Classification of MODIS Data

Xuechang Liu

Abstract

Three classification methods of MODIS data are performed and compared in the case study of biomass burning area of Okhotsk in far-east Russia on Jul 19, 2011. Total seven MODIS bands are used to classify five classes over the region. Thematic maps are produced by maximum likelihood classification, migrating means classification and hybrid classification, respectively. The statistical data and classification results are analyzed. Based on the comparison between thematic maps and ground truth, hybrid classification is the best method for this case study, and the second is maximum likelihood classification. Migrating means classification has relatively large misclassification.

1. Introduction

The Moderate Resolution Imaging Spectro-radiometer (MODIS) is an advanced and critical sensor equipped aboard the Terra (AM Orbit) and Aqua (PM Orbit) satellites. Both Terra and Aqua MODIS have 36 spectral bands ranging from visible to long-wave infrared, with minimum wavelength of 0.41µm to maximum wavelength of 14.4µm. Measurements at nadir are made at three spatial resolutions: 250m (Bands 1 and 2), 500m (Bands 3 - 7) and 1km (Bands 8 – 36) [Xiong et al., 2009].

For the study to quantitatively analyze the satellite remote sensing data, classification of the image is applied. Conventional multivariate classification methods used to be popular but are not ideal for the case because of their multi-types and uncommon units (thus the scaling problem) and the data's unequal reliability, whereas statistic methods that are based on Bayesian classification theory are general for all data types [Benediktsson et al., 1990]. Supervised classification is the machine learning process that derives an inferred function (classifier) from labeled training data. The supervised methods rely on this classifier to discriminate the actual members of that class [Barandela et al., 2002]. Many methods are developed for this task, and are tagged as "soft classification" or "hard classification" by whether they have overlaps or firm boundaries [Schowengerd, 1996]. Unlike supervised classification, which allows the analyst to supervises its learning about the parameters, unsupervised classification is the machine learning that finds hidden structure in unlabeled data. Hybrid classification is the combination of both of the above two methods. This approach first uses unsupervised method to perform classification to generate spectral clusters, and then classifies the whole image data with a supervised method. The accuracy, repeatability and efficiency of data classification can be improved this way [Lang et al., 2008].

Maximum likelihood classifier is a popular supervised classifier, which assumes that each pixel is comprised of a single class [Foody et al., 1996]. This requires sufficient numbers of representative pixels to be selected as one specific class, and then a discriminant function will be established on the statistical parameters of the selected pixels (training samples). This classifier will be used to determine the class membership of each pixel based on the highest of the class conditional probabilities. Clustering algorithms, perhaps the most common unsupervised methods, group a set of pixels with somehow similarity into a same cluster, within which the pixels are more similar to each other than to those in other clusters. Migrating means clustering (or k-means clustering) method, a most common clustering method, will be used for the case. A user-specified number of arbitrary cluster centers will first be assigned, and the distance of each pixel to those centers will be computed and closest ones assigned to that cluster, thus to make a crude initial cluster assignment. The clusters' means will generate a new set of cluster centers, and these new centers are passed to the last step to repeat the clustering. This iterative process stops when the cluster centers stop migrating or some user defined stopping rules are met, for example, the value of SSE stops decreasing or two sequential SSE values' difference is less than a specified number. Hybrid method first uses migrating means method on divided pieces of the data and then training samples will be picked on these pieces to perform maximum likelihood classification.

This case study employs the Aqua MODIS data over the eastern Russian sea in late July, 2011. Far-east Russia has one of the most distinctive temperate forests in the world. This region is also one of the best and last examples of temperate broadleaf and mixed forests, and home to many rare animals. So the wild fires that tend to burn in this region in summer are very devastating to the Siberia's unique ecosystem.

Section two introduces the datasets. Section three describes the methodologies that are used for this study. Section four analyzes and discusses the classification results of these three methods. The last section gives the conclusions.

2. Data

The domain used for this case study is the Sea of Okhotsk, Russia, as shown in Figure 1, the area on a projected map[1]. The time of the Aqua MODIS image acquired is July 19th, 2011 at 0220 UTC. From Figure 2, the three-band overlay true color image, we can see there was a very thick smoke plume spreading from the west side through the ocean area in the lower middle part. The size of the original data is 1354 pixels by 2030 pixels, and the domain for this study is shown inside the dashed lines, for which the size is 640 pixels by 512 pixels.

MODIS reflective solar bands (RSB) 1–19 and 26's wavelengths ranging from 0.41 to 2.2 µm collect data only during spacecraft daytime. Bands 20–25 and 27–36 are thermal emissive bands (TEB) that measure temperature during both daytime and nighttime. To perform classification, the selection of bands must be based on both the statistics of the whole 36 bands' data and spectral signatures. Table 1 is the correlation matrix[2] and Table 2 is the statistical facts about the data, from which we can see:

1) For the RSBs, bands are not highly correlated to each other except band 3 (0.459-0.479 µm) and band 4 (0.545-0.565 µm). Band 6 has 40% of negative values. Thus band 1 (0.620-0.670 µm), 2 (0.847-0.876 µm), 3, 5 (1.230-1.250 µm) are employed.

2) The reflectance values for and 8 through band 16 are all negatives because they are ocean color/phytoplankton/biogeochemistry bands, thus these bands are excluded for the study.

3) Band 20 to band 23 are surface/cloud temperature bands. Band 20 (3.660-3.840 µm) is selected for it contains no negative values low similarity with other bands.

4) Band 31 (10.780-11.280 µm) and band 32 (11.770-12.270 µm) are very important for aerosol detection [Zhao et al., 2010].

It can be seen from the image that this area is very complex with the pervasive smoke all around, so radar sounding, weather map and CALIPSO images are used too.

3. Methodology

Supervised classification, unsupervised classification and hybrid classification [Richards, 2013] will be described in this section.

3.1 Supervised Classification - Maximum Likelihood Classification (MLC)

The following steps are conducted to perform maximum likelihood classification.

Step 1: Decide what types and how many classes the image will segment into based on the true color image, i.e., vegetation, water, smoke, three classes in total.

Step 2: Choosing known, representative pixels for each class as training data from each training field. Sufficient training pixels must be chosen for the next step to avoid the covariance

[1] All figures are attached in Appendix A.
[2] All tables are attached in Appendix B.

matrix being singular. For an N dimensional spectral space at least N(N+1) independent samples are needed.

Step 3: Establish the signature of the classes by using the training samples to estimate MLC parameters, assuming the classes of pixels are normally distributed. For and N dimensional space, the multivariate Gaussian distribution function is

$$p(\boldsymbol{x}/\omega i)=2\pi^{-N/2}\exp\{-1/2(\boldsymbol{x}-\boldsymbol{m}_i)^T\boldsymbol{C}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i\ )\} \tag{1}$$

Where $\boldsymbol{m}_i$ and $\boldsymbol{C}_i$ are the mean vector and covariance matrix of the data in class ωi.

Step 4: Using the trained classifier to label every pixel in the image as belonging to one of the classes of Step 1. If no prior probabilities' information is known, then they are assumed to be equal. The MLC discriminant function is then written as

$$g_i(\mathbf{x}) = -\ln|\mathbf{C_i}| - (\mathbf{x} - \mathbf{m_i})^T \mathbf{C}_i^{-1}(\mathbf{x} - \mathbf{m_i}) \tag{2}$$

Step 5: Using the decision rule

$$\mathbf{x} \in \omega_i \text{ if } g_i(\mathbf{x}) > g_j(\mathbf{x}) \text{ for all } j \neq i \tag{3}$$

to decide to which class this pixel belongs to, and then assign the index of this class to the position of this pixel.

Step 6: A thematic map of class indices will be produced by applying the classifier in Step 5. Color-code the thematic map with different colors to denote different classes. The area of the different classes can be computed by removing the effect of panoramic distortion and earth curvature using the equations below

$$p_c = \beta[h + r_e(1 - \cos\emptyset)]\sec\theta\sec(\theta + \emptyset) \tag{4}$$

$$p_a = \beta[h + r_e(1 - \cos\emptyset)]\sec\theta \tag{5}$$

Where p_c is the cross track effect panoramic distortion. β is the instantaneous field of view (IFOV) and h is the height of the sensor. $\emptyset$ is the latitude and θ the viewing zenith angle (VZA) of the pixel. r_e is the radius of the Earth.

The corrected area of a pixel is

$$A = p_c \times p_a \tag{6}$$

3.2 Unsupervised Classification - Migrating Means Classification (MMC)

The k means or migrating means or iterative optimization algorithm is performed as the following steps.

Step 1: Select a value for C, as the number of clusters into which the pixels are to be grouped. This value can be arbitrary.

Step 2: Select C points in multi-spectral space to serve as candidate clusters centers. Each point is a vector with the dimension of the number of the bands, call these

$$\hat{\mu}_i \quad i = 1, 2, \dots C$$

The selection of $\hat{\mu}_i$ is arbitrary, with the exception that no same ones. But for better results, a diagonal set of $\hat{\mu}_i$ is chosen for this study.

Step 3: Perform Euclidean Distance for each pixel vector **x** to the candidate clusters to decide which group the pixel is assigned to. This will produce an initial cluster assignment.

Step 4: Compute a new set of cluster means from the groups formed in Step 3; call these:

$$\mu_i \quad i = 1, 2, \dots C$$

Step 5: If $\mu_i \approx \hat{\mu}_i$, for all C clusters, then the process is terminated. Otherwise $\hat{\mu}_i$ are assigned with the current values of μ_i and Step 3 will be repeated.

To examine if the μ_i and $\hat{\mu}_i$ are close enough, the sum of squared errors (SSE) is used, which is

$$SSE = \sum_{C_i} \sum_{x \in C_i} \left| \vec{x} - \overrightarrow{\hat{\mu}_i} \right| \tag{7}$$

Ideally, the “perfect” condition would be SSE=0, but for large dataset this is not likely to happen. So a user-defined threshold can be set to stop the iteration.
A more practical way to finish the cycling process is set stop-point when the cluster centers do not migrating any more.

Step 6: Label each class with the aid of ground truth information (the true-color image, etc.). Merge some clusters into one class if necessary.

Step 7: Create a colored thematic map and compute the area of each class using the classified indices.

3.3 Hybrid Classification

Hybrid classification is the combination of both MLC and MMC. It is performed as the following steps below:

Step 1: Divide the original data into several small pieces.

Step 2: Select arbitrary numbers and positions of cluster centers and perform k-means classification on the small pieces separately.

Step 3: Pick training samples based both on the divided pieces of unsupervised classification result and on ground truth information. Perform MLC with these training samples to produce a supervised classified result.

Step 4: Color-code the final result to display a thematic map and compute the areas.

4. Results and discussion

4.1 Supervised classification results

Figure 3 is the true-color extracted image of the area of study, which is displayed with three MODIS visible bands – band 1 (red), band 4 (green) and band 3 (blue). By looking at this image, we can roughly distinguish five difference classes from it: clouds, ocean, soil, vegetation and smoke. Figure 4 shows the training samples picked for maximum likelihood classifier. A total number of 10 picks of 10×10 pixel box for each class is used. Thus, for a single band, the number of training sample is 10×100×5=5000. From Table 3 the covariance matrix of training samples and Table 4, statistical data for training samples, we can see the reflectance values for the clouds are abnormally low, while the temperature values are too high. Compared to the true-color image, we can assume that this is because the clouds are contaminated by the smoke aerosols, since for most part the smoke can be seen mixed with clouds. Supervised classification of this data, however, can still be carried out since despite the low reflectance of clouds in visible and near infrared bands and higher-than-common temperatures of band 31 and 31, the differences between each training class still stand out. The reflectance values of all classes' pixels are not typical compared with common spectral signatures, but it explains well if taken into account that they are more or less affected by the smoke. For example, from the training data, the smoke aerosols could drag down the reflectance of all visible bands and pull up the temperature of cloud, so we see a mean reflectance of clouds of visible bands are under 0.5, the temperatures of band 31 and 31 are over 279 kelvin, and mean temperature of band 20 is up to 301 kelvin. The temperature of smoke is very close to (but slightly higher than) that of ocean, but the reflectance is higher. From the true-color image, the (barren) soil type and the yellowish vegetation type are somewhat in similar conditions, so I treat them as one type, therefor, some vegetation and soil pixels might be misclassified with each other. The major difference between soil and vegetation is that soil's temperature is the higher, actually the highest among all these five classes. Also, ocean training pixels have the smallest reflectance standard deviation and the soil pixels have the smallest temperature standard deviation, whereas the clouds have the highest standard deviation in both reflectance and temperature bands. The large standard deviation values of cloud is due to the multiple phase inside the cloud itself, the contamination of smoke aerosols and the overlapping of other classes, for instance, the cloud over vegetation/soil and the over ocean still looks cloud to eyes, but the temperature and mass reflectance might have been influenced.

Figure 5 is the thematic map of supervised classification. Compared with Figure 3, the result is quite satisfying. It addresses the middle part of the data decently, which is the most ambiguous part with a seemingly mixture of at least two types of clouds, water and smoke. MLC renders a result that is very true to the eyes with the three-band overlay, but gives more about the details, especially for the upper right corner, where I was not able to pick samples because of its complex nature. On the true-color image, this part seems to be severely contaminated by the smoke.

However, the ragged clouds and the land beneath the smoke/cloud are still visible, thus it is very hard to tell what class it is. What fails to show on the overlay is shown on the MLC result. Note that MLC thematic map is actually an index map, which assigns the pixel to the class it "most likely" belongs to, so albeit the edges and classes detected, this area is still taken by multiple layers, thus there is some misclassification that address smoke/cloud as soil type. Upper left corner and lower right corner ocean part have some pixels are misclassified, too. The cloud training pixels I picked from the left corner are very tricky, for they have very high temperature and a broad range of reflectance (from 0.2 to 0.6). As the smoke plume comes from the left, as the fire spots shown in Figure 6, the west of my study region of the same date, the cloud on the left part is almost completely contaminated. Some green water pixels (might be phytoplankton) can be seen on the east coast of the lower land, and these pixels are classified as vegetation.

The mean and standard deviation values of MLC classified data is shown in Table 5. The reflectance of clouds is even lower, all visible bands fall beneath 0.4, but still can be distinguished from other classes. All classes' temperature values of 3 thermal bands are lower than the training samples. The largest standard deviation values are still caused by clouds, which is due to their inhomogeneous structure and contaminations.

The areas and numbers of pixels for each class are calculated in Table 6, we can see that smoke takes the most pixels as well as the area, and the second is the ocean and then the cloud. But ocean' area is smaller than the cloud's due to the distortion. Table 7 is the confusion matrix for maximum likelihood classification. This shows that MLC has a very high producer accuracy, which means the procedure runs well, and is able to process most of the pixels the way they are picked samples. It takes 224.997 seconds to run the MLC procedure[3].

4.2 Unsupervised classification results

Unsupervised classification does not require any pre-knowledge of the ground truth information of the data or spectral signatures of the classes. This avoids the mistakes that could happen when picking training samples.

Figure 7 is the initial classified result with 11 clusters. From this image we can see that the k-means classifier's addressing of the classes of vegetation, soil and ocean is very coarse, but generates 7 clusters of different cloud types. These cloud types differ from their inner natures and the extent of being contaminated by the smoke. For example, the cloud on the left side, which looks on the true-color image like one single class, but is distinguished four sub types by k-means. The result shows that the west-most white part is also the cloud with highest reflectance, which is true to the training samples I picked.

Another example is the lower left corner, where these details help discriminate multiple layers of clouds and aerosols which are not easy to tell from true-color image. And the part on the upper right side, where I was not able to pick samples for MLC, is proved to be consisted with multiple types of clouds. These clouds' reflectance and temperature are not only affected by the smoke, but also by the underlying surface, their own physical phases, the water contents and the height.

Compared the final merged thematic map in Figure 8 (all cloud sub-clusters are merged as one class "Cloud", regardless of the contamination extent or physical phases) with the true-color image, the classification of land types is not very good. Apart from the misclassification on the left side (the same spot on the MLC result), it seems that k-means differentiates the land types by

[3] All procedures and main level programs involved are attached in Appendix C

the gap of the ocean/smoke, because the green vegetation and yellowish vegetation/soil look like half-half on the north of the waters, while the result shows an overwhelming portion of soil type. But it is also possible that the vegetation types across the waters are different, which cannot be fully verified by the true-color image.

The classification result for smoke and ocean does not agree with the three-band overlay very well. The ocean area on the lower middle between the smoke plume and the lower land is a relatively clear area without much aerosol to eyes, but the k-means assigns outrageous smoke pixels to that, while on the left side, the area right below the miscellaneous cloud, which ought to be smoke polluted, is detected as pure ocean. Adding more bands and more cluster centers might help improving k-means, but this will also increase the time to run the program. Seven bands of 640×512 MODIS data with eleven initial clusters take 472.232 seconds to run, and the final sum of squared errors (SSE) is 2.051×10^8, and 52 times of iterations are performed.

The mean and standard deviation values of k-means result are shown in Table 8. The reflectance of clouds of visible bands is still lower than normal, but is notably higher than that of MLC result. Smoke aerosol has the smallest standard deviation values, and this might verify that the smoke spreads all around this region, causing the anomalous spectral signatures of other classes. Table 9 is the tabular summary of k-means thematic map, it shows that ocean is the class that claims the most pixels and areas, followed by smoke. Unsupervised classification confusion matrix is tabulated in Table 10. This table again verifies that k-means is not very ideal for the detection between smoke and ocean in this case study.

