created an error matrix in order to obtain values including the total accuracy, producer's accuracy for each land cover class type, and consumer's accuracy for each land cover class type.

The first step is to select a sample design for creating testing sites. In general, random points are better than non-random points since it would be governed by chance and each point has an equal likelihood of being represented. It is also better to choose a sample of certain number of points instead of comparing all the points, otherwise it will take too long. The next step is to obtain ground reference data. Finally the last step is to produce the error matrix.

For this assessment, a random sample design was chosen and around 50 pixels were generated randomly from a noise image. These pixels were then overlaid on top of the satellite image and labels were given to them based on their similarities to particular land cover type classes in order to produce the ground reference data. An error matrix was then produced by comparing the randomly generated "ground truth" test pixels to the land cover type classes produced in the final images by using the "Confusion Matrix Using Ground Truth ROIs" function in ENVI. This was done for both the map classified using the Maximum Likelihood algorithm and the map classified using the SVM algorithm.

The error matrix produces the total accuracy and the kappa coefficient for the overall image, as well as user's and producer's accuracies for each land cover type class. The diagonal elements of the error matrix represent areas that were correctly classified. The off-diagonal elements of an error matrix represent the areas that were not correctly classified, in other words there was omission from the correct category or commission to the wrong category. The outer row and column totals are used to computer producer's and user's accuracy.

The total accuracy is produced by dividing the total number of pixels that were correctly identified (diagonal total) by the total number of pixels, and it gives a sense of how well the overall image is classified. The kappa coefficient reflects the difference between actual agreement and the agreement expected by chance.

User's accuracy is associated with the error of commision, and it is when an area is included in a category to which it does not truly belong. Producer's accuracy associated with the error of omission, and it is when an area is excluded in a category to which it does truly belong. In other words, a user would find that X% of the time an area visited on the ground that the classified map calls class A will actually be class A. Producer can claim that Y% of the time that an area is class A was identified as such.

Results

Final Maps

The result land cover classification of Alexandria, Egypt can be seen in **Figure 1.** for the Maximum Likelihood algorithm and **Figure 2.** for the Support Vector Machine algorithm. At the end there were 12 land cover type classes: deep water, shallow water, contaminated water, very contaminated red water, desert, urban area and roads, dry bare field, wet bare field, dark brown field, sparsely vegetated field, moderately vegetated field, and densely vegetation.

Overall, the SVM image looks a little better than the ML image. The desert, dry bare field, and the urban areas are better classified, in particular there is an area towards the right part of the image in the middle (**Figures 3** and **4**) where urban, dry brown field and desert are really well classified using SVM but not very well classified using ML since many desert areas are misclassified as urban. Moreover, the SVM classification is cleaner in the sense there are fewer single pixel areas in the SVM image, in particular there is a better delineation between cropland and desert (**Figures 5-8**). Also, the roads are also cleaner in the SVM image, overall they have a long narrow and crisper shape while in the ML image they often are wider and seem to include some of the pixels that should have been desert (**Figures 9 and 10**).

However, the ML algorithm produced a better result for the canals, which is classified as either shallow or deep water depending on the algorithm, for the image. In the SVM image, only the bigger canals in the central portion of the image were successfully delineated, while in the rest of the image where the canals are smaller they are often not correctly identified and often just blended in as roads, dense vegetation or brown fields (**Figures 11-14**). On the other hand, SVM did a better job at delineating the dried out lake, while in the ML image the borders of the lake is not so visible (**Figures 15 and 16**).

Finally, where both images had problems were for the vegetated water (**Figures 17 and 18**). There is an area towards the center top of the image where there are many big pools of water, potentially lakes, that have a lot of vegetation and mud. Accordingly, both classification algorithms misclassified parts of the lake as dense vegetation or dark brown field.

Overall Accuracy

The error matrix for the map resulting from the ML algorithm can be seen in **Table 1**, while the error matrix for the map resulting from the SVM algorithm can be seen in **Table 2**. The overall accuracy for the ML image is 62.5% with the kappa coefficient being 0.5591, and the overall accuracy for the SVM image is 63.79% with the kappa coefficient being 0.5804.

User's Accuracy

For the ML image, the classes with high user's accuracy include deep water and desert, moderate user's accuracy include urban, dense vegetation, moderately vegetated field, sparsely vegetated field and dry brown field, and low user's accuracy include wet brown field and dark brown field.

Deep water has a 100% user's accuracy, which means that users would find that all of the time (100% of the time) when they visit the area in this image that what the classified map calls deep water will actually be deep water. Desert has a 91.67% user's accuracy, which means that users would find that most of the time (91.67% of the time) when they visit the area in this image that what the classified map calls deep water will actually be deep water.

Urban, dense vegetation, moderately vegetated field, sparsely vegetated field and dry brown field all have moderately high user's accuracy, between 50 and 66.67%, which means that users would find that between quite often (50 and 66.67% of the time) when they visit the area in this image that what the classified map calls a specific class out of these classes will actually be that class.

Finally, wet brown field and dark brown field both have a 0% user's accuracy, which means that users would find that never (0% of the time) when they visit the area in this image that what the classified map calls wet brown field or dark brown field will actually be wet brown field or dark brown field.

For the SVM image, the classes with high user's accuracy include deep water and desert, moderate user's accuracy include urban, dense vegetation, moderately vegetated field, sparsely vegetated field and dry brown field, and low user's accuracy include wet brown field and dark brown field.

Deep water has a 100% user's accuracy, which means that users would find that all of the time (100% of the time) when they visit the area in this image that what the classified map calls deep water will actually be deep water.   Desert has a 84.62% user's accuracy, which means that users would find that very often (84.62% of the time) when they visit the area in this image that what the classified map calls deep water will actually be deep water. The one pixel that should have been urban and one pixel that should have been sparsely vegetated field was misclassified as desert.

Urban, dense vegetation, moderately vegetated field, sparsely vegetated field and dry brown field all have moderately high user's accuracy, between 50 and 71.43%, which means that users would find that often (between 50 and 71.43% of the time) when they visit the area in this image that what the classified map calls a specific class out of these classes will actually be that class. The urban class is most often confused with sparsely vegetated field, as the two pixels that should have been sparsely vegetated field was misclassified as urban.

Finally, dark brown field has a 33.33% user's accuracy, which means that users would find that not very often (33.33% of the time) when they visit the area in this image that what the classified map calls dark brown field will actually be dark brown field. Wet brown field has a 20% user's accuracy, which means that users would find that not very often (20% of the time) when they visit the area in this image that what the classified map calls wet brown field will actually be wet brown field. Wet brown field is most often confused with dry brown field, as the two pixels that should have been dry brown field was classified as wet brown field.

Producer's Accuracy

For the ML image, the classes with high producer's accuracy include deep water and desert, moderate producer's accuracy include urban, dense vegetation, moderately vegetated field, and sparsely vegetated field, and low producer's accuracy include wet brown field, dry brown field and dark brown field.

Deep water has a 100% producer's accuracy, which means all of the time (100% of the time) that an area should be deep water is actually classified as deep water.   Desert has a 91.67% user's accuracy, which means most of the time (91.67% of the time) that an area should be desert is actually classified as desert.

Urban, dense vegetation, moderately vegetated field, and sparsely vegetated field all have moderately high producer's accuracy, between 57.14 and 66.67%, which means that often (between 57.14 and 66.67% of the time) that an area should be a specific class out of these classes it is actually classified as that class.

Finally, dry brown field have a 16.67% producer's accuracy, which means that rarely (16.67% of the time) that an area should be dry brown field is actually classified as dry brown field. What should have been dry brown field is often misclassified as urban, sparsely vegetated field and wet brown field. And wet brown field have a 0% producer's accuracy, which means that never (0% of the time) that an area should be wet brown field is actually classified as wet brown field. What should have been wet brown field are all misclassified as dry brown field. Similarly, dark brown field have a 0% producer's accuracy, which means that never (0% of the time) that an area should be dark brown field is actually classified as dark brown field.

For the SVM image, the classes with high producer's accuracy include deep water, desert, and dark brown field, moderate producer's accuracy include urban, dense vegetation, moderately vegetated field, and sparsely vegetated field, and low producer's accuracy include wet brown field and dry brown field.

Deep water has a 100% producer's accuracy, which means all of the time (100% of the time) that an area should be deep water is actually classified as deep water. Desert has a 91.67% user's accuracy, which means most of the time (91.67% of the time) that an area should be desert is actually classified as desert. Dark brown field has a 100% producer's accuracy, which means all of the time (100% of the time) that an area should be dark brown field is actually classified as dark brown field.

Urban, dense vegetation, moderately vegetated field, and sparsely vegetated field all have moderately high producer's accuracy, between 41.67 and 71.43%, which means that often (between 41.67 and 71.43% of the time) that an area should be a specific class out of these classes it is actually classified as that class.

Finally, dry brown field have a 16.67% producer's accuracy, which means that rarely (16.67% of the time) that an area should be dry brown field is actually classified as dry brown field. What should have been dry brown field is often misclassified as urban, sparsely vegetated field and wet brown field. And wet brown field have a 33.33% producer's accuracy, which means that relatively infrequently (33.33% of the time) that an area should be wet brown field is actually classified as wet brown field.

Discussion

In the beginning instead of having a general urban class that includes roads, I had three separate classes: dense urban, sparse urban, and road. However, the algorithm cannot distinguish the three specific classes, so I decided to merge the three classes into one urban land cover class type, which made the image results a lot better.

The desert and dense urban areas were often confused, since it seems that the dense urban had similar spectral characters as desert (**Figures 19** and **20**) . I had to refine the polygons, draw more polygons and increase the threshold to 0.2 to get better results. The SVM algorithm also produced better results for these two land cover class types.

At the end, the vegetated water was the most difficult land cover class to map. Water has a lot of variation in its spectral character since there are contaminated water and vegetated water, on top of creating more polygons for the water class I also added more specific land cover type classes for water. For example, instead of just having a general water class I added classes including: deep water, shallow water, slightly contaminated water and very contaminated red water. The algorithms was able to successfully identify all of the deep water class successfully, and for the slightly contaminated water very contaminated red water and shallow water classes, it seems that the algorithm did a pretty good job classifying them, but since they are not very common classes none of the randomly generated pixels that were used to produce the error matrix included these classes, so I didn't get a accuracy metric for these particular classes, though they looked good visually. Where both images had problems were for the vegetated muddy water (**Figures 17** and **18**), as what should have been deep water was instead classified as dense vegetation or dark brown field. This is happening because vegetated water and dense vegetation has really similar spectral characteristics (**Figures 21** and **22**). To address that, I created another class "vegetated water" and ran both ML and SVM algorithms again, but this in turn caused a lot of misclassification of dense vegetation as vegetated water, so in the end I decided against adding this extra class.