4.3 Hybrid classification results

To perform the hybrid classification method, the original image is first split into four pieces to conduct k-means separately. I divide the image evenly by its width into four 160×512 pixel pieces, and roughly classify it into six classes. And then I pick samples as I did for MLC, but separately on these small pieces. Total number of 10×10 box of 10 pick on each piece for each class of pixels is picked, which is up to 100×10×5×4=20,000 training samples. The training samples' mean values and covariance matrix are shown in Table 11, and the mean vectors and standard deviation of training data are shown in Table 12. The reflectance of cloud is slightly lower than that of the MLC training samples, but the temperatures are lower too. The k-means thematic maps can be the reference for the sample picking, for example, for the areas, due to their complexity of various classes mingled, I was not able to pick samples for MLC, but I can pick them now. Because the small pieces of k-means thematic map has already had a coarse result for me, the picks can be more accurate and more convenient. The more homogeneous the data is, the standard deviation values are lower. So this explains why the standard deviation values are lower than those of the MLC.

Figure 9 is the thematic map of hybrid classification. Since the samples are picked with the knowledge of some details that are hidden from the true-color image but exposed on the k-means thematic map, it has the advantages of both MLC and k-means. For example, Figure 9 has a very fine and delicate expression of clouds over the central part, which does not only show a rough shape of the cloud (like in Figure 5, some characters are somewhat smoothed or blurred), but also the clear edges, like in k-means. For the left part, on the clouds, MLC has quite a few misclassified spots, which exist on k-means thematic map and hybrid result too, but the amount s on the latter two are much smaller. On the south to the clouds, MLC assigns too many smoke pixels while k-means almost assigns all ocean pixels to this region. But hybrid method renders a reasonable result that is consistent with the true-color data. And also, it has better performance on land types and smoke/ocean discrimination than the k-means. MMC distinguishes almost the whole land on the north of the water as soil type, but this problem is fixed when I pick samples. The land cover on the hybrid thematic map resembles k-means' result more, but is more acceptable.

Investigation into the statistics of the hybrid classified data, as shown in Table 13, the visible and near-infrared bands' reflectance of clouds is even lower. Actually it has the lowest cloud reflectance values and highest temperature values of all these three methods. But the differences of values of each band are still outstanding.

The number of pixels and distortion-removed areas of each class are shown in Table 14. For this method, clouds have the most pixels and largest area, which is compatible with the thematic map. Table 15 is the confusion matrix of hybrid method. The sample picking is based on the k-means result, so the accuracy for vegetation and soil is relatively lower than other classes.

Generally, hybrid result has the least misclassification pixels and best match with the true-color image so far.

4.4 Discussions

By the comparison between thematic maps of three classification methods and the true-color image, we can simply tell that hybrid method has the best result. The tricky part of the data I used for this case study is seriously contaminated by the smoke aerosols from the east. Generally, the reflectance of cloud of MODIS 1 to 3 is well above 0.7 and about 0.6 at band 5. And the temperature for clouds at band 31 and 31 is between 240K (ice phase) and 280K (water phase). My data, however, the mean values are about 0.25 (classified data) for reflective bands and 278K for thermal infrared bands. Other classes have similar conditions, but it is the most typical for the clouds.

To verify the effects of smoke, CALIPSO data are employed in Figure 10, which is about the same time of the MODIS data. Figure 11 is the CALIPSO data track overplotted on the true-color image. With the classification result, the track across the water areas is almost aerosol-free, but we still can see the low cloud over the land on the south is contaminated, and on some "cloud-free" parts, the clouds are mixed with the smoke. Figure 12 is the sounding image on that date of the area of the west-most miscellaneous clouds on the true-color image. We can see that the dew-point and the temperature profiles intersect at very low height (around 87m), so the cloud (or maybe fog) is very low, thus very vulnerable to the spreading smoke aerosols. Thus, we can infer that the clouds and other classes are contaminated by the smoke. But since they are still discriminable by using the classifiers, which means their distinctive natures are not totally eliminated or all the pixels are completely polluted, the thematic maps are still labeled as their original classes.

Figure 13 – Figure 15 are the histograms of the classification results. The general outlines of the methods are very similar, which denotes that the result of each classifier is somewhat consistent. From the results, hybrid method gives the most cloud pixels of 103041, which takes 31.45% of total pixels, and k-means give the most ocean pixels of 89082, which is 27.2% of total pixels, and maximum likelihood gives the most pixels of smoke, 85566 of pixels and 26.11% of total pixels. All three methods classified the vegetation as the smallest group, with MLC of 25076 pixels (7.65%), k-means of 37339 pixels (11.39%), and hybrid 31852 pixels (9.72%).

How the supervised classification and unsupervised classification related to each other is plotted in Figure 16. We can see that the plot of smoke aerosol sitting on the diagonal line, which means it has nearly the same areas in both methods. But this does not assure that k-means would be an ideal classifier for smoke. This figure and confusion matrixes together prove again that the vegetation class and soil class are very easy to be misclassified.

The average producer's accuracies for the three methods are 89.00%, 72.00%, 87.00% for maximum likelihood, migrating means and hybrid method, respectively. It seems that the MLC has the best accuracy, but the fact is, the accuracy is related to many factors, including the selection of ground truth data (which are the pixels from true-color overlay image), the number of the pixels, etc. So even if the accuracy of MLC is slightly higher, the hybrid method is still the best classification for this study.

5. Conclusion

Supervised, unsupervised and hybrid classifications for MODIS data are performed in this project. Maximum likelihood classifier (MLC) is used for supervised method. The very important point for MLC is the selection of training data. Migrating means (or k-means) classification is used for unsupervised classification. This is a highly automatic procedure. Hybrid method employs both methods for initial clusters determination and training sample picking. The sample picking based on k-means result improves the precision, thus provides better MLC result.

The MLC result has some misclassification of soil over cloud area, and some details being blurred, but most features are addressed well. The k-means method has same misclassifications too, and larger misclassifications between vegetation and soil classes (vegetation pixels are abundantly classified as soil type), smoke and ocean classes (classes switched). The Hybrid method still has some minor mistake of cloud/smoke pixels being classified as soil, nevertheless, the overall effect of most pixels is decent.

The results and statistical data are analyzed. Each method has its own advantages and advantages. But after comparing the thematic maps produced by the three methods with the true-color image, the producer accuracies and other data, for this case study, the best method is considered to be the hybrid method.

6. References

Barandela,R., Ferri,F., Nájera, T.: Some Experiments in Supervised Pattern Recognition with Incomplete Training Samples. SSPR/SPR, 518-527,2002

Benediktsson,J., Swain,P., and Ersoy,O., Neural network approaches versus statistical methods in classification of multisource remote sensing data, IEEE Trans. Geosci. Remote Sensing, 28,540-552, 1990

Foody,G., Arora., M. Incorporating mixed pixels in the training, allocation and testing stages of supervised classifications. Pattern Recognition Letters, 17(13), 1389-1398,1996.

Lang,R., Shao,R., Pijanowski,B. and Farnsworth, R. Optimizing unsupervised classifications of remotely sensed imagery with a data-assisted labeling approach. Computers & Geosciences 34(12), 1877-1885,2008

Schowengerdt, R., On the estimation of spatial-spectral mixing with classifier likelihood functions, Pattern Recognit. Lett., 17(13), 1379–1387, 1996.

Xiong, X, Chiang, K, Sun, J, Barnes, WL, Guenther, B, Salomonson, VV ., NASA EOS Terra and Aqua MODIS on-orbit performance. ADVANCES IN SPACE RESEARCH, 43(3), 413-422,2009.

Zhao, Tom X.-P., Ackerman, Steve; Guo, Wei. Dust and Smoke Detection for Multi-Channel Imagers. Remote Sensing. 2(10), 2347-2368,2010.

Appendix A

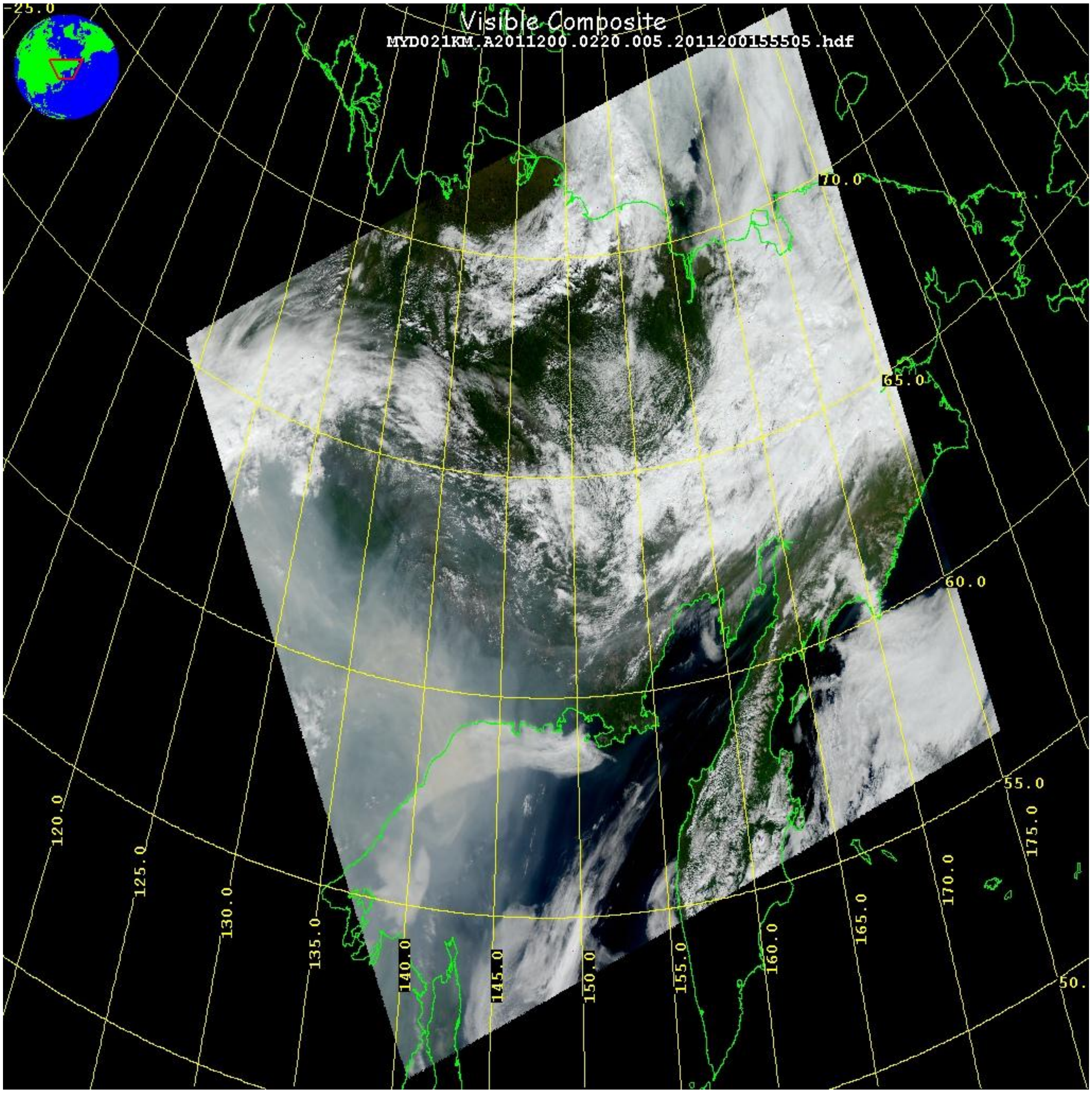

Figure 1The area on a projected map

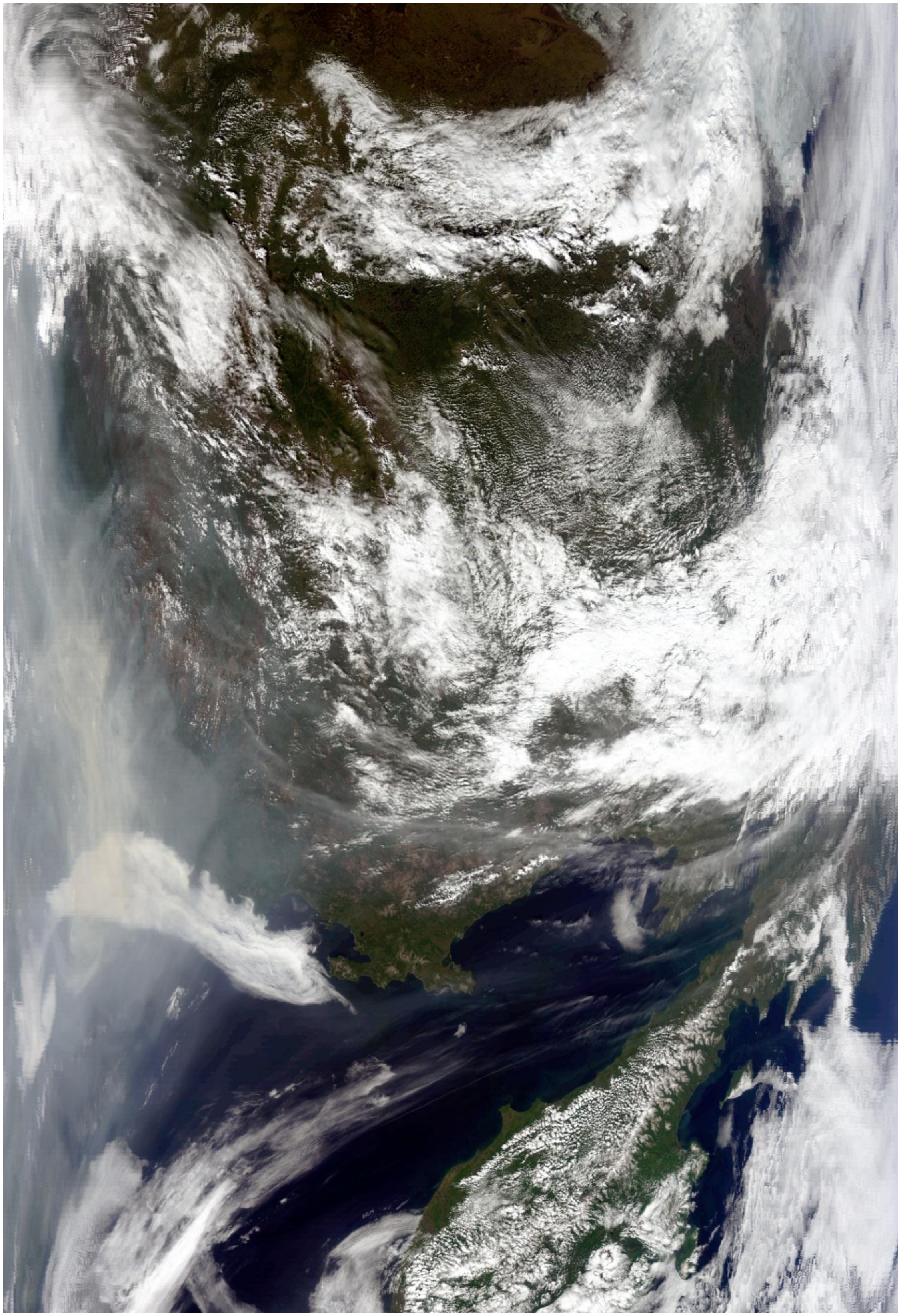

Figure 2 Three-band overlay true color image

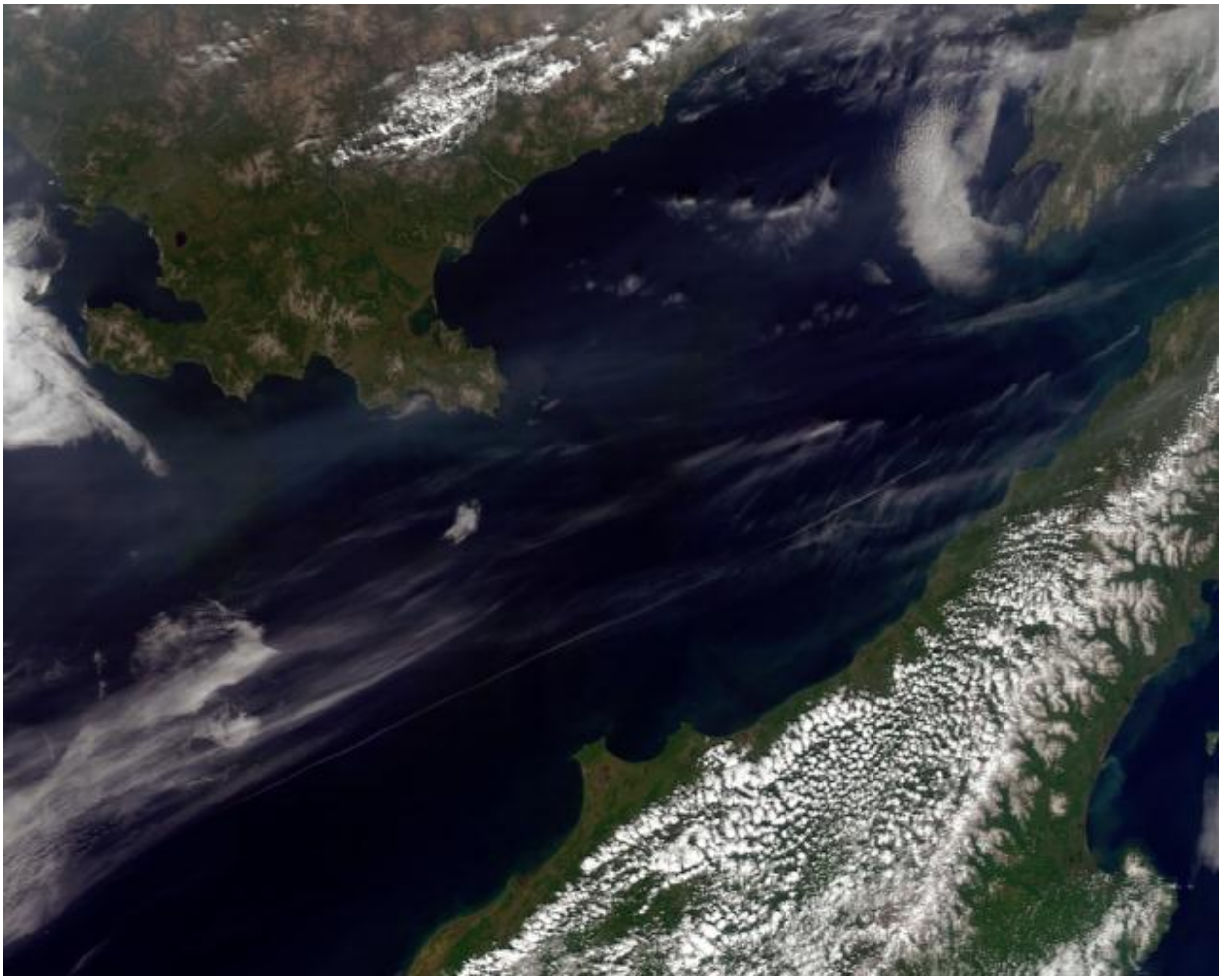

Figure 3 The true-color extracted image of the area of study

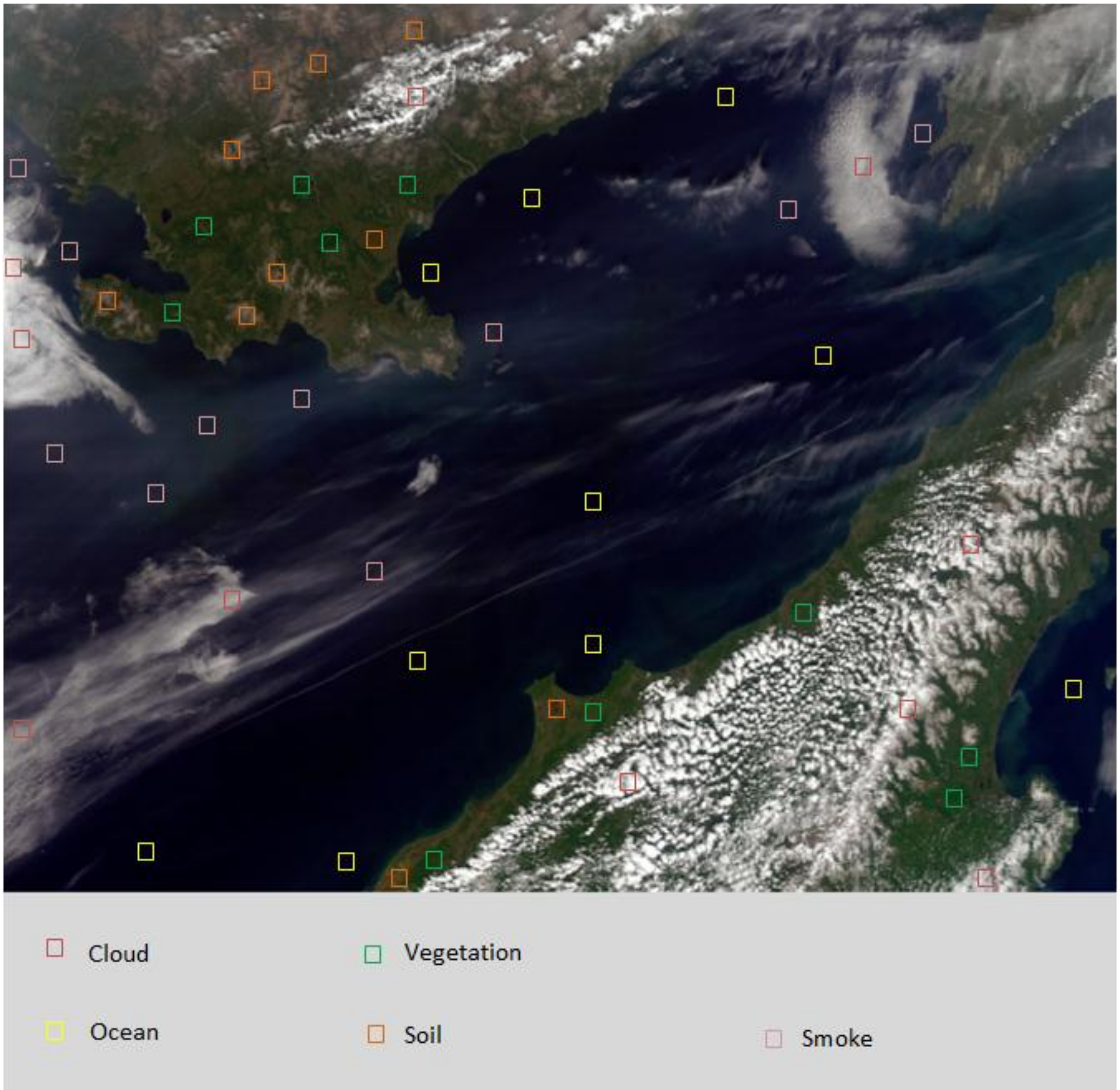

Figure 4 The training samples picked for maximum likelihood classifier

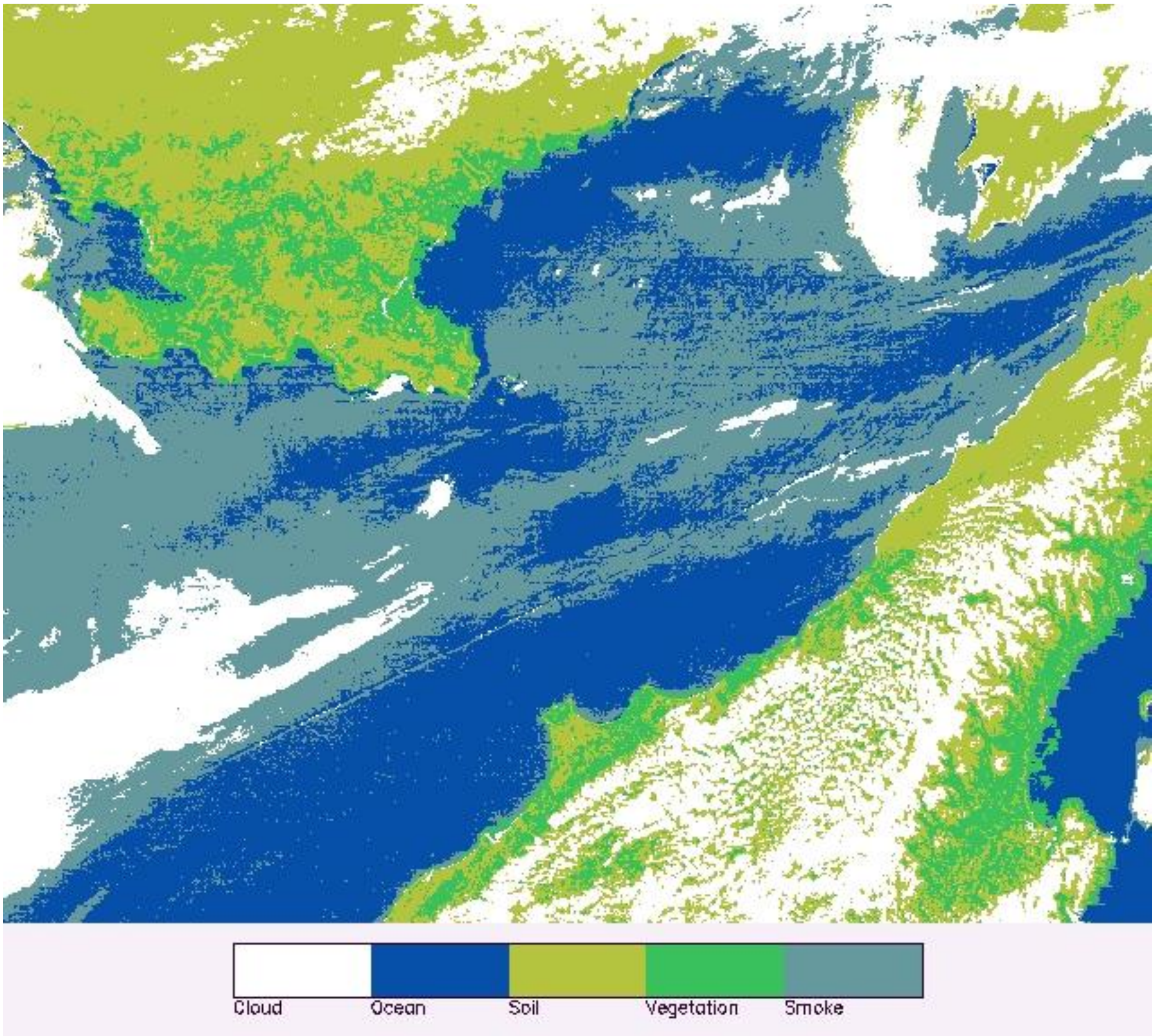

Figure 5 Thematic map of supervised classification

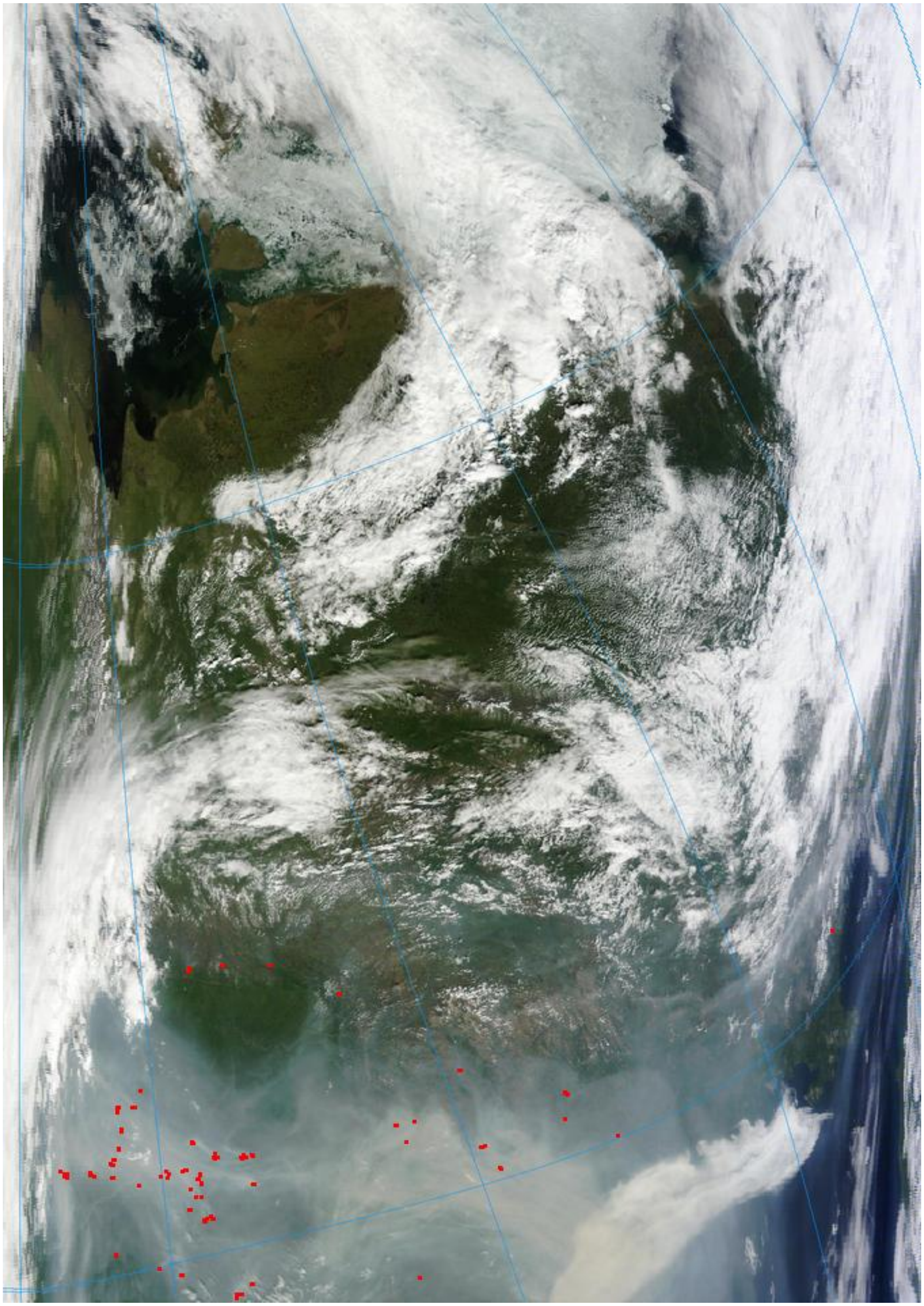