I also decided against using any filers. I tried both the majority filter and the sieve filter on the final images, but given that a lot of the smaller fields with crops planted in rows and the algae in the water pools are very detailed with higher resolution, and the canals are very narrow and fine, the filters actually made the results worse and seemed less accurate. For example, the canals were showing up blue in the image before the filter, but after it is just becomes gray. Another example is the dried out lake areas,there is a lot more detailed before the applying the filter (**Figures 16** and **26**).

In terms of differences in user's accuracy between the two images, overall the images have similar user's accuracy when if categorizing high, medium and low classes, but the numbers vary a bit. For example, desert and moderately vegetated field have higher user's accuracy in the ML image compared to the SVM image while urban, sparsely vegetated field, dark brown field and wet brown field have higher user's accuracy in the SVM image. The reason why deep water is accurate is because the spectral character is pretty consistent and doesn't vary as much as the other classes. It is also easier to create more accurate training dataset because the pixels often span a big area. Similarly, desert is also relatively high for both since the spectral characters are relatively similar, however it does get confused as urban every so often and in the case of SVM sparsely vegetated field.   For both images urban is not bad but often what should be dry brown field are misclassified as urban in both images. Dark brown field and wet brown field are low in both images, but in particular bad in the ML image. Often what should be moderately vegetated field, sparsely vegetated field or dry brown field is misclassified as either dark brown field or wet brown field. This is probably due to the fact that there are overlapping spectral characters in the training dataset since there are some fields that have a gradient of pixels, often with sparse vegetation scattered across bare fields (**Figures 23 - 25**).

In terms of differences in producer's accuracy, the biggest difference between the two images is that the dark brown field has a 100% producer's accuracy in the SVM image while 0% in the ML image. This is because there is only one pixel in training dataset out of the 50+ pixels is actually dark brown field. Since the ML algorithm didn't successfully classify that one pixel it got a 0% producer's accuracy, while SVM successfully classified that pixel so it got a 100% producer's accuracy. Urban also had quite a different producer's accuracy, with 57.14% in the ML image and 71.43% in the SVM image. What should have been urban is often misclassified as wet brown field or urban, more often in the ML image. The wet brown field in the SVM image also did better than in the ML image, with 33.33% instead of 0% producer's accuracy.

For the ML image, most classes are pretty consistent when it comes to having similar user's and producer's accuracies, except for dry brown field, since it has a low producer's accuracy (16.67%) but moderate user's accuracy (50%). What should have been dry brown field is often misclassified as wet brown field or sparsely vegetated field. I think this is due to the fact that there are often regions of interests with mixed pixels since the fields are often not purely one class, in reality they often are multiple classes since some bare fields will have some vegetation here and there, and some fields could be between dry and wet.

For the SVM image, most classes are pretty consistent when it comes to having similar user's and producer's accuracies, except for sparsely vegetated field and the dark brown field. Sparsely vegetated field has moderately high user's accuracy (71.43%), but moderately low producer's accuracy (41.67%).   For the most part, when a user visits the field they would find that what is classified as sparsely vegetated field would actually be sparsely vegetated field, but every once in a while they are actually find a wet brown field or dry brown field. That is because wet brown fields and dry brown fields quite often have vegetation sparsely scattered on them, so that would explain why the algorithm misclassified them since these classes are mixed by nature. On the other hand, what should have been classified as sparsely vegetated field is more often classified as something else, for the most part moderately vegetated field but every once in a while desert or wet brown field. Dark brown field has a relatively low user's accuracy (33.33%) but an extremely high producer's accuracy (100%). Not so often when a user visits the field they would find that what is classified as dark brown field would actually be dark brown field, instead it is often actually moderately vegetated field. On the other hand, all of the dark brown fields have been correctly identified as dark brown fields. This is because there is only one pixel in training dataset out of the 50+ pixels is actually dark brown field. So as long as the SVM algorithm successfully classifies the one pixel as dark brown field, it will have 100% producer's accuracy. But if it misclassifies other classes as dark brown field then the user's accuracy would be lower, which is what happened in this case.

In general, the classification could be improved by combining some of the classes (such as dry and wet brown fields), and by creating a better training dataset with more refined ROIs, and maybe creating the same number of ROIs for each land cover type class.