Figure 6 The west area of the study

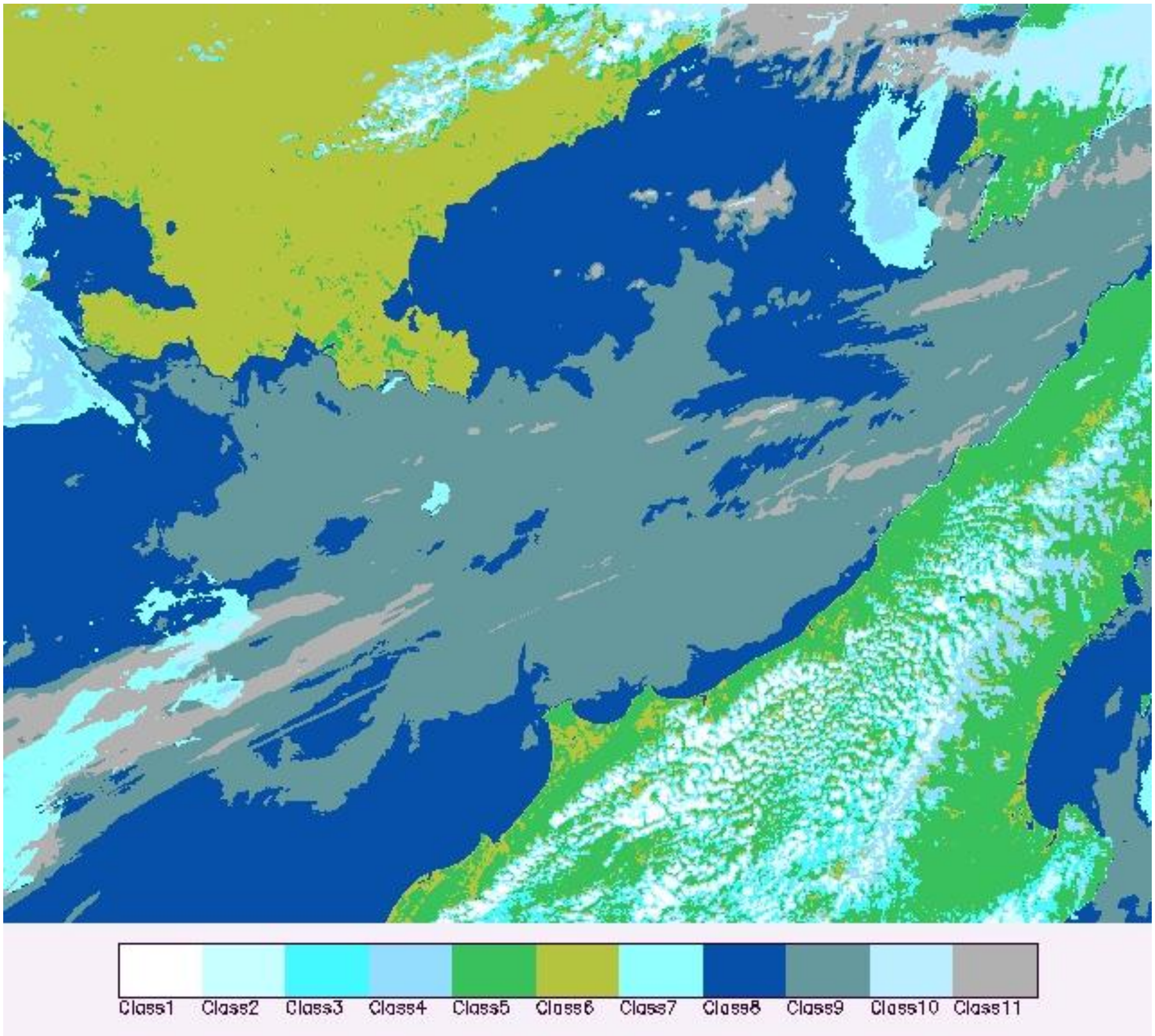

Figure 7 The initial classified result with 11 clusters

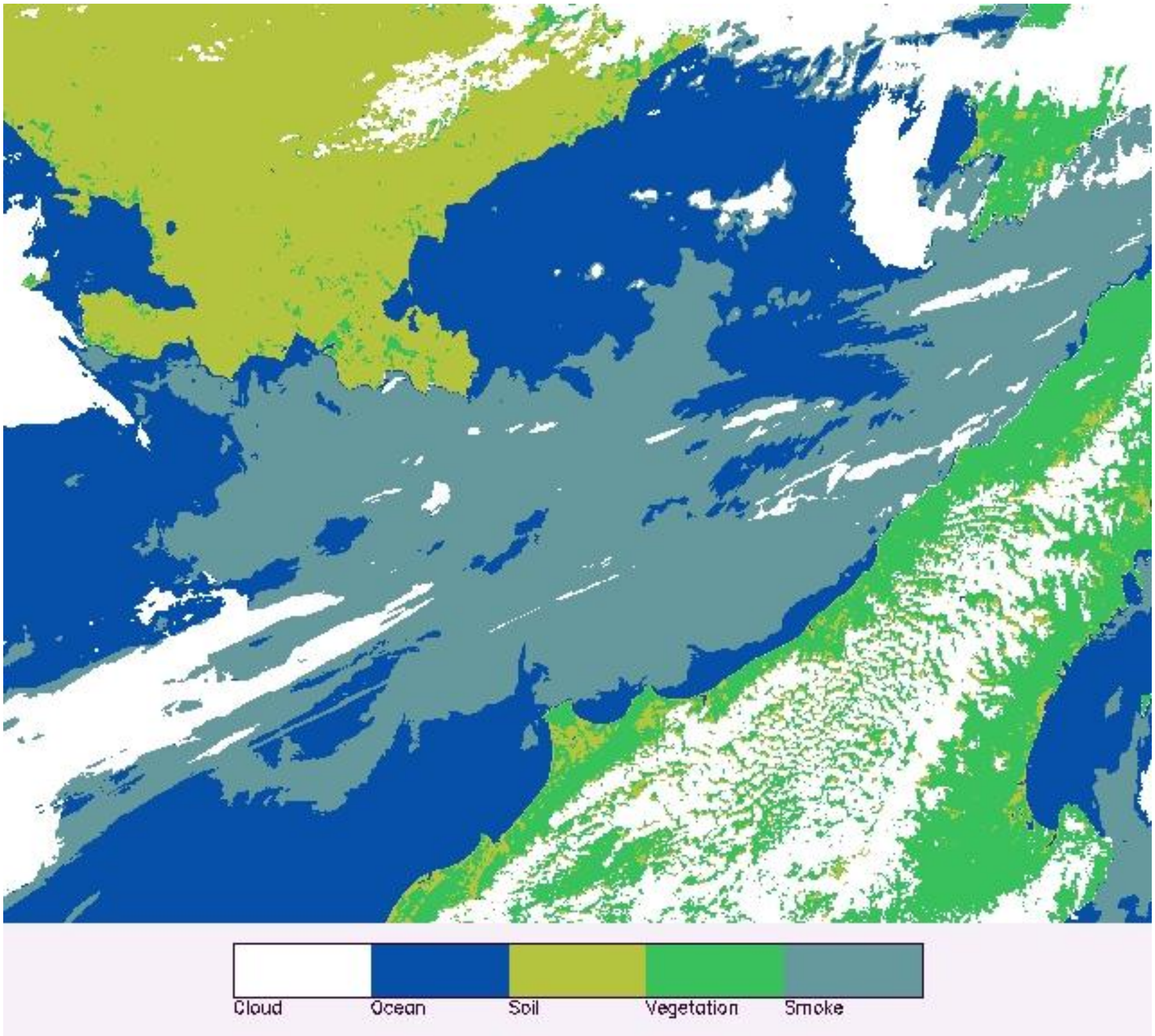

Figure 8 Final merged thematic map

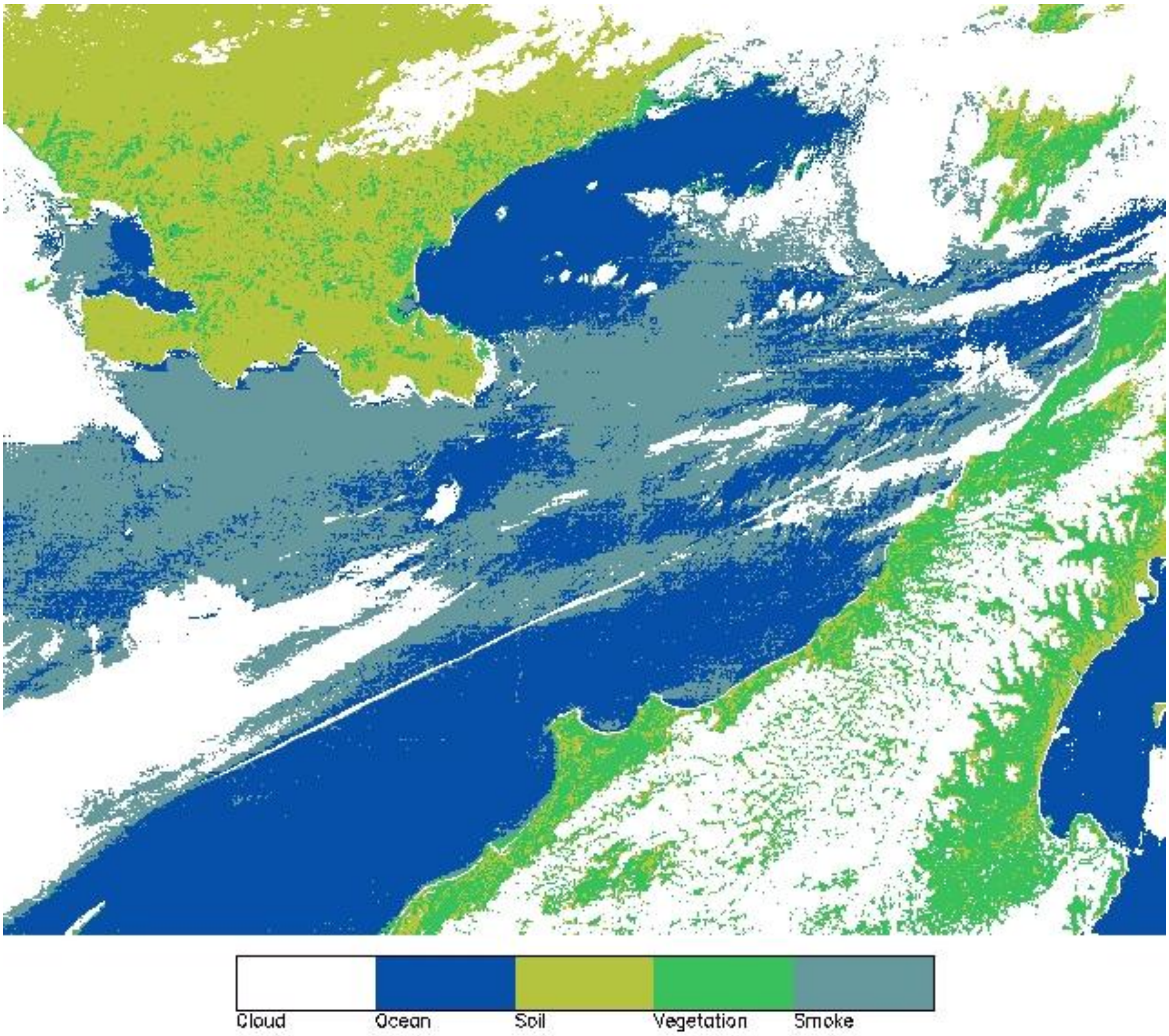

Figure 9 Thematic map of hybrid classification

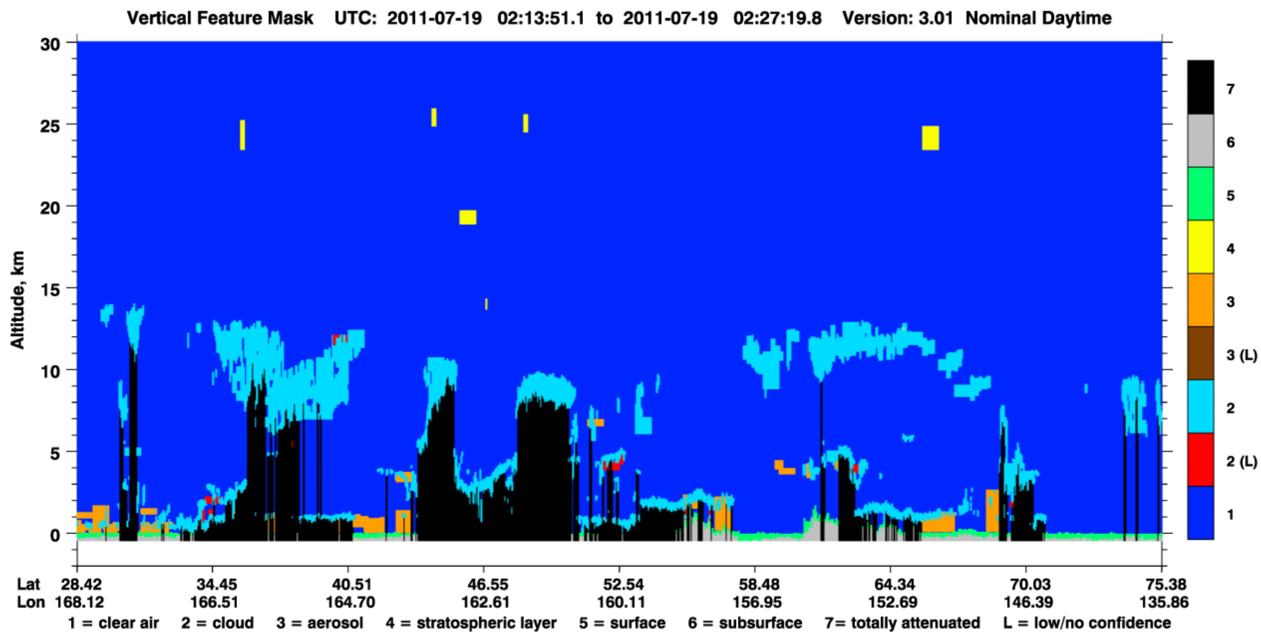

Figure 10 CALIPSO data

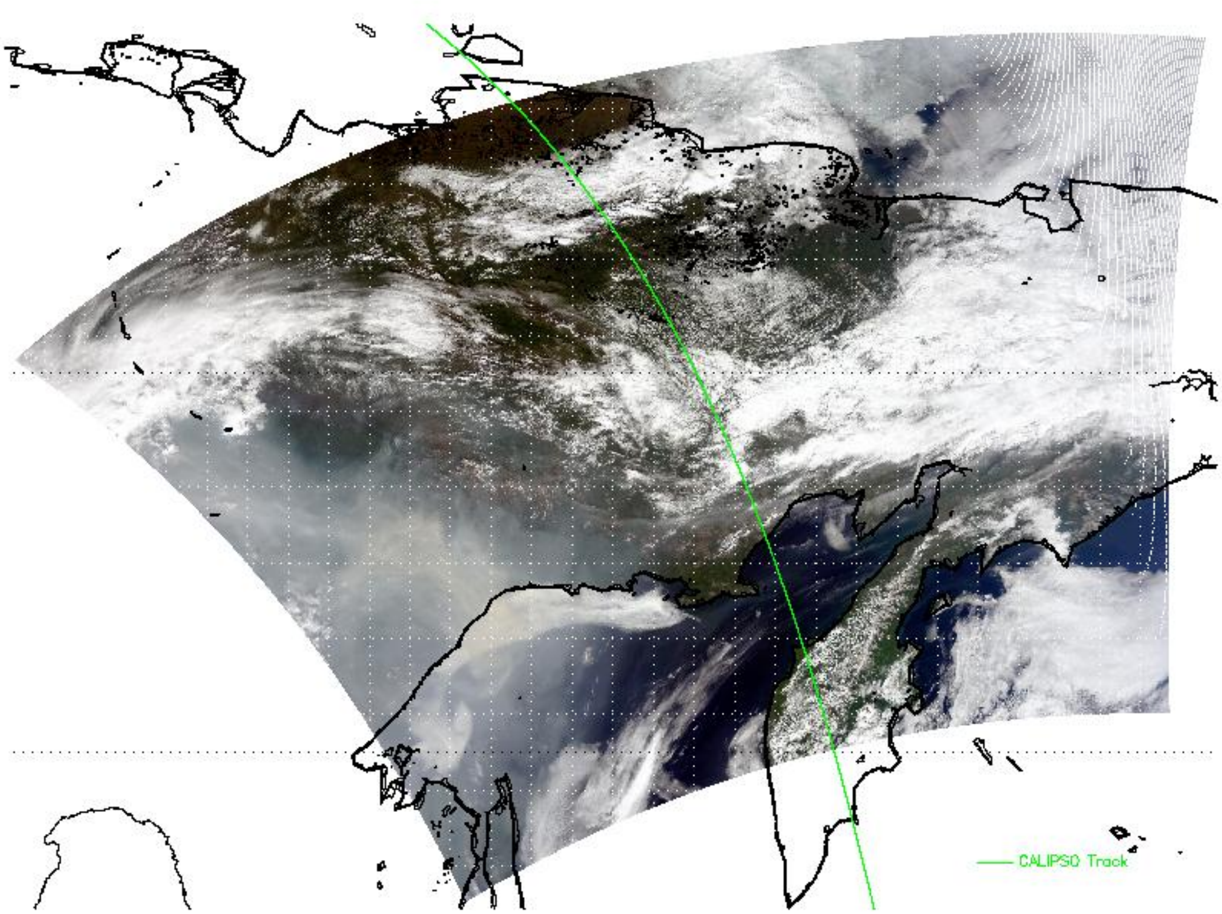

Figure 11 CALIPSO data track overplotted on the true-color image UTC 2:13-2:27, Jul 19, 2011

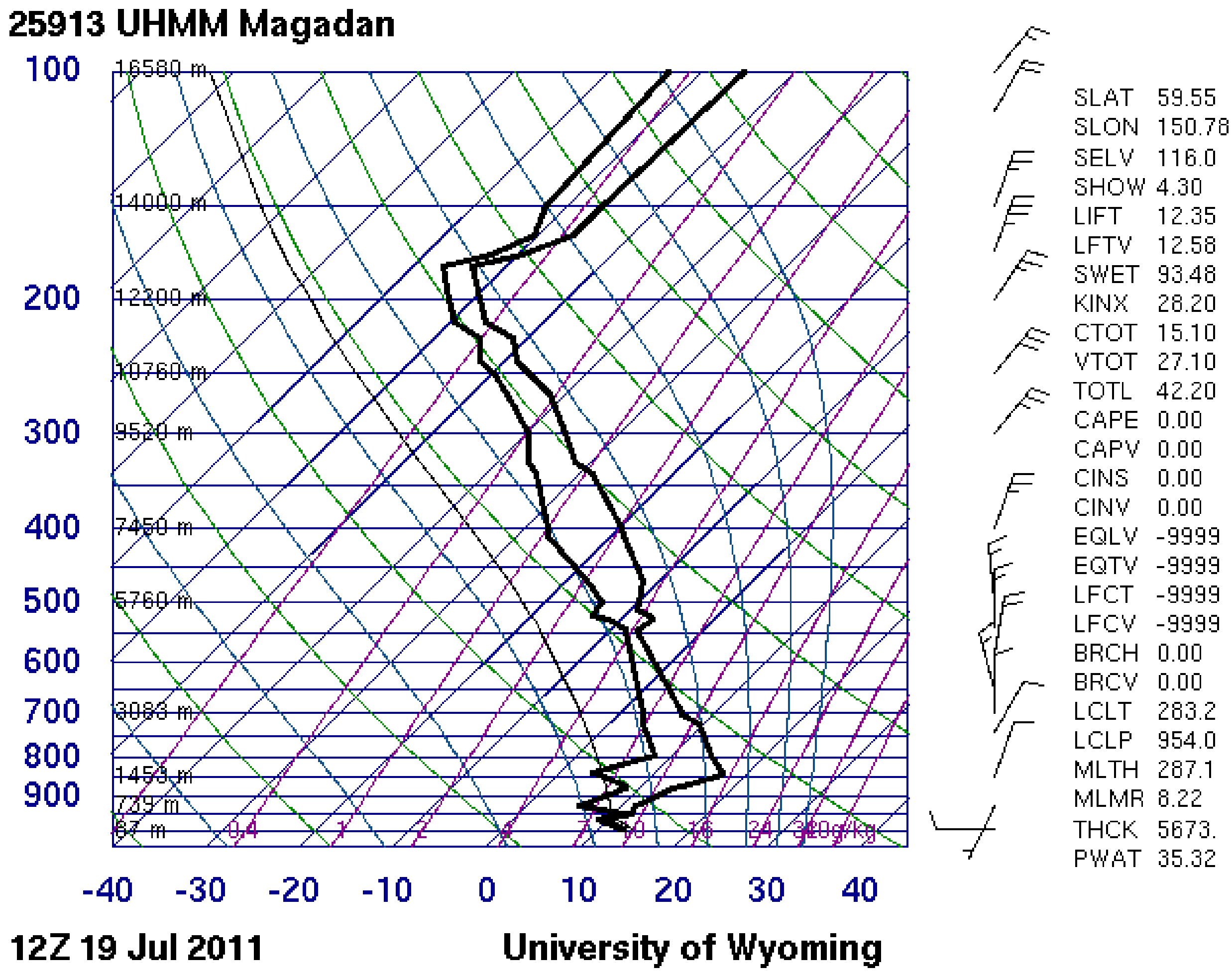

Figure 12 Sounding image

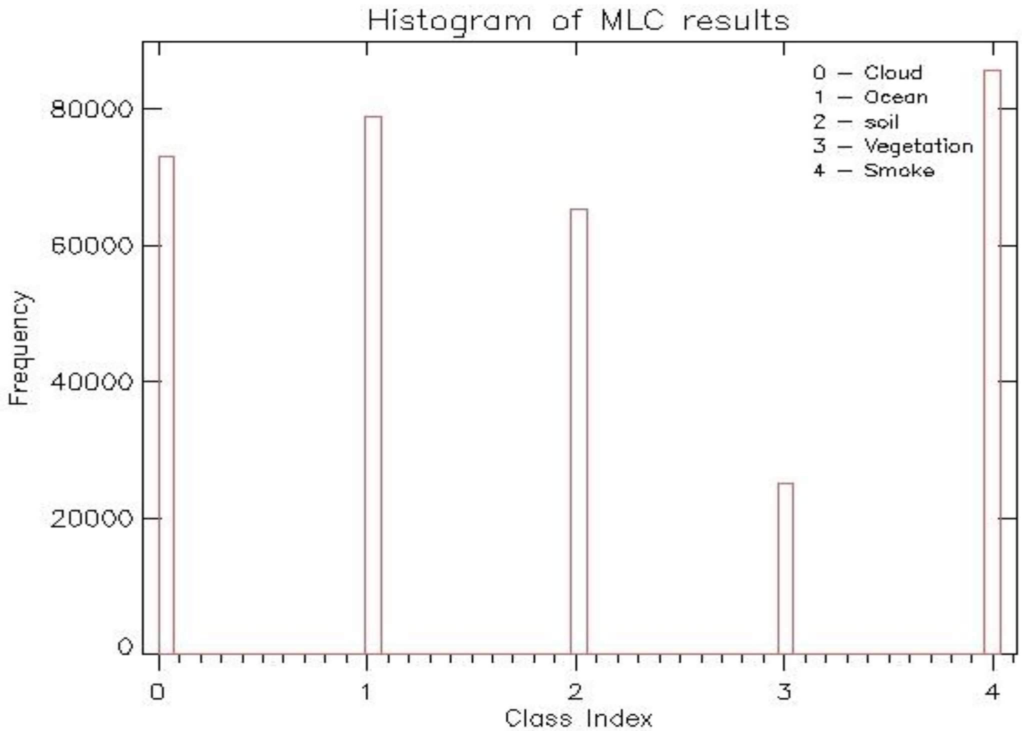

Figure 13 Histograms of MLC classification

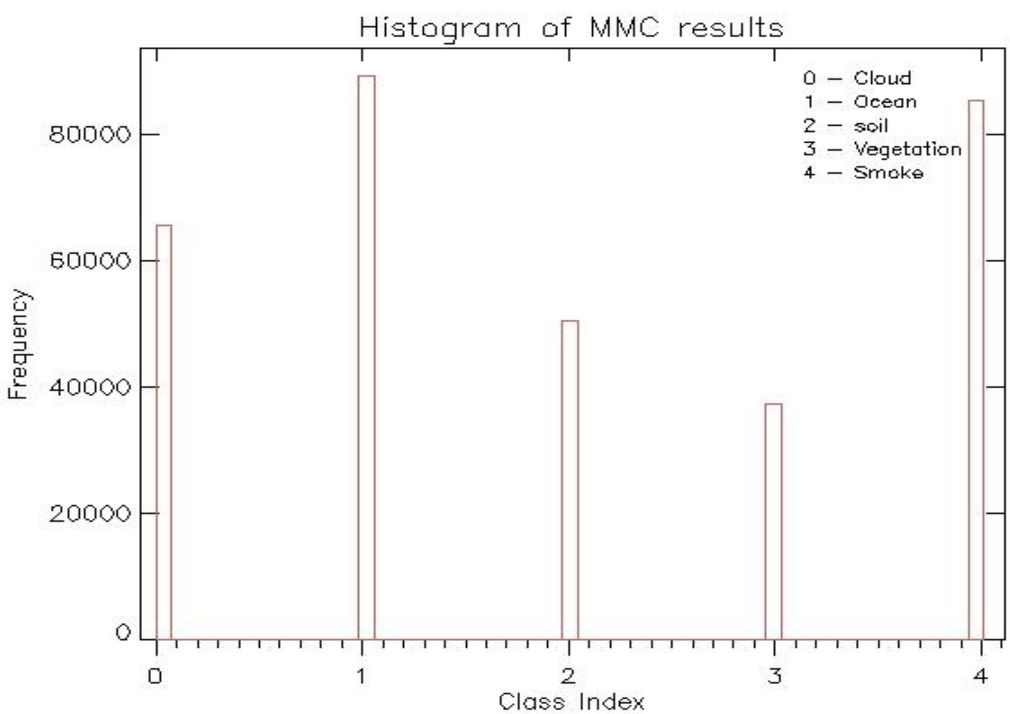

Figure 14 Histograms of MMC classification

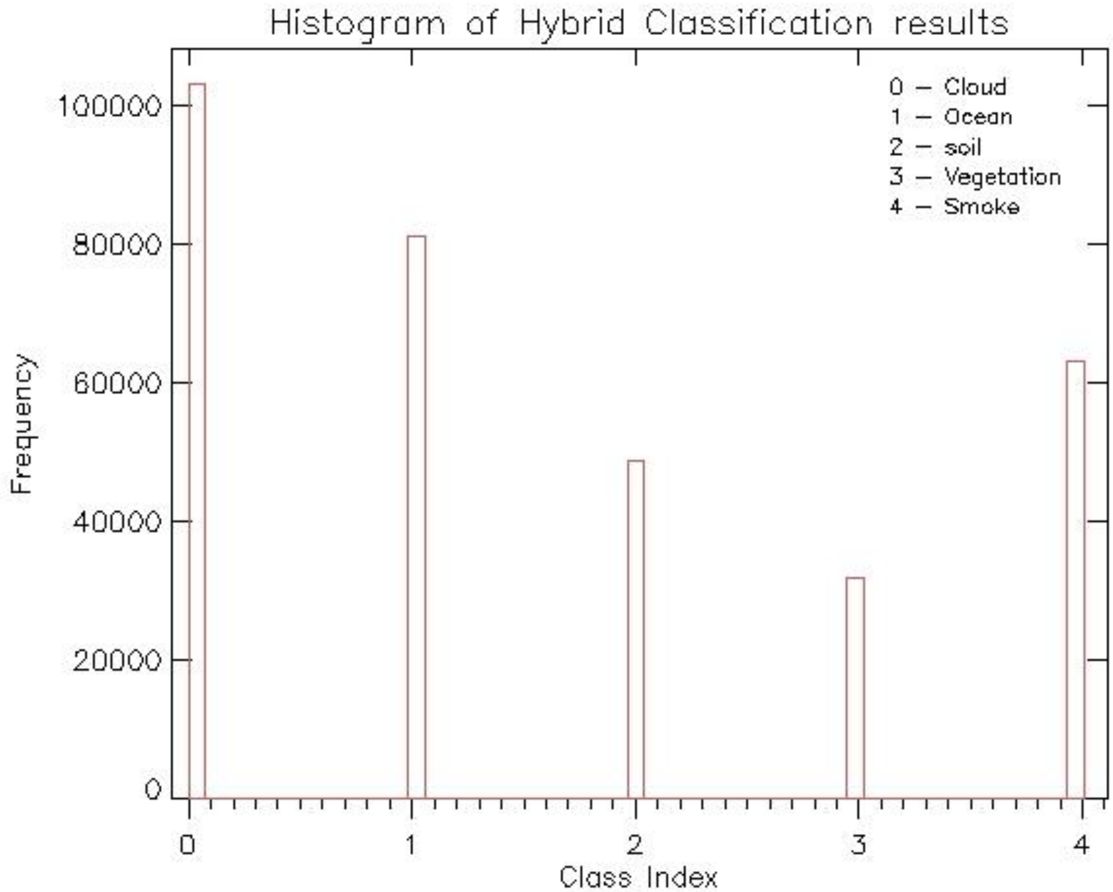

Figure 15 Histograms of hybridclassification

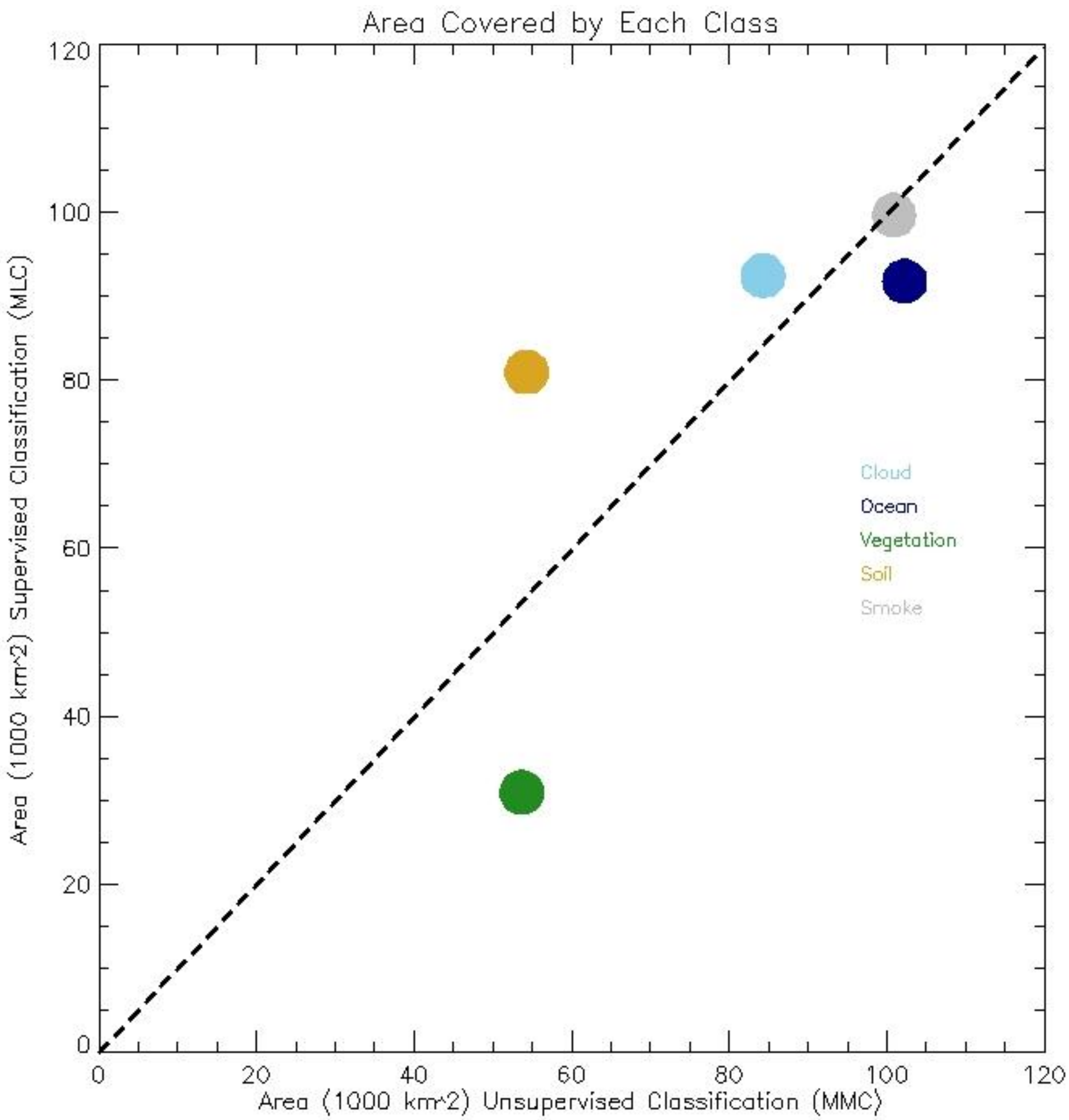

Figure 16 Area covered by each class

Appendix B

Table 1 Correlation Matrix

Band	**Mean**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	*0.10*	1.00	0.76	0.99	1.00	0.73	0.10	0.87	0.90	-0.64
2	*0.16*	0.76	1.00	0.71	0.78	0.98	0.11	0.86	0.66	-0.40
3	*0.16*	0.99	0.71	1.00	0.99	0.67	0.09	0.83	0.91	-0.65
4	*0.12*	1.00	0.78	0.99	1.00	0.74	0.10	0.87	0.90	-0.64
5	*0.15*	0.73	0.98	0.67	0.74	1.00	0.11	0.88	0.62	-0.37
6	*-0.45*	0.10	0.11	0.09	0.10	0.11	1.00	0.11	0.08	-0.05
7	*0.06*	0.87	0.86	0.83	0.87	0.88	0.11	1.00	0.77	-0.43
8	*0.20*	0.90	0.66	0.91	0.90	0.62	0.08	0.77	1.00	-0.58
9	*0.11*	-0.64	-0.40	-0.65	-0.64	-0.37	-0.05	-0.43	-0.58	1.00
10	*0.00*	-0.81	-0.57	-0.82	-0.81	-0.52	-0.07	-0.68	-0.76	0.50
11	*-0.07*	-0.82	-0.60	-0.82	-0.81	-0.55	-0.07	-0.72	-0.77	0.37
12	*-0.13*	-0.79	-0.61	-0.79	-0.79	-0.58	-0.08	-0.74	-0.75	0.28
13	*-0.29*	-0.70	-0.59	-0.69	-0.70	-0.58	-0.08	-0.76	-0.65	0.18
14	*-0.28*	-0.71	-0.59	-0.70	-0.71	-0.58	-0.08	-0.76	-0.66	0.18
15	*-0.56*	-0.52	-0.84	-0.46	-0.54	-0.86	-0.09	-0.73	-0.43	0.11
16	*-0.56*	-0.52	-0.84	-0.46	-0.54	-0.85	-0.09	-0.73	-0.42	0.11
17	*0.12*	0.49	0.76	0.44	0.51	0.75	0.08	0.66	0.48	-0.21
18	*0.06*	0.89	0.92	0.87	0.91	0.89	0.11	0.87	0.79	-0.55
19	*0.08*	0.86	0.97	0.82	0.87	0.94	0.11	0.88	0.75	-0.51
20	*290.06*	0.54	0.80	0.47	0.55	0.84	0.10	0.80	0.43	-0.17
21	*287.97*	0.29	0.64	0.21	0.30	0.69	0.08	0.58	0.18	-0.04
22	*288.13*	0.32	0.68	0.24	0.33	0.72	0.08	0.61	0.21	-0.06
23	*285.56*	0.18	0.58	0.10	0.19	0.62	0.06	0.48	0.08	0.01
24	*245.91*	-0.16	0.05	-0.23	-0.17	0.09	0.00	0.04	-0.25	0.05
25	*264.54*	-0.18	0.16	-0.26	-0.18	0.21	0.01	0.07	-0.27	0.12
26	*0.01*	0.31	0.13	0.35	0.31	0.13	0.01	0.19	0.33	-0.05
27	*243.69*	0.12	0.31	0.08	0.13	0.30	0.04	0.19	0.06	-0.18
28	*259.24*	-0.03	0.14	-0.07	-0.03	0.13	0.02	0.04	-0.08	-0.14
29	*282.42*	-0.14	0.30	-0.22	-0.13	0.34	0.03	0.13	-0.23	0.13
30	*264.09*	-0.14	0.24	-0.22	-0.14	0.29	0.04	0.12	-0.23	0.13
31	*283.73*	-0.14	0.29	-0.22	-0.14	0.32	0.03	0.12	-0.23	0.11
32	*283.07*	-0.16	0.26	-0.24	-0.15	0.29	0.02	0.09	-0.24	0.10
33	*265.88*	-0.22	0.04	-0.28	-0.22	0.07	0.01	-0.04	-0.28	0.02
34	*255.69*	-0.22	-0.05	-0.27	-0.22	-0.03	0.00	-0.08	-0.28	-0.01
35	*247.95*	-0.21	-0.08	-0.26	-0.22	-0.06	0.00	-0.09	-0.28	-0.02
36	*233.63*	-0.15	-0.07	-0.19	-0.16	-0.06	-0.02	-0.06	-0.22	-0.09