Figures and Tables

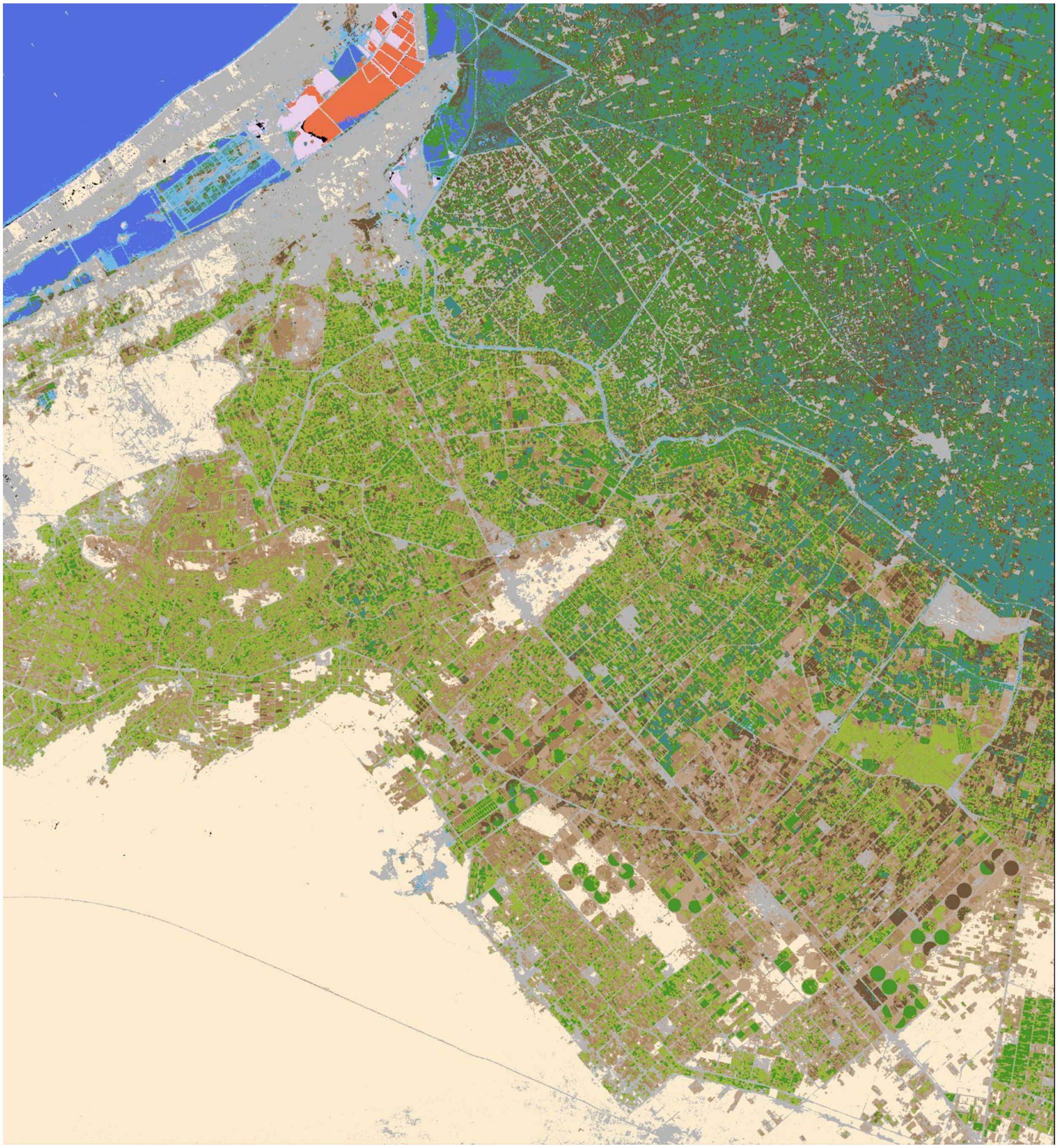

Figure 1. Final map produced using ML

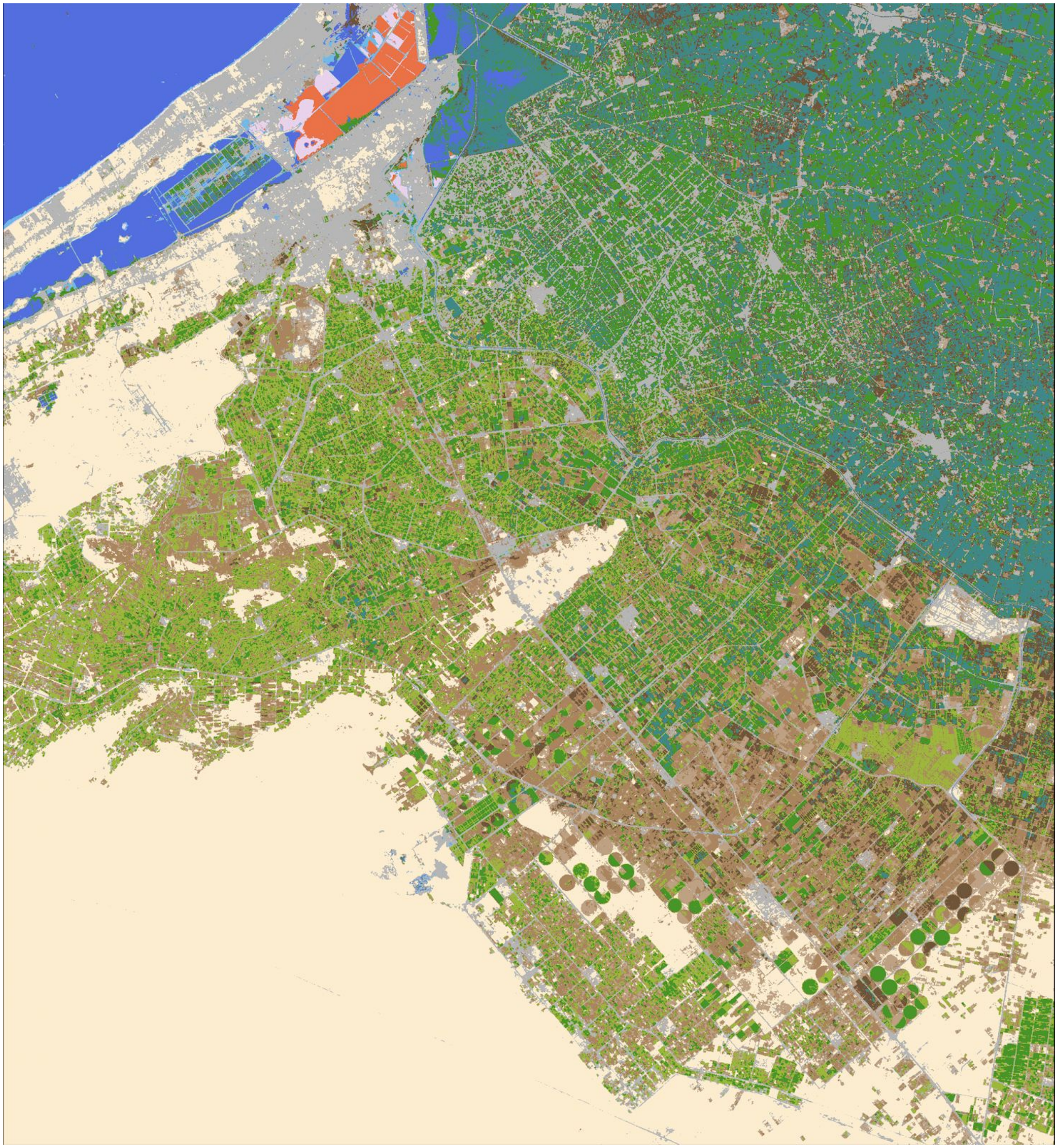

Figure 2. Final map produced using SVM


Figure 3. Not very good desert vs urban vs dry brown field classification in ML

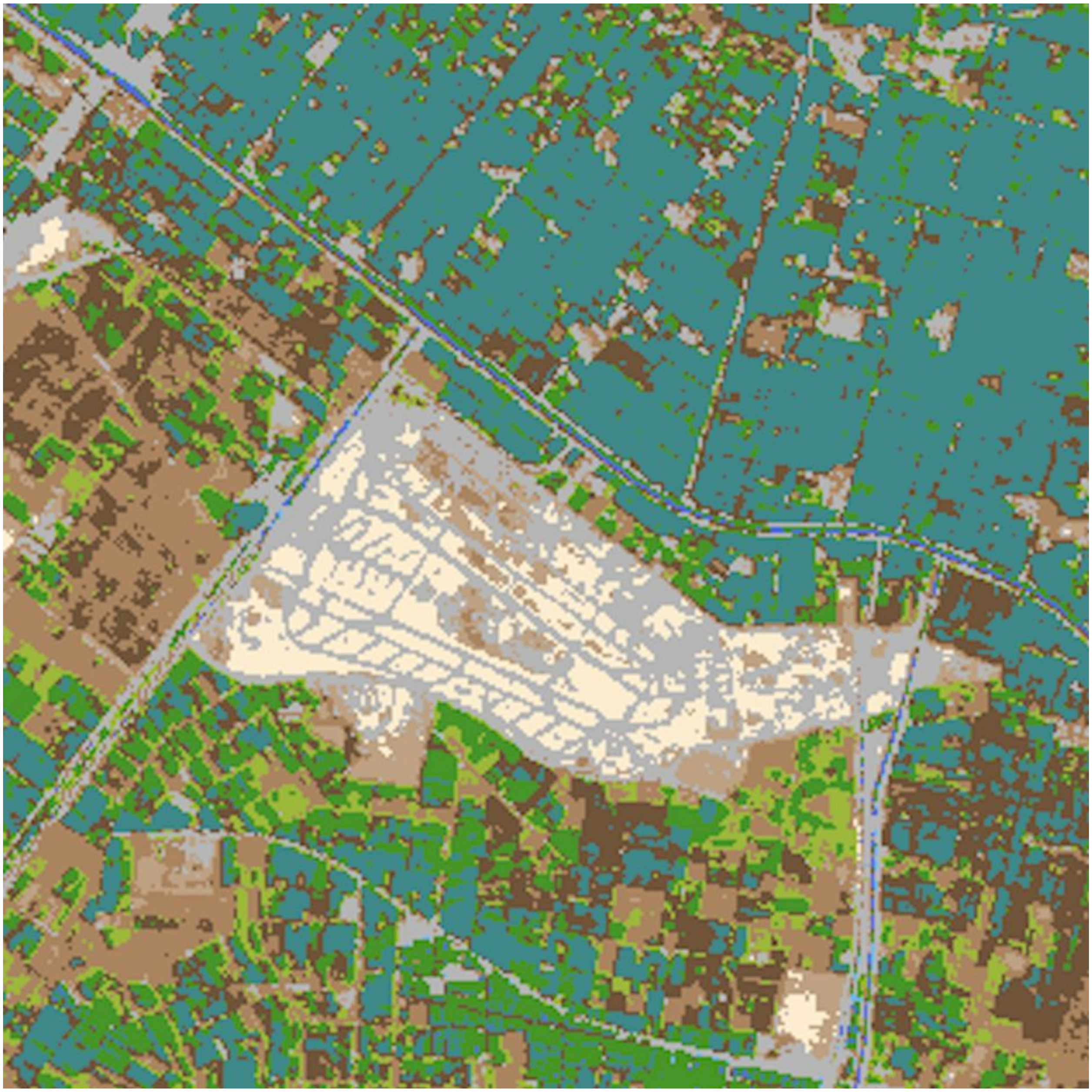

Figure 4. Better Desert + Urban classification in SVM

Figure 5. Not very clean classification of fields in ML

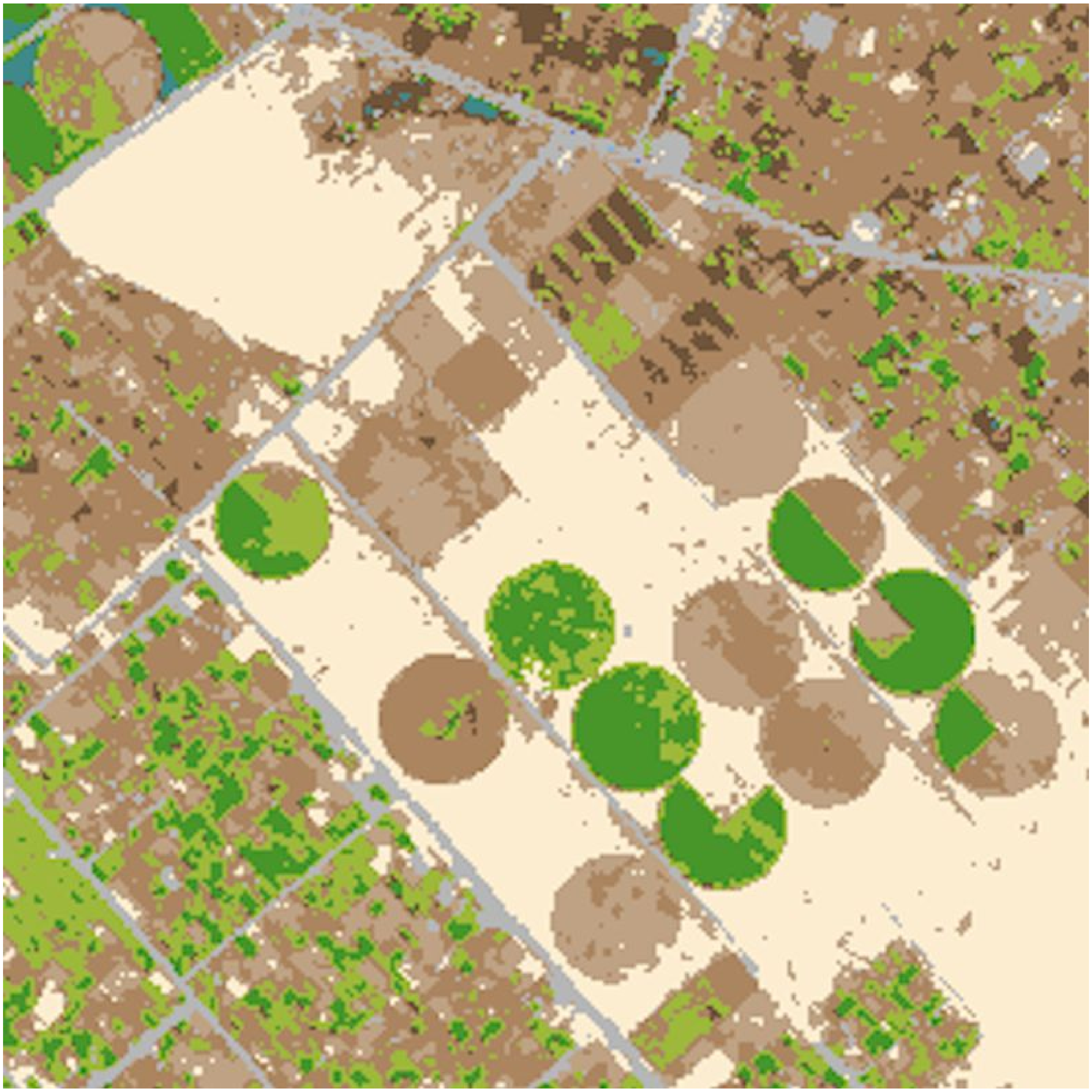

Figure 6. Clean classification of fields in SVM

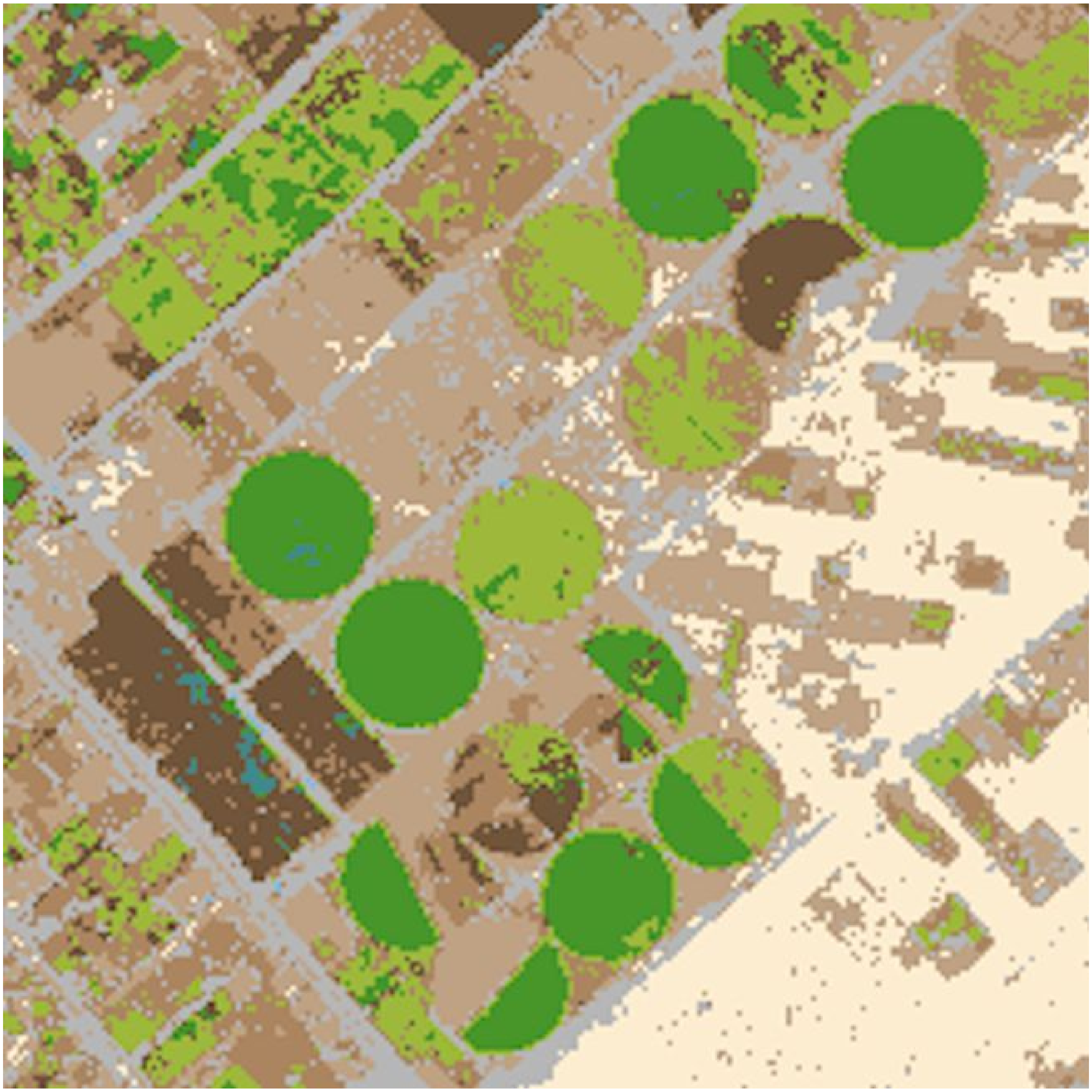

Figure 7. Not very clean classification of fields in ML

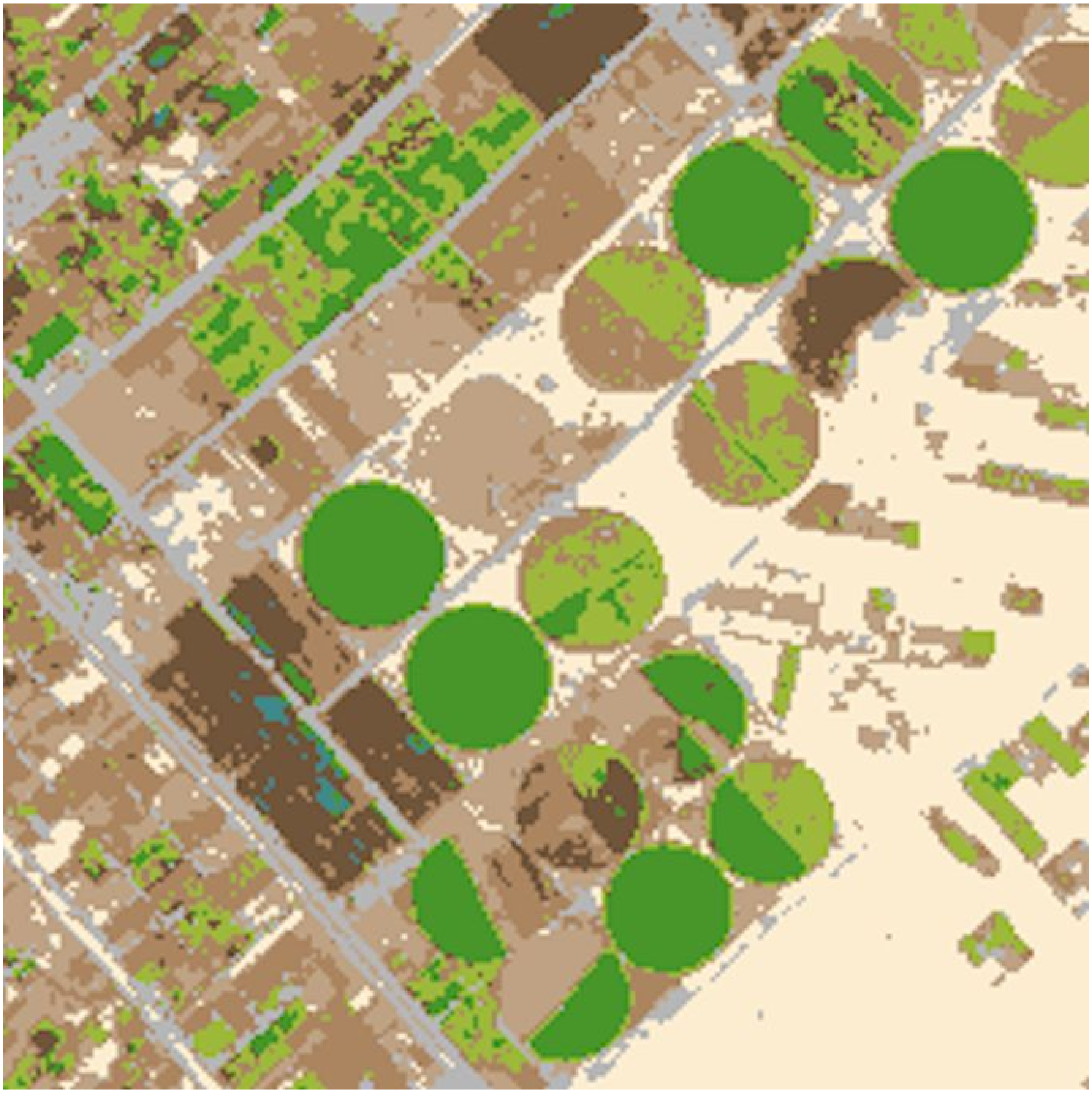

Figure 8. Clean classification of fields in SVM

Figure 9. Not very clean road classification in ML