Band	10	11	12	13	14	15	16	17	18	19
1	-0.81	-0.82	-0.79	-0.70	-0.71	-0.52	-0.52	0.49	0.89	0.86
2	-0.57	-0.60	-0.61	-0.59	-0.59	-0.84	-0.84	0.76	0.92	0.97
3	-0.82	-0.82	-0.79	-0.69	-0.70	-0.46	-0.46	0.44	0.87	0.82
4	-0.81	-0.81	-0.79	-0.70	-0.71	-0.54	-0.54	0.51	0.91	0.87
5	-0.52	-0.55	-0.58	-0.58	-0.58	-0.86	-0.85	0.75	0.89	0.94
6	-0.07	-0.07	-0.08	-0.08	-0.08	-0.09	-0.09	0.08	0.11	0.11
7	-0.68	-0.72	-0.74	-0.76	-0.76	-0.73	-0.73	0.66	0.87	0.88
8	-0.76	-0.77	-0.75	-0.65	-0.66	-0.43	-0.42	0.48	0.79	0.75
9	0.50	0.37	0.28	0.18	0.18	0.11	0.11	-0.21	-0.55	-0.51
10	1.00	0.81	0.66	0.47	0.48	0.30	0.30	-0.42	-0.68	-0.64
11	0.81	1.00	0.82	0.59	0.60	0.37	0.37	-0.46	-0.69	-0.66
12	0.66	0.82	1.00	0.73	0.74	0.45	0.45	-0.48	-0.69	-0.67
13	0.47	0.59	0.73	1.00	0.98	0.63	0.63	-0.48	-0.63	-0.62
14	0.48	0.60	0.74	0.98	1.00	0.62	0.62	-0.47	-0.63	-0.62
15	0.30	0.37	0.45	0.63	0.62	1.00	0.99	-0.69	-0.67	-0.75
16	0.30	0.37	0.45	0.63	0.62	0.99	1.00	-0.69	-0.67	-0.74
17	-0.42	-0.46	-0.48	-0.48	-0.47	-0.69	-0.69	1.00	0.64	0.69
18	-0.68	-0.69	-0.69	-0.63	-0.63	-0.67	-0.67	0.64	1.00	0.99
19	-0.64	-0.66	-0.67	-0.62	-0.62	-0.75	-0.74	0.69	0.99	1.00
20	-0.35	-0.40	-0.44	-0.56	-0.55	-0.85	-0.85	0.65	0.64	0.72
21	-0.14	-0.16	-0.20	-0.36	-0.35	-0.77	-0.77	0.54	0.42	0.52
22	-0.16	-0.19	-0.23	-0.39	-0.38	-0.80	-0.80	0.57	0.45	0.55
23	-0.05	-0.06	-0.10	-0.27	-0.26	-0.73	-0.73	0.49	0.33	0.43
24	0.14	0.21	0.25	0.15	0.15	-0.21	-0.21	0.03	-0.17	-0.09
25	0.20	0.25	0.26	0.10	0.11	-0.36	-0.36	0.15	-0.11	0.00
26	-0.16	-0.26	-0.34	-0.37	-0.37	-0.15	-0.15	0.09	0.34	0.25
27	-0.19	-0.14	-0.09	0.04	0.03	-0.21	-0.21	0.23	0.16	0.23
28	-0.09	-0.02	0.06	0.19	0.19	-0.04	-0.04	0.10	-0.01	0.06
29	0.18	0.20	0.20	0.06	0.07	-0.47	-0.47	0.27	0.00	0.12
30	0.19	0.22	0.22	0.05	0.06	-0.45	-0.45	0.22	-0.04	0.07
31	0.17	0.21	0.21	0.08	0.09	-0.44	-0.44	0.26	-0.01	0.11
32	0.17	0.22	0.24	0.12	0.13	-0.40	-0.40	0.23	-0.04	0.08
33	0.13	0.22	0.29	0.27	0.27	-0.11	-0.11	0.03	-0.19	-0.10
34	0.11	0.21	0.30	0.30	0.30	0.01	0.01	-0.06	-0.25	-0.17
35	0.10	0.20	0.29	0.30	0.30	0.04	0.04	-0.08	-0.26	-0.19
36	0.02	0.13	0.22	0.28	0.27	0.09	0.09	-0.09	-0.20	-0.14

Band	20	21	22	23	24	25	26	27	28	29
1	0.54	0.29	0.32	0.18	-0.16	-0.18	0.31	0.12	-0.03	-0.14
2	0.80	0.64	0.68	0.58	0.05	0.16	0.13	0.31	0.14	0.30
3	0.47	0.21	0.24	0.10	-0.23	-0.26	0.35	0.08	-0.07	-0.22
4	0.55	0.30	0.33	0.19	-0.17	-0.18	0.31	0.13	-0.03	-0.13
5	0.84	0.69	0.72	0.62	0.09	0.21	0.13	0.30	0.13	0.34
6	0.10	0.08	0.08	0.06	0.00	0.01	0.01	0.04	0.02	0.03
7	0.80	0.58	0.61	0.48	0.04	0.07	0.19	0.19	0.04	0.13
8	0.43	0.18	0.21	0.08	-0.25	-0.27	0.33	0.06	-0.08	-0.23
9	-0.17	-0.04	-0.06	0.01	0.05	0.12	-0.05	-0.18	-0.14	0.13
10	-0.35	-0.14	-0.16	-0.05	0.14	0.20	-0.16	-0.19	-0.09	0.18
11	-0.40	-0.16	-0.19	-0.06	0.21	0.25	-0.26	-0.14	-0.02	0.20
12	-0.44	-0.20	-0.23	-0.10	0.25	0.26	-0.34	-0.09	0.06	0.20
13	-0.56	-0.36	-0.39	-0.27	0.15	0.10	-0.37	0.04	0.19	0.06
14	-0.55	-0.35	-0.38	-0.26	0.15	0.11	-0.37	0.03	0.19	0.07
15	-0.85	-0.77	-0.80	-0.73	-0.21	-0.36	-0.15	-0.21	-0.04	-0.47
16	-0.85	-0.77	-0.80	-0.73	-0.21	-0.36	-0.15	-0.21	-0.04	-0.47
17	0.65	0.54	0.57	0.49	0.03	0.15	0.09	0.23	0.10	0.27
18	0.64	0.42	0.45	0.33	-0.17	-0.11	0.34	0.16	-0.01	0.00
19	0.72	0.52	0.55	0.43	-0.09	0.00	0.25	0.23	0.06	0.12
20	1.00	0.93	0.95	0.89	0.46	0.58	-0.12	0.39	0.27	0.65
21	0.93	1.00	0.98	0.97	0.62	0.76	-0.27	0.41	0.33	0.83
22	0.95	0.98	1.00	0.98	0.61	0.76	-0.25	0.42	0.33	0.83
23	0.89	0.97	0.98	1.00	0.70	0.85	-0.33	0.42	0.36	0.90
24	0.46	0.62	0.61	0.70	1.00	0.94	-0.65	0.42	0.52	0.81
25	0.58	0.76	0.76	0.85	0.94	1.00	-0.63	0.45	0.51	0.95
26	-0.12	-0.27	-0.25	-0.33	-0.65	-0.63	1.00	-0.57	-0.77	-0.61
27	0.39	0.41	0.42	0.42	0.42	0.45	-0.57	1.00	0.93	0.54
28	0.27	0.33	0.33	0.36	0.52	0.51	-0.77	0.93	1.00	0.56
29	0.65	0.83	0.83	0.90	0.81	0.95	-0.61	0.54	0.56	1.00
30	0.64	0.83	0.82	0.90	0.89	0.98	-0.54	0.42	0.45	0.97
31	0.64	0.82	0.82	0.89	0.82	0.95	-0.64	0.55	0.58	1.00
32	0.60	0.79	0.79	0.86	0.82	0.95	-0.68	0.57	0.62	0.99
33	0.38	0.55	0.54	0.63	0.88	0.88	-0.88	0.66	0.79	0.85
34	0.28	0.44	0.43	0.51	0.89	0.83	-0.88	0.61	0.76	0.73
35	0.25	0.41	0.39	0.48	0.89	0.80	-0.86	0.57	0.73	0.69
36	0.21	0.32	0.31	0.38	0.81	0.69	-0.81	0.58	0.73	0.57

Band	30	31	32	33	34	35	36
1	-0.14	-0.14	-0.16	-0.22	-0.22	-0.21	-0.15
2	0.24	0.29	0.26	0.04	-0.05	-0.08	-0.07
3	-0.22	-0.22	-0.24	-0.28	-0.27	-0.26	-0.19
4	-0.14	-0.14	-0.15	-0.22	-0.22	-0.22	-0.16
5	0.29	0.32	0.29	0.07	-0.03	-0.06	-0.06
6	0.04	0.03	0.02	0.01	0.00	0.00	-0.02
7	0.12	0.12	0.09	-0.04	-0.08	-0.09	-0.06
8	-0.23	-0.23	-0.24	-0.28	-0.28	-0.28	-0.22
9	0.13	0.11	0.10	0.02	-0.01	-0.02	-0.09
10	0.19	0.17	0.17	0.13	0.11	0.10	0.02
11	0.22	0.21	0.22	0.22	0.21	0.20	0.13
12	0.22	0.21	0.24	0.29	0.30	0.29	0.22
13	0.05	0.08	0.12	0.27	0.30	0.30	0.28
14	0.06	0.09	0.13	0.27	0.30	0.30	0.27
15	-0.45	-0.44	-0.40	-0.11	0.01	0.04	0.09
16	-0.45	-0.44	-0.40	-0.11	0.01	0.04	0.09
17	0.22	0.26	0.23	0.03	-0.06	-0.08	-0.09
18	-0.04	-0.01	-0.04	-0.19	-0.25	-0.26	-0.20
19	0.07	0.11	0.08	-0.10	-0.17	-0.19	-0.14
20	0.64	0.64	0.60	0.38	0.28	0.25	0.21
21	0.83	0.82	0.79	0.55	0.44	0.41	0.32
22	0.82	0.82	0.79	0.54	0.43	0.39	0.31
23	0.90	0.89	0.86	0.63	0.51	0.48	0.38
24	0.89	0.82	0.82	0.88	0.89	0.89	0.81
25	0.98	0.95	0.95	0.88	0.83	0.80	0.69
26	-0.54	-0.64	-0.68	-0.88	-0.88	-0.86	-0.81
27	0.42	0.55	0.57	0.66	0.61	0.57	0.58
28	0.45	0.58	0.62	0.79	0.76	0.73	0.73
29	0.97	1.00	0.99	0.85	0.73	0.69	0.57
30	1.00	0.96	0.95	0.82	0.74	0.71	0.59
31	0.96	1.00	1.00	0.87	0.76	0.72	0.60
32	0.95	1.00	1.00	0.90	0.79	0.75	0.63
33	0.82	0.87	0.90	1.00	0.97	0.95	0.87
34	0.74	0.76	0.79	0.97	1.00	0.99	0.93
35	0.71	0.72	0.75	0.95	0.99	1.00	0.94
36	0.59	0.60	0.63	0.87	0.93	0.94	1.00

Table 2 Statistical data of all bands and selected bands

Band	Mean	Maximum	Minimum	Number of Negatives	Negatives %
1	0.0972052	1.05966	0.0236627	0	0
2	0.161205	0.886927	0.00919296	0	0
3	0.158907	1.03398	0.0890108	0	0
4	0.122164	1.04255	0.0449823	0	0
5	0.151871	0.932241	0.00279129	0	0
6	-0.446501	0.711457	-1.35634	130560	39.8438
7	0.062143	0.556089	0.000496993	0	0
8	0.195549	0.809068	-1.32441	95	0.0289917
9	0.109547	0.445791	-1.32642	12135	3.70331
10	-0.00236096	0.290779	-1.35699	29088	8.87695
11	-0.0678052	0.221532	-1.35699	39321	11.9998
12	-0.1318	0.173427	-1.35699	52549	16.0367
13	-0.287891	0.101061	-1.35699	82628	25.2161
14	-0.279633	0.103536	-1.35699	80530	24.5758
15	-0.555807	0.0951416	-1.35699	146223	44.6237
16	-0.564032	0.0871442	-1.35699	147223	44.9289
17	0.123243	0.749687	-1.32441	1005	0.306702
18	0.0587429	0.701953	0.00440805	0	0
19	0.0835167	0.816782	0.00512074	0	0
20	290.018	338.383	262.029	0	0
21	287.998	331.436	240.284	0	0
22	288.067	329.591	-1	1	0.00030518
23	285.484	325.881	249.356	0	0
24	245.926	253.999	231.274	0	0
25	264.556	283.41	233.705	0	0
26	0.00654864	0.1958	-0.00647087	20647	6.30096
27	243.636	252.576	223.154	0	0
28	259.213	265.17	229.434	0	0
29	282.375	304.781	236.02	0	0
30	264.146	282.508	235.598	0	0
31	283.663	307.043	232.66	0	0
32	282.976	304.922	231.56	0	0
33	265.735	274.794	229.162	0	0
34	255.871	261.399	227.981	0	0
35	248.206	252.238	227.706	0	0
36	233.607	235.537	225.479	0	0

Table 3 MLC training sample Covariance Matrix

Class	Mean Vector	Covariance Matrix						
Cloud	0.4142	0.0071	0.0090	0.0066	0.0090	0.1159	0.1096	0.1212
	0.4744	0.0090	0.0125	0.0082	0.0124	0.2561	0.0967	0.1001
	0.4403	0.0066	0.0082	0.0062	0.0084	0.0810	0.0911	0.1042
	0.4495	0.0090	0.0124	0.0084	0.0134	0.2186	0.0266	0.0309
	301.1000	0.1159	0.2561	0.0810	0.2186	18.4781	1.5301	0.1121
	279.7670	0.1096	0.0967	0.0911	0.0266	1.5301	13.5551	14.3043
	279.2620	0.1212	0.1001	0.1042	0.0309	0.1121	14.3043	15.3461
Ocean	0.0332	0.0000	0.0000	0.0000	0.0000	0.0054	0.0048	0.0050
	0.0165	0.0000	0.0000	0.0000	0.0000	0.0036	0.0028	0.0031
	0.1040	0.0000	0.0000	0.0001	0.0000	0.0059	0.0066	0.0067
	0.0081	0.0000	0.0000	0.0000	0.0000	0.0012	0.0004	0.0006
	283.3460	0.0054	0.0036	0.0059	0.0012	15.1279	14.9434	14.2778
	282.7630	0.0048	0.0028	0.0066	0.0004	14.9434	14.9398	14.2703
	282.4820	0.0050	0.0031	0.0067	0.0006	14.2778	14.2703	13.6448
Soil	0.0983	0.0003	-0.0006	0.0003	-0.0006	0.0229	0.0057	0.0058
	0.2342	-0.0006	0.0028	-0.0007	0.0021	-0.0720	-0.0238	-0.0243
	0.1321	0.0003	-0.0007	0.0003	-0.0006	0.0235	0.0062	0.0060
	0.2495	-0.0006	0.0021	-0.0006	0.0018	-0.0632	-0.0189	-0.0194
	302.8690	0.0229	-0.0720	0.0235	-0.0632	2.6793	1.0193	0.9687
	296.9920	0.0057	-0.0238	0.0062	-0.0189	1.0193	0.5381	0.5027
	295.9380	0.0058	-0.0243	0.0060	-0.0194	0.9687	0.5027	0.4938
Vegetation	0.0526	0.0000	0.0006	0.0000	0.0006	0.0159	0.0114	0.0106
	0.2804	0.0006	0.0222	0.0002	0.0203	0.4283	0.2988	0.2812
	0.1057	0.0000	0.0002	0.0000	0.0002	-0.0029	-0.0027	-0.0023
	0.2613	0.0006	0.0203	0.0002	0.0187	0.4229	0.2976	0.2794
	295.6540	0.0159	0.4283	-0.0029	0.4229	14.7496	10.8472	10.0950
	292.6530	0.0114	0.2988	-0.0027	0.2976	10.8472	8.0252	7.4627
	291.8770	0.0106	0.2812	-0.0023	0.2794	10.0950	7.4627	6.9427
Smoke	0.0550	0.0001	0.0001	0.0001	0.0000	0.0122	0.0122	0.0118
	0.0346	0.0001	0.0000	0.0001	0.0000	0.0087	0.0076	0.0073
	0.1314	0.0001	0.0001	0.0002	0.0000	0.0165	0.0174	0.0161
	0.0202	0.0000	0.0000	0.0000	0.0000	0.0042	0.0021	0.0020
	284.3160	0.0122	0.0087	0.0165	0.0042	5.5261	4.9235	4.6838
	282.3750	0.0122	0.0076	0.0174	0.0021	4.9235	5.2879	5.1356
	282.0860	0.0118	0.0073	0.0161	0.0020	4.6838	5.1356	5.0442

Table 4 MLC mean & std for trainning samples

MLC mean & std for training samples														
Band	1		2		3		5		20		31		32	
Class	µ	s	µ	s	µ	s	µ	s	µ	s	µ	s	µ	s
Cloud	0.414	0.152	0.474	0.154	0.440	0.137	0.450	0.154	301.100	4.768	279.767	5.326	279.262	5.341
Ocean	0.033	0.004	0.017	0.004	0.104	0.007	0.008	0.003	283.346	3.725	282.763	3.697	282.482	3.532
Soil	0.098	0.024	0.234	0.064	0.132	0.019	0.250	0.054	302.869	2.581	296.992	1.665	295.938	1.606
Vegetation	0.053	0.007	0.280	0.145	0.106	0.005	0.261	0.133	295.654	3.999	292.653	2.977	291.877	2.770
Smoke	0.055	0.008	0.035	0.006	0.131	0.013	0.020	0.004	284.316	2.264	282.375	2.217	282.086	2.167

Table 5 MLC mean & std for classified data

MLC mean & std for classified data														
Band	1		2		3		5		20		31		32	
Class	µ	s	µ	s	µ	s	µ	s	µ	s	µ	s	µ	s
Cloud	0.238	0.164	0.314	0.206	0.283	0.144	0.298	0.201	296.023	9.383	278.436	7.872	277.388	7.942
Ocean	0.036	0.006	0.018	0.006	0.109	0.010	0.009	0.004	282.451	3.422	281.616	3.484	281.349	3.329
Soil	0.094	0.036	0.290	0.086	0.144	0.035	0.290	0.070	299.898	3.955	293.498	4.081	292.350	3.915
Vegetation	0.057	0.009	0.276	0.132	0.114	0.010	0.264	0.124	295.235	4.765	291.876	3.762	291.148	3.518
Smoke	0.048	0.009	0.031	0.009	0.123	0.012	0.020	0.008	282.809	2.680	280.091	3.081	279.707	3.119
	table5													

Table 6 Areas of MLC

Class	Number of pixels	Area
Cloud	73043	92435.4
Ocean	78766	91744.9
Soil	65229	80950
Vegetation	25076	30888.2
Smoke	85566	99647.7

Table 7 MLC Confusion Matrix

Thematic Map	Ground Truth					
	Cloud	Ocean	Soil	Vegetation	Smoke	User Accuracy
Cloud	20	0	0	0	0	100.00%
Ocean	0	16	0	0	2	88.89%
Soil	0	1	19	4	0	79.17%
Vegetation	0	0	1	16	0	79.17%
Smoke	0	3	0	0	18	85.71%
Producer Accuracy	100.00%	80.00%	95.00%	80.00%	90.00%	89.00%