Figure 10. Cleaner and finer road classification in SVM

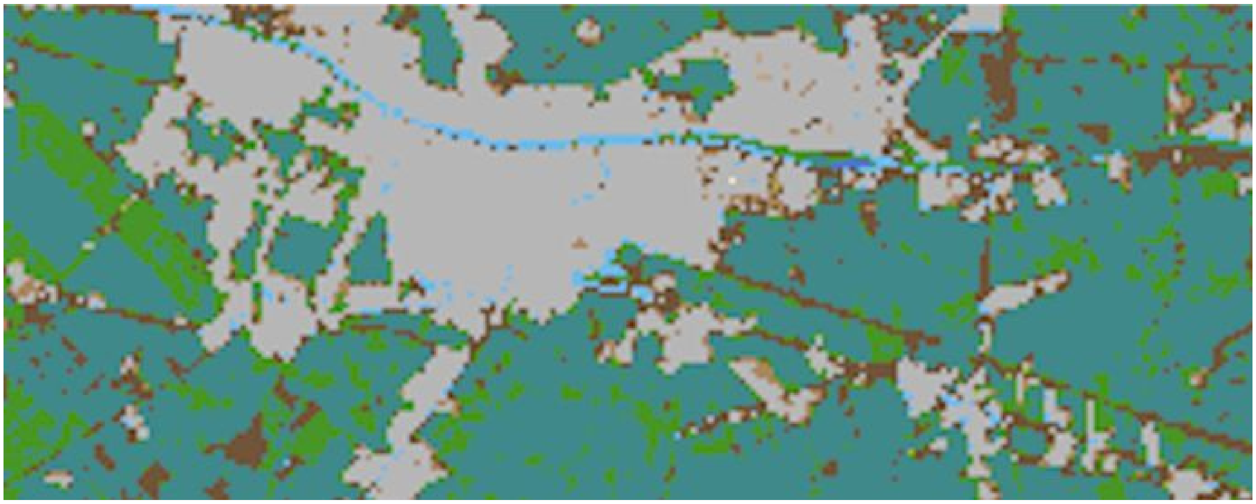

Figure 11. Good classification of canals in ML

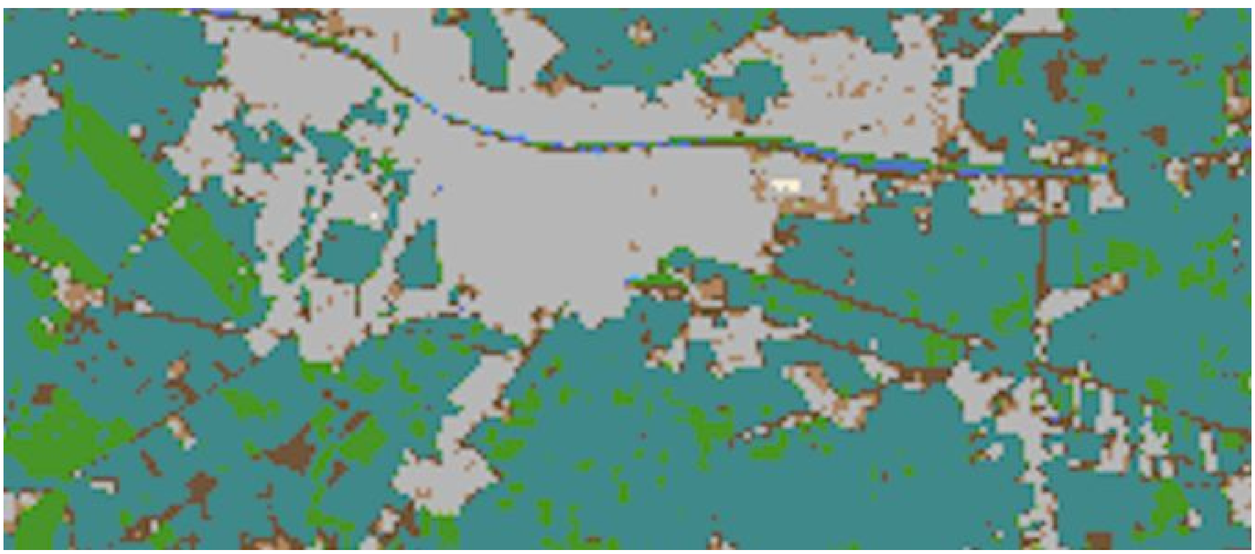

Figure 12. Not very good classification of canals in SVM

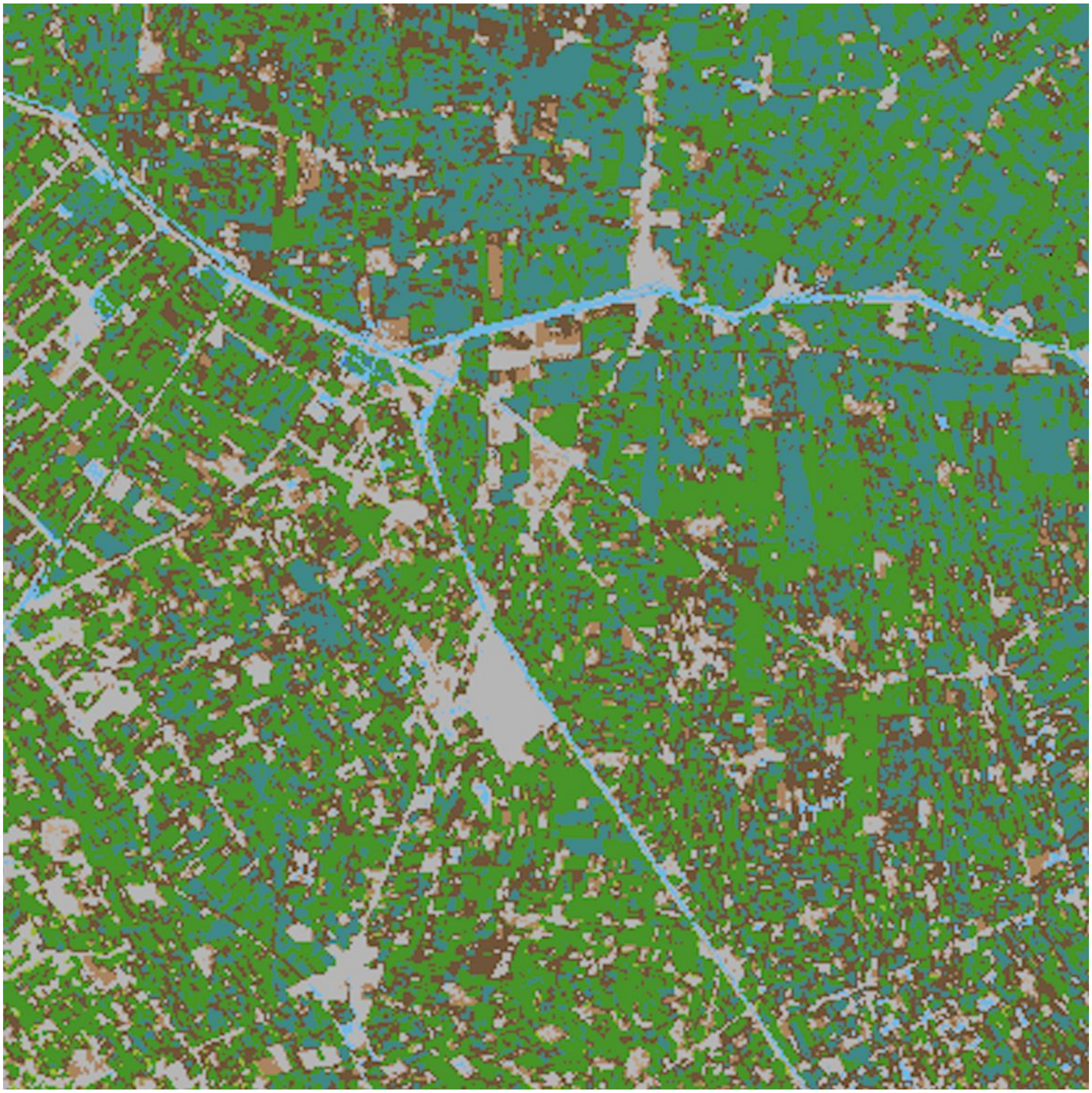

Figure 13. Good classification of canals in ML

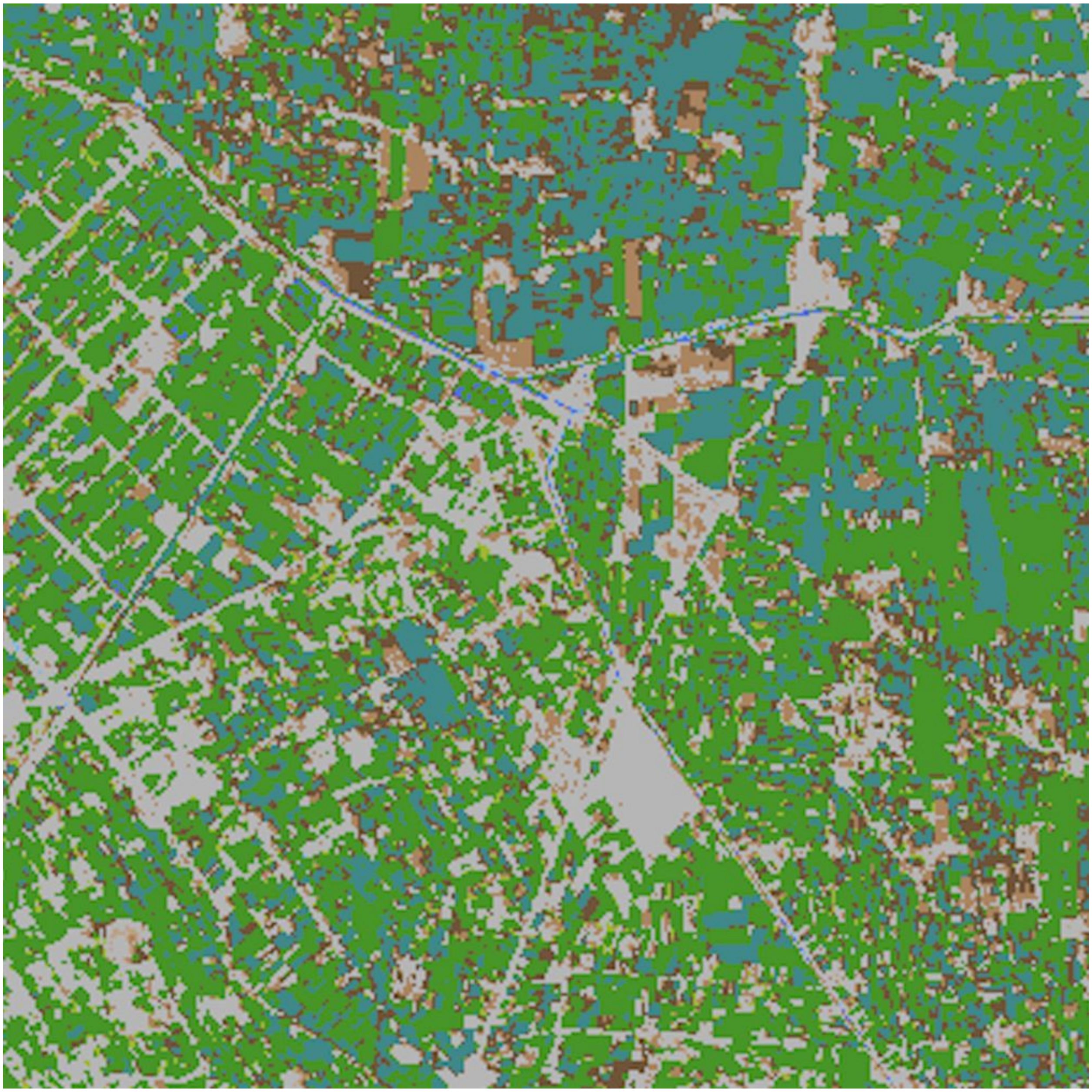

Figure 14. Not very good classification of canals in SVM

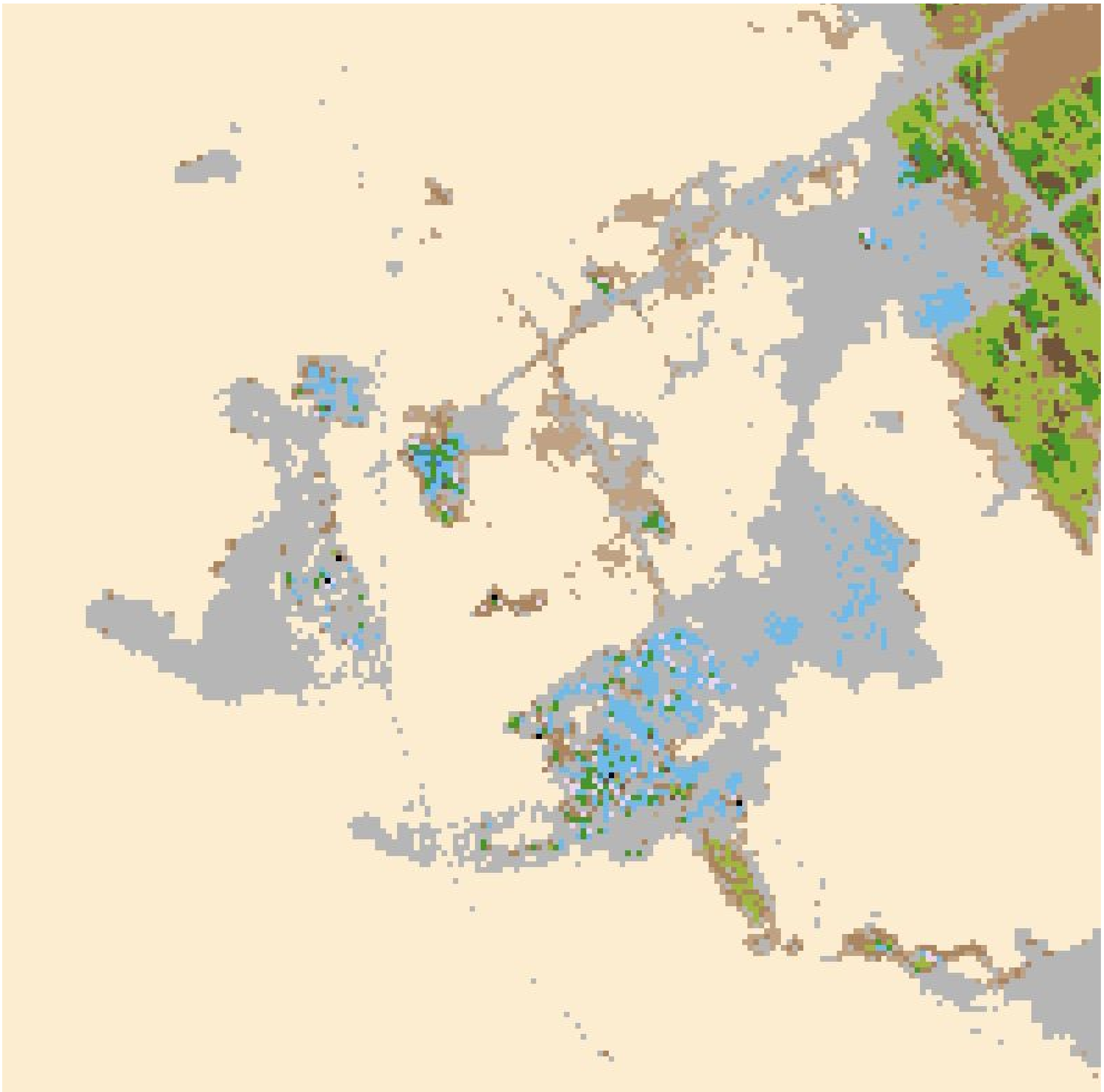

Figure 15. Not very good classification of drying up lake in ML

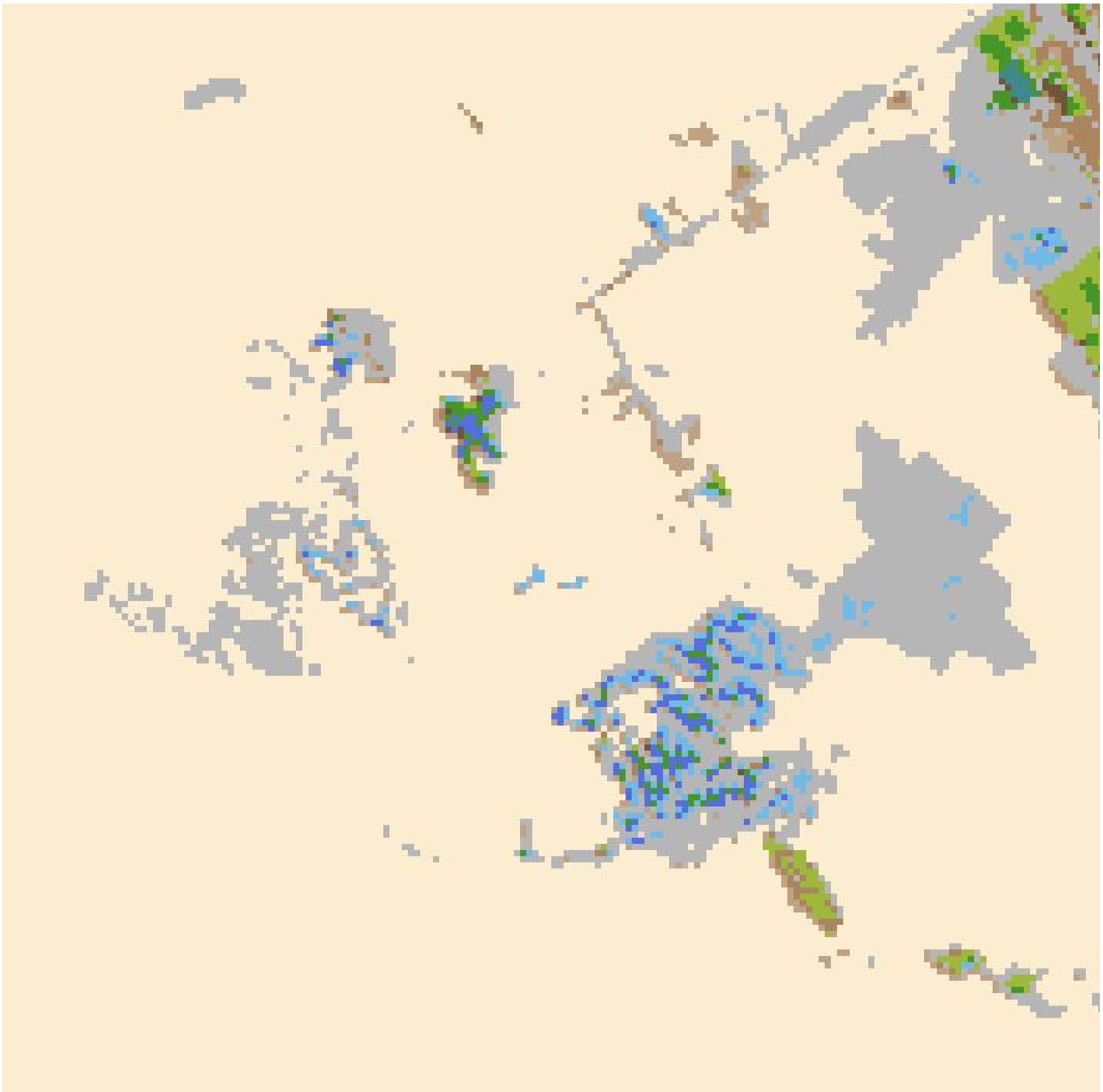

Figure 16. Better classification of drying up lake in SVM

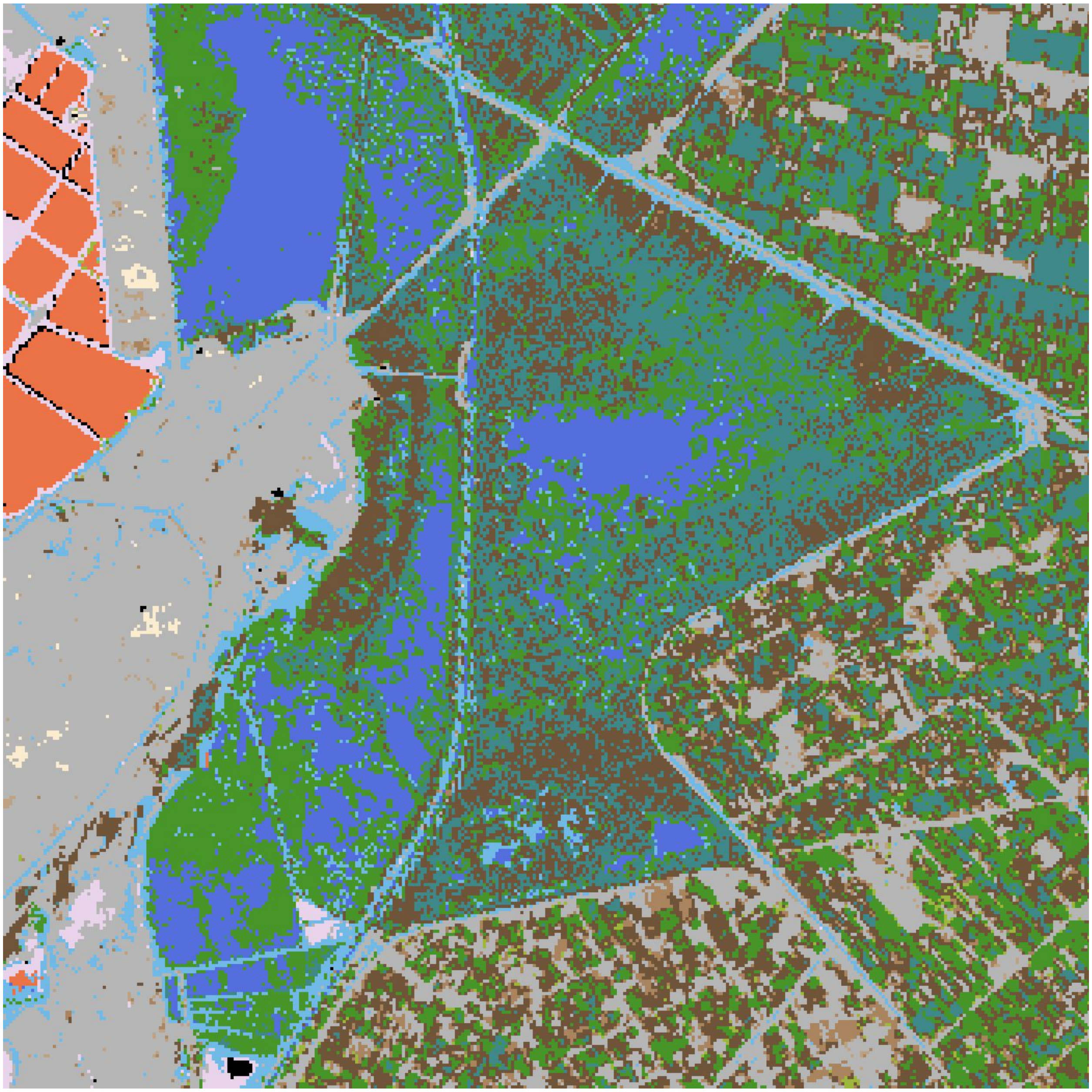

Figure 17. Not very good classification of deep water (with vegetation and mud) in ML