Table 8 MMC mean & std for classified data

MMC mean & std for classified data														
Band	1		2		3		5		20		31		32	
Class	µ	s	µ	s	µ	s	µ	s	µ	s	µ	s	µ	s
Cloud	0.256	0.164	0.336	0.205	0.301	0.143	0.314	0.200	295.938	9.617	277.495	7.698	276.487	7.875
Ocean	0.043	0.013	0.026	0.016	0.116	0.014	0.016	0.014	285.232	2.269	283.553	2.053	283.176	2.003
Soil	0.085	0.028	0.240	0.042	0.133	0.026	0.253	0.039	301.024	3.709	295.326	2.744	294.088	2.641
Vegetation	0.086	0.032	0.376	0.055	0.140	0.029	0.361	0.043	297.606	3.147	291.050	3.139	290.067	3.224
Smoke	0.044	0.010	0.027	0.011	0.118	0.013	0.017	0.010	280.619	2.357	278.346	2.014	278.073	2.030

Table 9 Areas of MMC

Class	Number of pixels	Area
Cloud	65624	84351.2
Ocean	89082	102321
Soil	50476	54300.8
Vegetation	37339	53743.7
Smoke	85159	100950

Table 10 MMC Confusion Matrix

Thematic Map	Ground Truth					
	Cloud	Ocean	Soil	Vegetation	Smoke	User Accuracy
Cloud	19	0	0	0	0	100.00%
Ocean	0	13	0	1	11	52.00%
Soil	0	0	17	5	0	77.27%
Vegetation	1	0	3	14	0	77.78%
Smoke	0	7	0	0	9	56.25%
Producer Accuracy	95.00%	65.00%	85.00%	70.00%	45.00%	72.00%

Table 11 Hybrid Training Samples Covariance Matrix

Class	Mean Vector	Covariance Matrix						
Cloud	0.352	0.0059	0.0056	0.0053	0.0050	0.0963	0.0379	0.0585
	0.441	0.0056	0.0059	0.0050	0.0052	0.0963	0.0387	0.0590
	0.386	0.0053	0.0050	0.0047	0.0045	0.0867	0.0341	0.0529
	0.421	0.0050	0.0052	0.0045	0.0047	0.0990	0.0389	0.0571
	300.437	0.0963	0.0963	0.0867	0.0990	8.7584	6.6382	7.1540
	278.542	0.0379	0.0387	0.0341	0.0389	6.6382	6.2753	6.5936
	277.628	0.0585	0.0590	0.0529	0.0571	7.1540	6.5936	7.0075
Ocean	0.034	0.0000	0.0000	0.0000	0.0000	-0.0005	-0.0021	-0.0020
	0.016	0.0000	0.0000	0.0000	0.0000	-0.0008	-0.0025	-0.0024
	0.107	0.0000	0.0000	0.0000	0.0000	-0.0005	-0.0017	-0.0016
	0.008	0.0000	0.0000	0.0000	0.0000	-0.0015	-0.0033	-0.0031
	283.236	-0.0005	-0.0008	-0.0005	-0.0015	3.3008	3.4391	3.2600
	282.628	-0.0021	-0.0025	-0.0017	-0.0033	3.4391	3.8268	3.6231
	282.334	-0.0020	-0.0024	-0.0016	-0.0031	3.2600	3.6231	3.4304
Soil	0.095	0.0002	-0.0003	0.0001	-0.0002	0.0290	0.0164	0.0147
	0.210	-0.0003	0.0008	-0.0002	0.0006	-0.0628	-0.0366	-0.0331
	0.135	0.0001	-0.0002	0.0001	-0.0001	0.0187	0.0101	0.0087
	0.236	-0.0002	0.0006	-0.0001	0.0004	-0.0434	-0.0259	-0.0238
	304.087	0.0290	-0.0628	0.0187	-0.0434	5.4655	3.2883	2.9617
	297.741	0.0164	-0.0366	0.0101	-0.0259	3.2883	2.1386	1.9748
	296.435	0.0147	-0.0331	0.0087	-0.0238	2.9617	1.9748	1.8459
Vegetation	0.068	0.0001	0.0000	0.0001	0.0000	0.0033	-0.0036	-0.0036
	0.346	0.0000	0.0007	0.0000	0.0004	-0.0017	-0.0041	-0.0036
	0.119	0.0001	0.0000	0.0001	0.0000	0.0027	-0.0039	-0.0038
	0.336	0.0000	0.0004	0.0000	0.0003	0.0005	-0.0026	-0.0024
	298.250	0.0033	-0.0017	0.0027	0.0005	0.5949	0.3557	0.3481
	293.519	-0.0036	-0.0041	-0.0039	-0.0026	0.3557	0.8386	0.8261
	292.573	-0.0036	-0.0036	-0.0038	-0.0024	0.3481	0.8261	0.8146
Smoke	0.048	0.0000	0.0000	0.0000	0.0000	0.0010	0.0005	0.0004
	0.030	0.0000	0.0000	0.0000	0.0000	0.0011	0.0003	0.0002
	0.122	0.0000	0.0000	0.0000	0.0000	0.0007	0.0005	0.0004
	0.018	0.0000	0.0000	0.0000	0.0000	0.0012	0.0002	0.0001
	281.755	0.0010	0.0011	0.0007	0.0012	0.8949	0.4485	0.3993
	280.101	0.0005	0.0003	0.0005	0.0002	0.4485	0.3820	0.3729
	279.938	0.0004	0.0002	0.0004	0.0001	0.3993	0.3729	0.3697

Table 12 Hybrid mean & std for trainning samples

Hybrid mean & std for trainning samples														
Band	1		2		3		5		20		31		32	
Class	μ	s	μ	s	μ	s	μ	s	μ	s	μ	s	μ	s
Cloud	0.352	0.077	0.441	0.077	0.386	0.069	0.421	0.069	300.437	2.959	278.542	2.505	277.628	2.647
Ocean	0.034	0.003	0.016	0.004	0.107	0.003	0.008	0.004	283.236	1.817	282.628	1.956	282.334	1.852
Soil	0.095	0.015	0.210	0.028	0.135	0.010	0.236	0.021	304.087	2.338	297.741	1.462	296.435	1.359
Vegetation	0.068	0.009	0.346	0.026	0.119	0.008	0.336	0.017	298.250	0.771	293.519	0.916	292.573	0.903
Smoke	0.048	0.002	0.030	0.002	0.122	0.004	0.018	0.002	281.755	0.946	280.101	0.618	279.938	0.608

Table 13 Hybrid mean & std for classified data

Hybrid mean & std for classified data														
Band	1		2		3		5		20		31		32	
Class	μ	s	μ	s	μ	s	μ	s	μ	s	μ	s	μ	s
Cloud	0.190	0.158	0.257	0.210	0.244	0.137	0.238	0.204	293.120	9.720	279.232	7.362	278.311	7.452
Ocean	0.036	0.006	0.018	0.006	0.109	0.010	0.010	0.005	283.114	3.289	282.193	3.278	281.886	3.133
Soil	0.084	0.027	0.251	0.054	0.132	0.023	0.264	0.047	300.620	3.991	295.073	3.050	293.859	2.898
Vegetation	0.076	0.031	0.340	0.106	0.130	0.027	0.331	0.097	297.149	3.791	291.734	3.440	290.847	3.375
Smoke	0.046	0.006	0.028	0.006	0.119	0.009	0.018	0.005	282.060	2.582	279.908	2.432	279.640	2.412

Table 14 Hybrid Areas

Class	Number of pixels	Area
Cloud	103041	132748
Ocean	81130	94702.3
Soil	48628	54300
Vegetation	31852	43598.6
Smoke	63029	70317.1

Table 15 Hybrid Confusion Matrix

Thematic Map	Ground Truth					
	Cloud	Ocean	Soil	Vegetation	Smoke	User Accuracy
Cloud	20	0	0	0	4	83.33%
Ocean	0	18	0	0	0	100.00%
Soil	0	0	19	6	0	73.08%
Vegetation	0	1	1	14	0	86.67%
Smoke	0	1	0	0	16	94.12%
Producer Accuracy	100.00%	90.00%	95.00%	70.00%	80.00%	87.00%

Appendix C

The procedure to read in any modis data

```
; docformat = 'rst'
;+
;
; The procedure to read in any MODIS data and store 36 bands in .sav file
;
; :Private:
;
; :Categories:
; Remote Sensing/ATS670
;
; :Examples:
; shay_modis, 'MYD021KM.A2010109.1305.005.2010111050209.hdf'
;
; :Uses:
; modis_l1b_read
;
; :Author:
; Xuechang "Shay" Liu, UAH, xuechang at nsstc.uah.edu
;
; :History:
; First written: Feb, 2013
;
;-
;
Pro shay_modis, filename,Save=save, Display=display
;+
; The main procedure to read in any MODIS data and store 36 bands in .sav
file
; Option to display 36 bands on screen every 1 seoncd
;
;
; :Params:
; filename : in, required, type=string
;
; :Keywords:
; Save : in, optional, type=boolean
; Option to save 36 bands as a .sav file
;  Display: in, optional, type=boolean
; Option to display 36 bands on screen every 1 seoncd
;
; :Uses:
; modis_l1b_read
;
;-

compile_opt idl2

;error catcher:
CATCH, theError
IF theError NE 0 THEN BEGIN
```

```
CATCH, /cancel
HELP, /LAST_MESSAGE, OUTPUT=errMsg
FOR i=0, N_ELEMENTS(errMsg)-1 DO print, errMsg[i]
RETURN
ENDIF

;Check parameter
IF ~filename THEN MESSAGE, 'Please pass a variable'

;Check keyword
do_save = KEYWORD_SET(save)
do_display=KEYWORD_SET(display)

;filename='MYD021KM.A2010109.1305.005.2010111050209.hdf'

nx = 1354
ny = 2030
tn = 36
bands=fltarr(nx,ny,tn)
stddev=fltarr(36)
;---------------------------Read reflectance from bands 1-19,26-------------
 for i = 1,19 do begin
   modis_l1b_read,filename,i,band,/reflectance
   j=i-1
   bands[*,*,j]=band[*,*]
 endfor

for i = 26,26 do begin
   modis_l1b_read,filename,i,band,/reflectance
   j=i-1
   bands[*,*,j]=band
 endfor
;---------------------------Read temperature from bands 20-25, 27-36--------
 for i = 20,25 do begin
   modis_l1b_read,filename,i,band,/temperature
   j=i-1
   bands[*,*,j]=band
 endfor
 for i = 27,36 do begin
   modis_l1b_read,filename,i,band,/temperature
   j=i-1
   bands[*,*,j]=band
 endfor

;Horizontal flip bands' data
for i=0,35 do begin
 bands[*,*,i]=reverse(bands[*,*,i])
endfor

modis_l1b_read,filename,1,band,/reflectance,VIEWZENITH=vza,LATITUDE=LAT



;display each band on screen every 1 second.
if do_display then begin
 for i=0,35 do begin
  window,XSIZE=nx/2,YSIZE=ny/2
```

```
  tvscl,bands[*,*,i]
  wait,1
 endfor
endif

;vis=bands[*,*,3]

if do_save then begin
 save,bands,file='russia_smoke2.sav'; Save all the bands
 save,vza,lat,file='vza_lat.sav'; Save all the bands
endif
end
;main
shay_modis,'MYD021KM.A2011200.0220.005.2011200155505.hdf',/save
end
```

The program to extract a 640*512 image and save as a .sav file

```
;This program is to extract a 640*512 box from any Modis file of 36 bands,
;separately. Alternate is to savethe images as .eps files.
;Also, the extracted arrays are saved as band.sav file.
;This needs 36 original bands being stored in 'russia_smoke2.sav'

restore,'russia_smoke2.sav'
nx = 1354.
ny = 2030.
stddev=fltarr(36)

band=fltarr(640,512,36)
for i=0,35 do begin
 band[*,*,i]=bands[450:1089,200:711,i]
 ;device,retain=2
 ;ps_start,xsize=6.40,ysize=5.12,/inches,/encapsulated,file='hw2_6_vis.eps'
 ;window,/free,xsize=640,ysize=512
 ;tv,hist_equal(band[*,*,i])
 ;wait,2
endfor

restore,'ru2_band.sav'
vzar=vza[450:1089,200:711]
vzar=vzar*.01
f=rebin(lat,1355,2030)
latr=f[450:1089,200:711]
save,latr,vzar,file='extract_vza_lat.sav'


device,retain=2
window,/free,xsize=640,ysize=512
tv,hist_equal(band[*,*,3])
save,band,file='ru2_band.sav'
end
```

The procedure to pick samples

```
; docformat = 'rst'
;+
;
; This program will draw a visible channel image on screen
; and let user pick 10 samples interactively.
;
; :Private:
;
; :Categories:
; Remote Sensing Research/ATS670 Homework
;
; :Examples:
;
shay_pick_sample,'band.sav',U=u,Std=std,Val_std=val_std,Val_u=val_u,Sample=sa
mple
;
;
; :Author:
; Xuechang "Shay" Liu, UAH, xuechang at nsstc.uah.edu
;
; :History:
; Modification History::
; First written: March 4,2013
;
;-
;
Pro
shay_pick_sample,filename,U=u,Std=std,Val_std=val_std,Val_u=val_u,Sample=samp
le
;+
; This procedure will display band 4 of Modis data on screen and let user
pick 5 samples.
; Optional is to output the mean values and standard deviation of each class.
;
;
; :Params:
; filename: in, required, type=string
; This should be the filename of a .sav file, which contains 36 bands of
640*512 Modis data.
;
; :Keywords:
; U : in, optional, type=boolean
; Std : in, optional, type=boolean
; Val_u : out, optional, type=numerical array
;     A float array that stores mean values of 10 samples of 10 bands
; Val_std : out, optional, type=numerical array
;     A float array that stores std.dev values of 10 samples of 10 bands
; Sample : out, optional, type=numerical array
;     The array to store all picks' mean value by band
;
;-
compile_opt idl2

;error catcher:
```

```
CATCH, theError
IF theError NE 0 THEN BEGIN
CATCH, /cancel
HELP, /LAST_MESSAGE, OUTPUT=errMsg
FOR i=0, N_ELEMENTS(errMsg)-1 DO print, errMsg[i]
RETURN
ENDIF

restore,filename
b=band
b1=b[*,*,0];Red
b4=b[*,*,3];Green
b3=b[*,*,2];Blue
num=10
device,retain=2
window,xsize=640,ysize=512,0
;display a 3-band overlay on the screen to for sample-pick
;overlay=fltarr(640,512,3)
;overlay=[[[b[*,*,0]]],[[b[*,*,1]]],[[b[*,*,2]]]]
CH1=HIST_EQUAL(b1)
CH2=HIST_EQUAL(b4)
CH3=HIST_EQUAL(b3)
overlay=COLOR_QUAN(ch1,ch2,ch3,Red,Green,Blue,COLORS=256)
tvlct,red,green,blue
tvimage,overlay
;tv,hist_equal(overlay)
;WRITE_JPEG, 'test.jpeg', TVRD(TRUE=1), TRUE=1, QUALITY=100

; pick region of interest using box_cursor
xs = 640 ; xsize for window
ys = 512 ; ysize for window

xw=10.0 ;the size of a single sample
yw=10.0

sample1=fltarr(yw,xw,num)
sample2=fltarr(yw,xw,num)
sample3=fltarr(yw,xw,num)
sample4=fltarr(yw,xw,num)
sample5=fltarr(yw,xw,num)
sample6=fltarr(yw,xw,num)
sample7=fltarr(yw,xw,num)
sample8=fltarr(yw,xw,num)

sample_mean=fltarr(7,num)

;Pick region using box cursor
i=0
repeat begin
aa = wmenu(['Select: ','Get region','Exit'],title=0,init=1)
if aa eq 2 || i ge num then begin
 wdelete,0
endif

if aa eq 1 && i lt num then begin
wset,0
box_cursor,x0,y0,xw,yw,FIXED_SIZE=50.0
```

```
;center of the box
xcen = (x0+xw/2)-1
ycen = (y0+yw/2)-1
newxw = xw/2 - 1
newyw = yw/2 - 1
print,format='(a4,i1,a15,a16,a36)','Pick',i+1,'Start Position','Size','Center
Postion'
print,x0,y0,xw,yw,xcen,ycen

;get data from box with xcen,ycen as center and xw,yw as widths

img1 = b[xcen-newxw:xcen+newxw+1,ycen-newyw:ycen+newyw+1,0];band 1
img2 = b[xcen-newxw:xcen+newxw+1,ycen-newyw:ycen+newyw+1,1];band 2
img3 = b[xcen-newxw:xcen+newxw+1,ycen-newyw:ycen+newyw+1,2];band 3
img5 = b[xcen-newxw:xcen+newxw+1,ycen-newyw:ycen+newyw+1,4];band 5
img20 = b[xcen-newxw:xcen+newxw+1,ycen-newyw:ycen+newyw+1,19];band 20
img31 = b[xcen-newxw:xcen+newxw+1,ycen-newyw:ycen+newyw+1,30];band 31
img32 = b[xcen-newxw:xcen+newxw+1,ycen-newyw:ycen+newyw+1,31];band 32

sample1[*,*,i]=img1
sample2[*,*,i]=img2
sample3[*,*,i]=img3
sample4[*,*,i]=img5
sample5[*,*,i]=img20
sample6[*,*,i]=img31
sample7[*,*,i]=img32

print,'Mean values of slected bands:'
print,format='(a3," = ",f7.3)','B 1=',mean(img1),'B 2=',mean(img2),'B
3=',mean(img3),$
'B 5=',mean(img5),'B20=',mean(img20),'B31=',mean(img31),'B32=',mean(img32)


sample_mean[*,i]=[[mean(img1)],[mean(img2)],[mean(img3)],[mean(img5)],$
                         [mean(img20)],[mean(img31)],[mean(img32)]]

print,'----------------------------'
endif

i += 1
endrep until(aa eq 2 || i ge num)

sample=sample_mean

if keyword_set(u) then begin
 print,'The mean values for the samples of each band:'
 bn_num=['Band 1','Band 2','Band 3','Band 5',$
            'Band18','Band20','Band31','Band32']
 u=fltarr(7)
 u[0]=mean(sample1)
 u[1]=mean(sample2)
 u[2]=mean(sample3)
 u[3]=mean(sample4)
 u[4]=mean(sample5)
 u[5]=mean(sample6)
 u[6]=mean(sample7)
```

```
 for i =0,6 do print,format='(a7," = ",f7.3)',bn_num[i],u[i]
endif
val_u=u

if keyword_set(std) then begin
 print,'The Standard Deviation for the samples of each band:'
 bn_num=['Band 1','Band 2','Band 3','Band 5',$
             'Band18','Band20','Band31','Band32']
 std=fltarr(7)
 std[0]=stddev(sample1)
 std[1]=stddev(sample2)
 std[2]=stddev(sample3)
 std[3]=stddev(sample4)
 std[4]=stddev(sample5)
 std[5]=stddev(sample6)
 std[6]=stddev(sample7)


 for i =0,6 do print,format='(a6," = ",f7.3)',bn_num[i],std[i]
endif
val_std=std

end

;main
shay_pick_sample,
'ru2_band.sav',/u,/std,Val_std=val_Std,val_u=val_u,Sample=sample
end
```