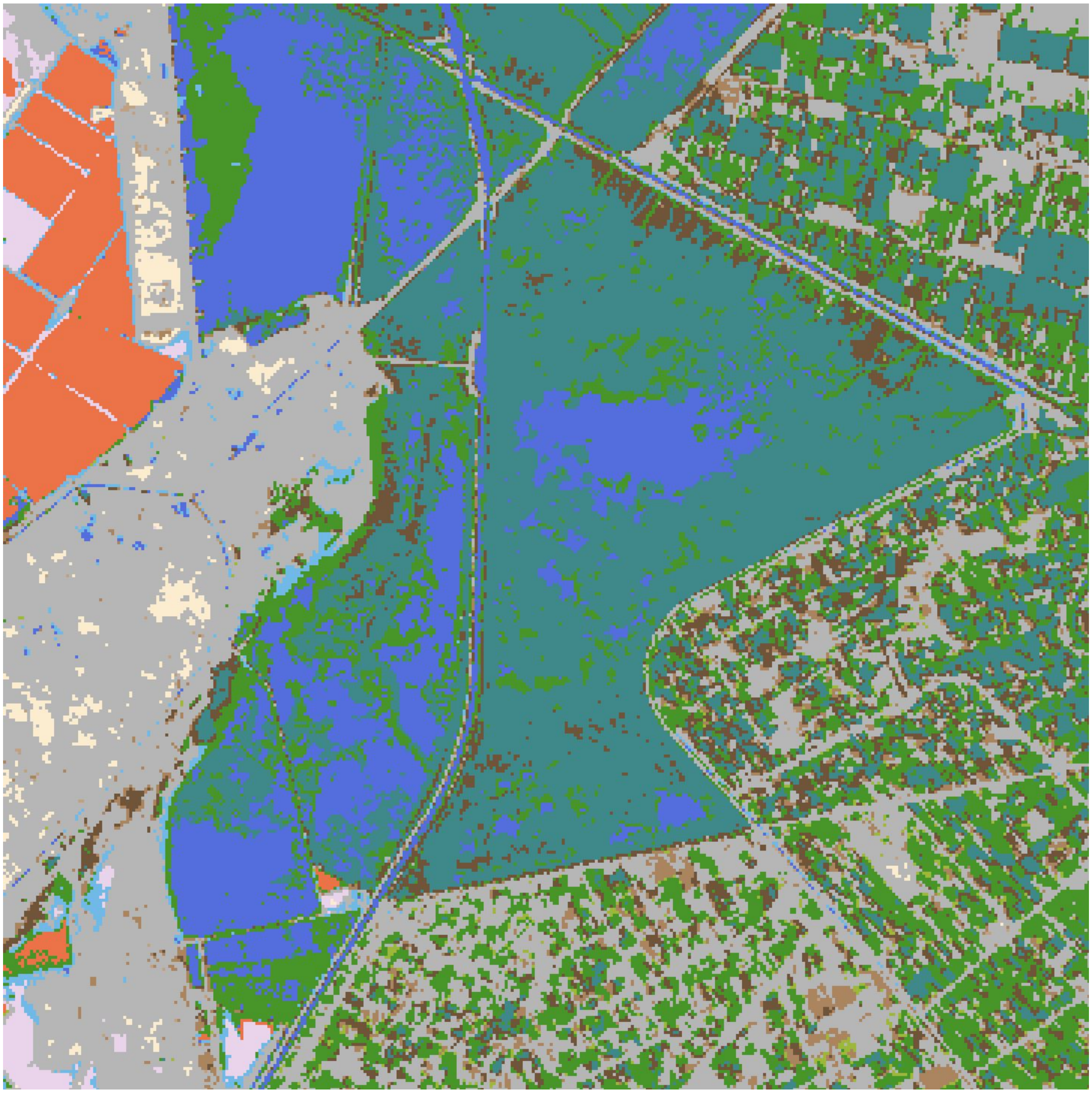

Figure 18. Not very good classification of deep water (with vegetation and mud) in SVM

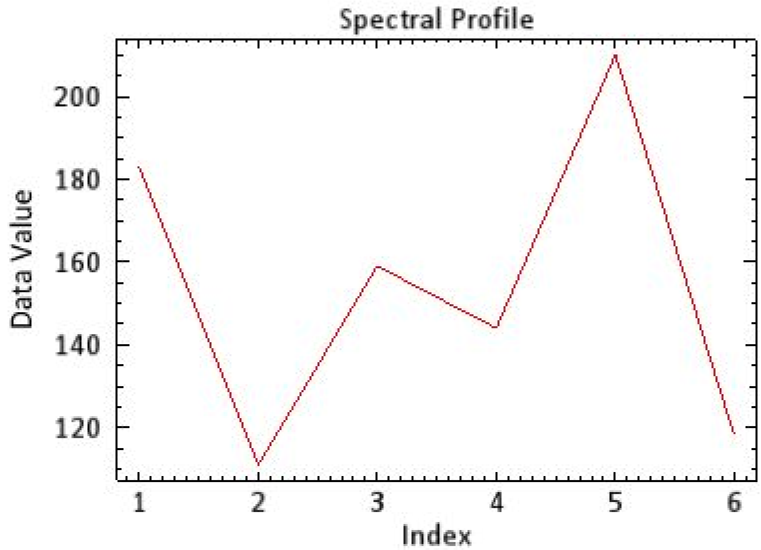

Figure 19. Spectral profile of dense urban

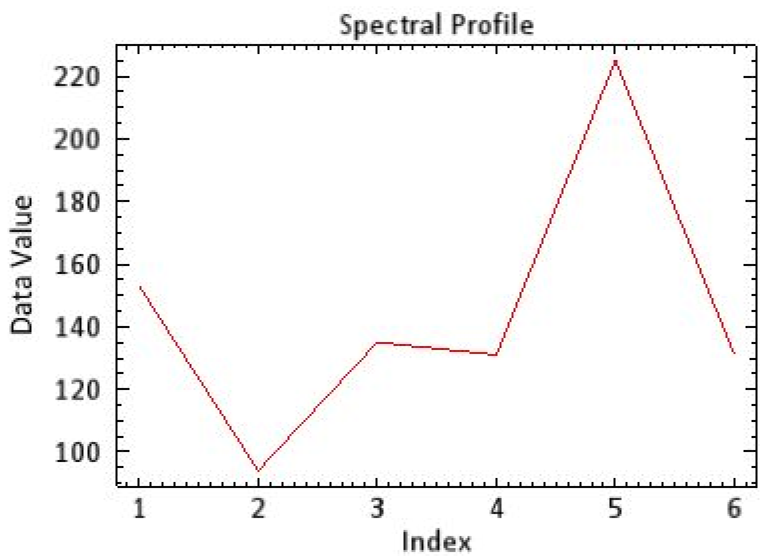

Figure 20. Spectral profile of desert

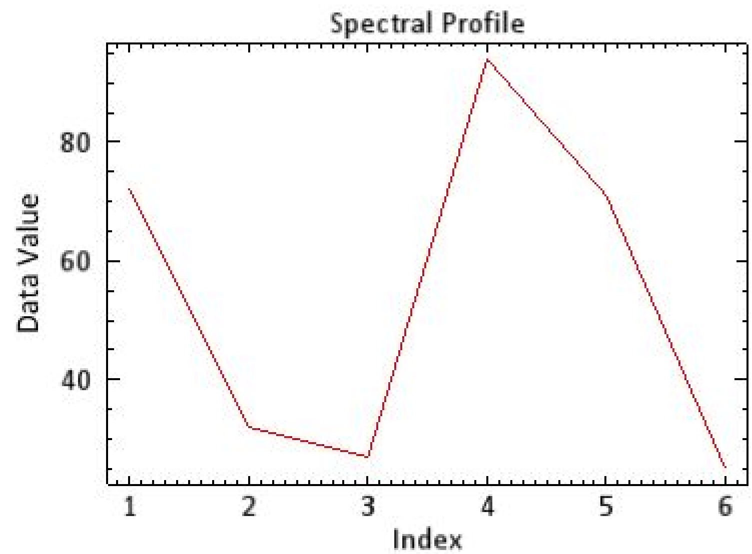

Figure 21. Spectral profile of dense vegetation

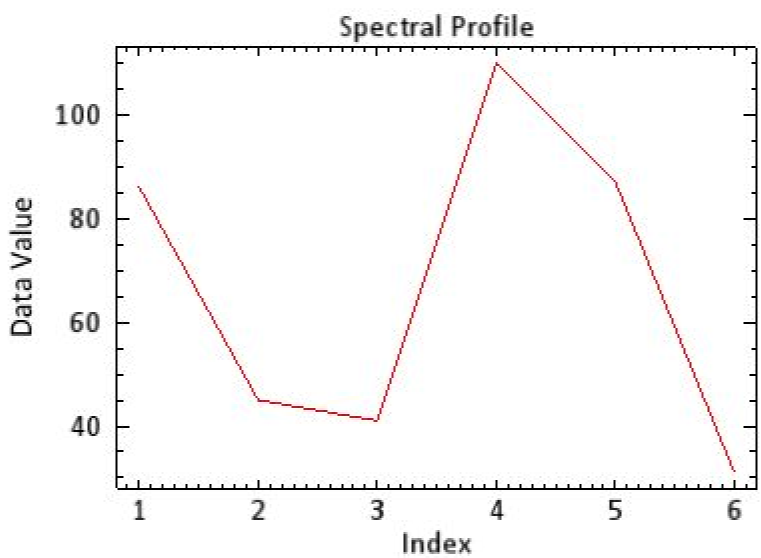

Figure 22. Spectral profile of vegetated water

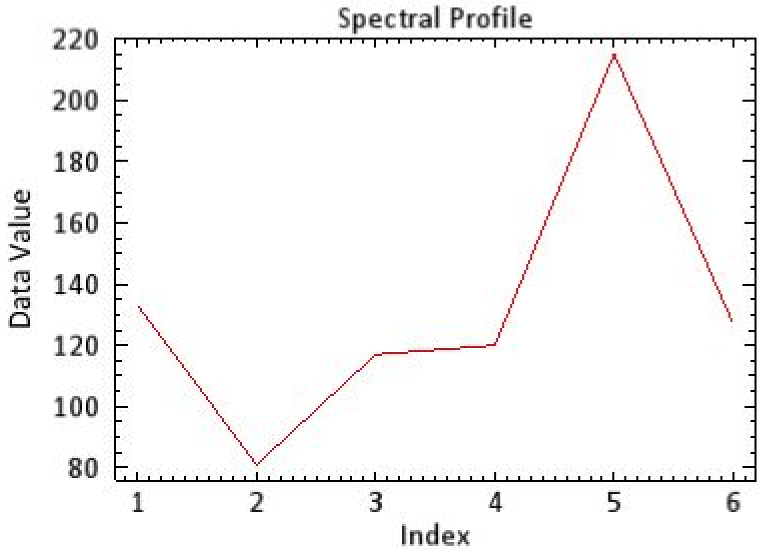

Figure 23. Spectral profile of dry brown field

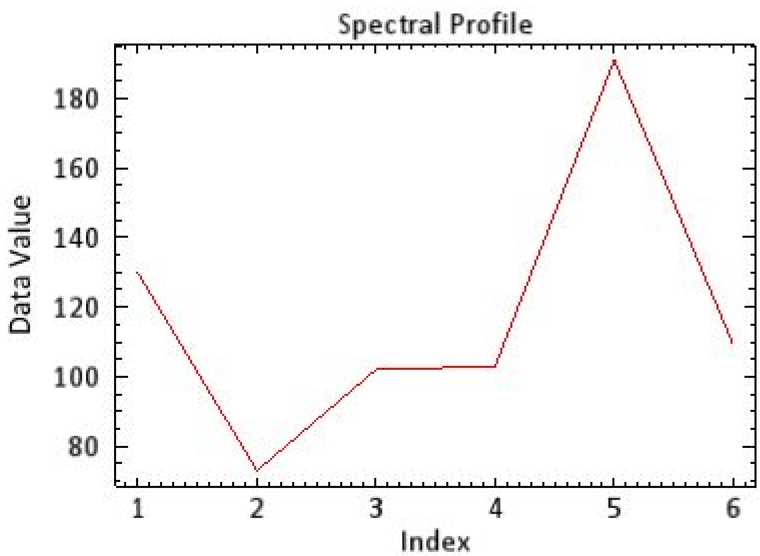

Figure 24. Spectral profile of wet brown field

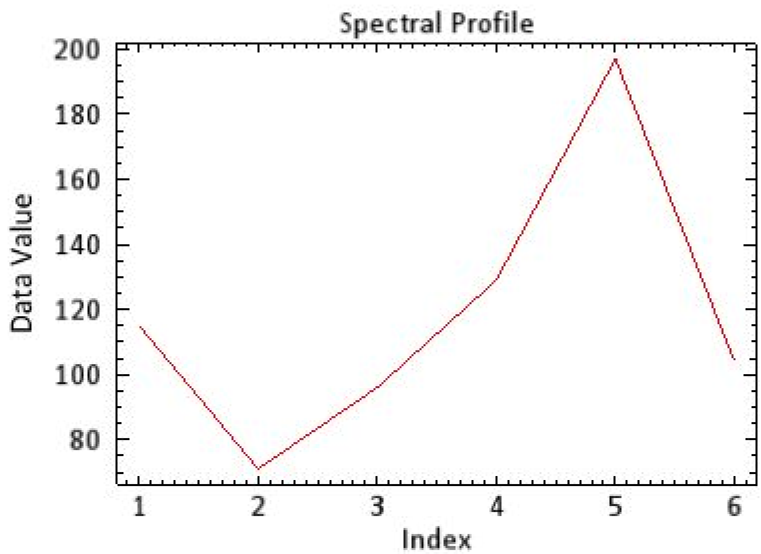

Figure 25. Spectral profile of sparsely vegetated field

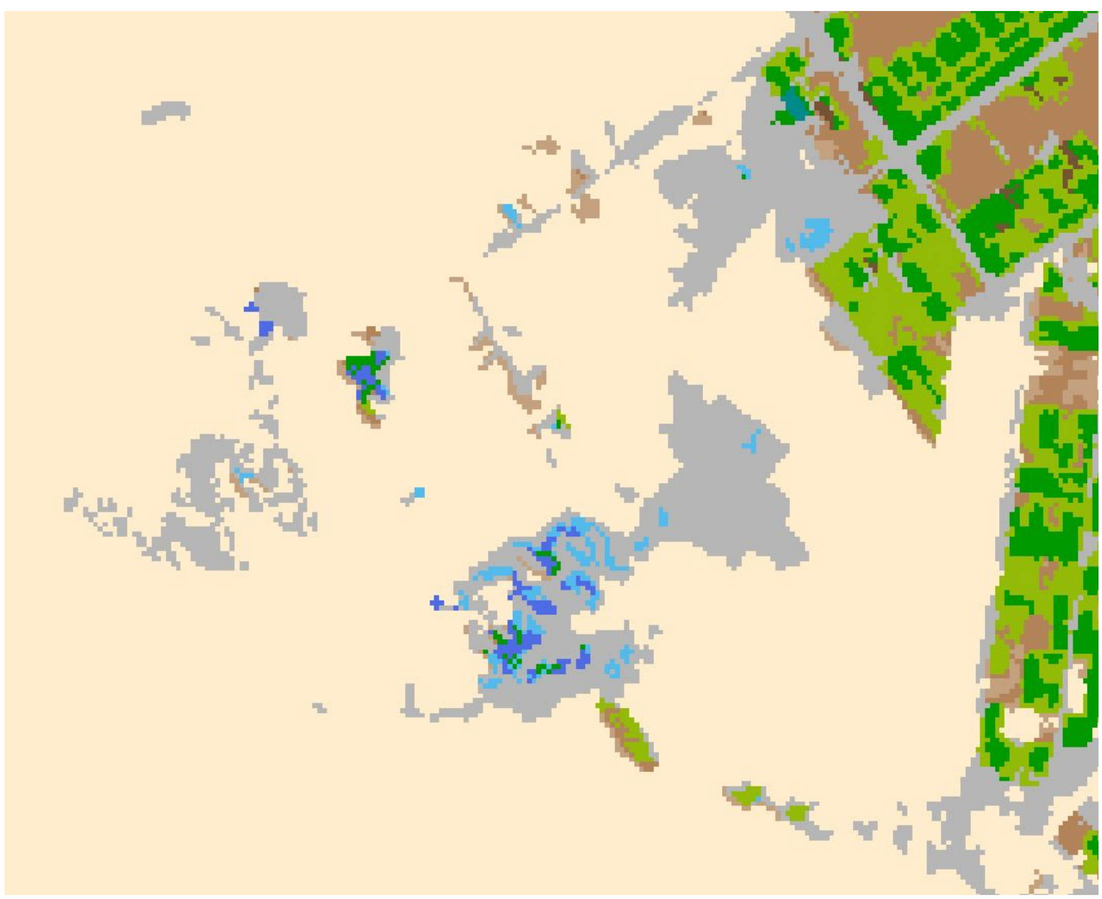

Figure 26. Dried up lake in SVM image loses detail with sieve filter

Class	deep water test	urban test	moderately vegetated field test	dense vegetation test	desert test	sparsely vegetated field test	dry brown field test	wet brown field test	dark brown field test	contaminated water test	very contaminated red water test	shallow water test	Total	User's accuracy
water	4	0	0	0	0	0	0	0	0	0	0	0	4	100
urban	0	4	0	0	0	0	2	0	0	0	0	0	6	66.67
moderately vegetated field	0	0	6	1	0	2	0	0	0	0	0	0	9	66.67
dense vegetation	0	0	2	2	0	0	0	0	0	0	0	0	4	50
desert	0	1	0	0	11	0	0	0	0	0	0	0	12	91.67
sparsely vegetated field	0	0	0	0	0	7	2	3	0	0	0	0	12	58.33
dry brown field	0	0	0	0	1	0	1	0	0	0	0	0	2	50
wet brown field	0	2	0	0	0	3	1	0	0	0	0	0	6	0
dark brown field	0	0	1	0	0	0	0	0	0	0	0	0	1	0
contaminated water	0	0	0	0	0	0	0	0	0	0	0	0	0	0
very contaminated red water	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Shallow water	0	0	0	0	0	0	0	0	0	0	0	0		
Total	4	7	9	3	12	12	6	3	0	0	0		56	
Producer's accuracy	100	57.14	66.67	66.67	91.67	58.33	16.67	0	0	0	0			

Table 1. Error Matrix for ML

Class	deep water test	urban test	moderately vegetated field test	dense vegetation test	desert test	sparsely vegetated field test	dry brown field test	wet brown field test	dark brown field test	contaminated water test	very contaminated red water test	shallow water test	Total	User's accuracy
Unclassified	0	0	0	0	0	0	0	1	0	0	0	0	1	
deep water	5	0	0	0	0	0	0	0	0	0	0	0	5	100
urban	0	5	0	0	0	0	2	0	0	0	0	0	7	71.43
moderately vegetated field	0	0	6	1	0	5	0	0	0	0	0	0	12	50
dense vegetation	0	0	1	2	0	0	0	0	0	0	0	0	3	66.67
desert	0	1	0	0	11	1	0	0	0	0	0	0	13	84.62
sparsely vegetated field	0	0	0	0	0	5	1	1	0	0	0	0	7	71.43
dry brown field	0	0	0	0	1	0	1	0	0	0	0	0	2	50
wet brown field	0	1	0	0	0	1	2	1	0	0	0	0	5	20
dark brown field	0	0	2	0	0	0	0	0	1	0	0	0	3	33.33
contaminated water	0	0	0	0	0	0	0	0	0	0	0	0	0	0
very contaminated red water	0	0	0	0	0	0	0	0	0	0	0	0	0	0
shallow water	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	5	7	9	3	12	12	6	3	1	0	0	0	58	
Producer's accuracy	100	71.43	66.67	66.67	91.67	41.67	16.67	33.33	100	0	0	0		

Table 2. Error Matrix for SVM