The MLC procedure

```
; docformat = 'rst'
;+
;
; Maximum likelihood classifyer function
;
; :Private:
;
; :Categories:
; Remote Sensing/ATS670 Homework 4
;
; :Examples:
; for i =0L,num-1 do begin
;  img[i]=shay_mlc(data[i,*],sample=sample)
; endfor
;
;
; :Author:
; Xuechang "Shay" Liu, UAH, xuechang at nsstc.uah.edu
;
; :History:
; First written: March 20, 2013
;
;-
;
Pro
shay_mlc_param,sample,U=u,Determinant=determinant,Inver_epsilon=inver_epsilon
;+
; A helper procedure that computes mean, standard deviation, determinant and
$
; inverse covariance matrix.
;
;
; :Params:
; sample : in, required, type=numerical array
; A 3-dimensional training samples' array of bands*pixels*classes
;
; :Keywords:
; U : out, optional, type=numerical array
; Determinant : out, optional, type=numerical array
; Inver_epsilon : , optional, type=numerical array
;
;
;-

n_class=(size(sample))[3]
num_band=(size(sample))[1]
u=dblarr(n_class,num_band)
std=dblarr(n_class,num_band)
epsilon=dblarr(num_band,num_band,n_class)
inver_epsilon=dblarr(num_band,num_band,n_class)
determinant=dblarr(n_class)

;compute each class' mean by band
;column:class number,row:band number
```

```
for k = 0,n_class-1 do begin
 for i =0,num_band-1 do begin
  u[k,i]=mean(sample[i,*,k])
  std[k,i]=stddev(sample[i,*])
 endfor
endfor

;compute the covariance matrix of each class
for k=0,n_class-1 do begin
  epsilon[*,*,k]=correlate(sample[*,*,k],/covariance)
  determinant[k]=determ(epsilon[*,*,k])
  inver_epsilon[*,*,k]=invert(epsilon[*,*,k])
endfor

end

Function shay_mlc,vector,Sample=sample
;+
; Main function that computes and compares and classify a single MODIS $
; image pixel of 7 bands.
;
; :Returns:
; A integer that describes the class of the this pixel.
;
; :Params:
; vector : in, required, type=numerical array
; a 1-column by multi-row array of the pixel value of all selected bands
;
; :Keywords:
; Sample : in, optional, type=numerical array
; Training samples of all classes
;
; :Uses:
; shay_mlc_param
;
;-

compile_opt idl2

;error catcher:
CATCH, theError
IF theError NE 0 THEN BEGIN
CATCH, /cancel
HELP, /LAST_MESSAGE, OUTPUT=errMsg
FOR i=0, N_ELEMENTS(errMsg)-1 DO print, errMsg[i]
RETURN,-1
ENDIF

shay_mlc_param,sample,U=u,Determinant=determinant,Inver_epsilon=inver_epsilon
n_class=(size(u))[1]
num_band=(size(u))[2]
deviate=dblarr(n_class,num_band) ;x-u
deviate_t=transpose(deviate)
g=dblarr(n_class)

;For a single pixel 7-band vector
for k=0,n_class-1 do begin
```

```
      for j =0,num_band-1 do begin
       deviate[k,j]=vector[0,j]-u[k,j]
      endfor
 deviate_t[*,k]=transpose(deviate[k,*])
 g[k]=-alog(abs(determinant[k]))-
deviate_t[*,k]##inver_epsilon[*,*,k]##deviate[k,*]
endfor
ind=where(g eq max(g))
return,ind
end

;main
T = SYSTIME(1)

restore,'ru2_band.sav'
restore,'cloud_training_samples.sav'
restore,'ocean_training_samples.sav'
restore,'soil_training_samples.sav'
restore,'veg_training_samples.sav'
restore,'smoke_training_samples.sav'

num=640*512L

sample=[[[cloud]],[[ocean]],[[soil]],[[veg]],[[smoke]]]
;SAMPLE          FLOAT     = Array[7, 10, 5]



b=[ [[band[*,*,0]]],$
      [[band[*,*,1]]],$
      [[band[*,*,2]]],$
      [[band[*,*,4]]],$
      [[band[*,*,19]]],$
      [[band[*,*,30]]],$
      [[band[*,*,31]]] ]


data=reform(b,num,(size(b))[3])
img=dblarr(640,512)
;DATA            FLOAT     = Array[327680, 10]


for i =0L,num-1 do begin
 img[i]=shay_mlc(data[i,*],sample=sample)
endfor
save,img,file='mlc.sav'

PRINT, SYSTIME(1) - T, 'Seconds'

;make background
device,decomposed=0,retain=2
loadct,13,bottom=0,ncolors=255
!p.background=255
!p.color=0

;colortable
; white         navy    olive green g-blue
```

```
r=[255,            007,  180,  057,  102];
g=[255,            080,  196,  193,  153];
b=[255,            170,  062,  093,  158];
color_ind=[0,1,2,3,4]

;colorbar
window,/free, xsize=640, ysize=580
tvlct,r,g,b
tv,img,0,68

ClassArr=['Cloud','Ocean','Soil','Vegetation','Smoke']
;The order should be the consistent with array sample.

shay_colorbar,ClassArr,color_ind

image3d = TVRD(TRUE=1)
WRITE_JPEG, 'MLC.jpeg', image3d, TRUE=1, QUALITY=100

end
```

The MMC program

```
;Cluster migrating means method
T = SYSTIME(1)

restore,'ru2_band.sav'

n=640*512L
;b2 =bytscl(band[*,*, 1],max=1.)*1.0
;b32=bytscl(band[*,*,31])*1.0
;b2_1d =reform(b2,1,n)
;b32_1d=reform(b32,1,n)
;b=[b2_1d,b32_1d]

bd=[  [[bytscl(band[*,*,0],max=1.)*1.0]],[[bytscl(band[*,*,1],max=1.)]],$
            [[bytscl(band[*,*,2],max=1.)]],[[bytscl(band[*,*,4])]],$
            [[bytscl(band[*,*,19])]],$
            [[bytscl(band[*,*,30])]],[[bytscl(band[*,*,31])]]  ]

num_band=(size(bd))[3]
b=fltarr(num_band,n)
for i = 0,num_band-1 do begin
 b[i,*]=reform(bd[*,*,i],1,n)
endfor


;9 arbitrary cluster centers
;8 bands
u=[ [210,210,200,210,210,200,205],$
      [190,195,195,190,195,195,185],$
      [180,180,175,180,180,170,180],$
      [165,160,165,165,165,160,165],$
      [140,140,140,145,140,140,145],$
      [125,115,125,125,125,130,125],$
      [100,107,100,105,100,100,105],$
      [087,085,080,080,089,080,080],$
      [065,069,065,065,075,065,071],$
      [052,050,050,055,050,050,050],$
      [027,035,025,035,035,032,035]   ]


;Set initial parameters for repeat cycle
SSE=0.
new_u_valid=u
new_u=make_array((size(u))[1],(size(u))[2],/float,value=300.)
k=0

repeat begin
u=new_u_valid
cols=(size(u))[1]
rows=(size(u))[2]
cluster = intarr(640,512);Classifying results set to 0
last_sse=sse
e=fltarr(rows)
se=dblarr(rows)

for i = 0L,n-1 do begin
```

```
 for j =0,rows-1 do begin
   e[j]=sqrt((b[0,i]-u[0,j])^2+(b[1,i]-u[1,j])^2+(b[2,i]-u[2,j])^2$
      +(b[3,i]-u[3,j])^2+(b[4,i]-u[4,j])^2+(b[5,i]-u[5,j])^2$
      +(b[6,i]-u[6,j])^2);Eucliean distancce
 endfor
  min_ind=where(e eq min(e))
  cluster[i]=min_ind
endfor

for j = 0,rows-1 do begin
 cluster_sse,cluster,b,u,j,SSE=sse0,new_u=new_u0
 se[j]=sse0
; SSE=SSE+sse0
 new_u[*,j]=new_u0
endfor

sse=total(se)

valid_ind=where(new_u[0,*] lt 300)
new_u_valid=new_u[*,valid_ind]

ap=array_equal(u,new_u_valid)

print,'new u'
print,new_u[*,valid_ind]
print,'SSE=',SSE

diff=last_sse - SSE

k++
endrep until ((diff lt 2E4 and diff gt 0) or (SSE lt 5E4) or ap eq 1)

;save,cluster,file='ru2_cluster.sav'

print,'Cycles:',k
PRINT, SYSTIME(1) - T, 'Seconds'

;------------------------display & colorbar------------------

;restore,'ru2_cluster.sav',/verb
;make background
device,decomposed=0,retain=2
loadct,13,bottom=0,ncolors=255
!p.background=255
!p.color=0

;colortable
;        white        tawry   olive navy  g-blue      cyan  light green grey
      red
;red  =[255,            183,  180,  052,  102,  068,  200,  213,  137,  255,
      255]
;green=[255,            150,  196,  143,  153,  249,  255,  253,  137,  000,
      219]
;blue =[255,            042,  042,  255,  158,  255,  255,  197,  137,  000,
      012]

;colortable
```

```
;            white light cyan1 cyan2 lightgrey   green olive grey  navy  cyan3
red  =[255,        200,  068,  149,  197,        000,  180,  137,  008,  185,
      084]
green=[255,        255,  249,  221,  222,        170,  196,  137,  023,  204,
      134]
blue =[255,        255,  255,  255,  233,        015,  062,  137,  160,  233,
      173]

color_ind=[0,1,2,3,4,5,6,7,8,9,10]

;colorbar
window,/free, xsize=640, ysize=580
tvlct,red,green,blue
tv,cluster,0,68


ClassArr=['Class1','Class2','Class3','Class4','Class5','Class6','Class7','Cla
ss8','Class9','Class10','Class11']

shay_colorbar,ClassArr,color_ind,xstart=.1,xend=.9,loc = [0.1, 0.044, 0.9,
0.096]

image3d = TVRD(TRUE=1)
WRITE_JPEG, 'MMC2.jpeg', image3d, TRUE=1, QUALITY=100


;      white     navy    olive green g-blue
r1=[255,           007,  180,  057,   102];
g1=[255,           080,  196,  193,   153];
b1=[255,           170,  062,  093,   158];
color_ind1=[0,1,2,3,4]



end
```

the helper procedure for MMC

```
; docformat = 'rst'
;+
;
; A procedure to calculate the SSE and mean values
;
; :Private:
;
; :Categories:
; Remote Sensing/ATS670
;
; :Examples:
; cluster_sse,cluster,b,u,5,SSE=sse,new_u=new_u
;
;
; :Author:
; Xuechang "Shay" Liu, UAH, xuechang at nsstc.uah.edu
;
; :History:
; Modification History::
; First written: Mar 18, 2013
;
;-
;
Pro cluster_sse,cluster,b,u,cluster_number,SSE=sse,New_u=new_u
;+
; Description...What does this do?
;
; :Returns:
; SSE and new mean values as output keywords
;
; :Params:
; cluster : in, required, type=numerical array
;      a 640*512 array with cluster indices
; b : in, required, type=numerical array
;      a concatenated array of selected bands
; cluster_number : in, required, type=integer
;      the number of the single cluster of interest
;
; :Keywords:
; SSE : out, optional, type=float
;     the SSE of A SINGLE CLUSTER which is chosen by user by cluster_number
; New_u : out, optional, type=numerical array
;     new computed mean values of the original cluster
;
;-

compile_opt idl2

;error catcher:
CATCH, theError
IF theError NE 0 THEN BEGIN
CATCH, /cancel
HELP, /LAST_MESSAGE, OUTPUT=errMsg
FOR i=0, N_ELEMENTS(errMsg)-1 DO print, errMsg[i]
```

```
RETURN
ENDIF

cluster_array = [0]

cluster_ind=where(cluster eq cluster_number,count)
if count gt 0 then cluster_array=reform(b[*,cluster_ind],7,count)

SSE=0.

;SSE of a single cluster
if count gt 0 then begin
 for i = 0,  (size(cluster_array))[2]-1 do begin
      SSE=SSE+(cluster_array[0,i]-u[0,cluster_number])^2+(cluster_array[1,i]-
u[1,cluster_number])^2$
                  +(cluster_array[2,i]-
u[0,cluster_number])^2+(cluster_array[3,i]-u[3,cluster_number])^2$
                  +(cluster_array[4,i]-
u[4,cluster_number])^2+(cluster_array[5,i]-u[5,cluster_number])^2$
                  +(cluster_array[6,i]-u[6,cluster_number])^2
 endfor
endif

new_u=fltarr((size(cluster_array))[1])
if count gt 0 then begin
 for i=0,(size(cluster_array))[1]-1 do begin
  new_u[i]=mean(cluster_array[i,*])
 endfor
endif else begin
 print,'No pixel belongs to cluster center',u[*,cluster_number]

new_u=make_array((size(cluster_array))[1],(size(cluster_array))[1],/float,val
ue=300)
endelse
end
```

Program to merge MMC clusters

```
restore,'ru2_cluster.sav',/verb
;make background
device,decomposed=0,retain=2
loadct,13,bottom=0,ncolors=255
!p.background=255
!p.color=0

;colortable
;        white        tawry   olive navy  g-blue      cyan  light grey  grey
      red
;red  =[255,            183,  180,  052,  102,  068,  200,  137,  213,  255,
      255]
;green=[255,            150,  196,  143,  153,  249,  255,  137,  253,  000,
      219]
;blue =[255,            042,  042,  255,  158,  255,  255,  137,  197,  000,
      012]

;colortable
;           white light cyan1 cyan2 green olive cyan3 navy  g-blue      cyan4
      grey
red  =[255,        200,  068,  149,  057,  180,  145,  007,  102,  187,  177]
green=[255,        255,  249,  221,  193,  196,  255,  080,  153,  238,  177]
blue =[255,        255,  255,  255,  093,  062,  254,  170,  158,  255,  177]

color_ind=[0,1,2,3,4,5,6,7,8,9,10]

;colorbar
window,/free, xsize=640, ysize=580
tvlct,red,green,blue
tv,cluster,0,68


ClassArr=['Class1','Class2','Class3','Class4','Class5','Class6','Class7','Cla
ss8','Class9','Class10','Class11']

shay_colorbar,ClassArr,color_ind,xstart=.1,xend=.9,loc = [0.1, 0.044, 0.9,
0.096]

image3d = TVRD(TRUE=1)
WRITE_JPEG, 'MMC.jpeg', image3d, TRUE=1, QUALITY=100


;     white     navy    olive green g-blue
r1=[255,          007,  180,  057,  102];
g1=[255,          080,  196,  193,  153];
b1=[255,          170,  062,  093,  158];
color_ind1=[0,1,2,3,4]

merge=bytarr(640,512)
num=640*512L
for i=0,num-1 do begin
 if cluster[i] eq 0 or cluster[i] eq 1 or cluster[i] eq 2 or cluster[i] eq 3
or cluster[i] eq 4 $
```

```
 or cluster[i] eq 6 or cluster[i] eq 10 then merge[i]=0
 if cluster[i] eq 4 then merge[i] = 3
 if cluster[i] eq 5 then merge[i] = 2
 if cluster[i] eq 7 then merge[i] = 1
 if cluster[i] eq 8 then merge[i] = 4
endfor

window,/free, xsize=640, ysize=580
tvlct,r1,g1,b1
tv,merge,0,68
ClassArr1=['Cloud','Ocean','Soil','Vegetation','Smoke']

shay_colorbar,ClassArr1,color_ind1

image3d = TVRD(TRUE=1)
WRITE_JPEG, 'MMC_merge.jpeg', image3d, TRUE=1, QUALITY=100

end
```

Hybrid Classification program

```
;hybrid_trainings

 restore,'hcloud1_training_samples.sav'
 restore,'hcloud2_training_samples.sav'
 restore,'hcloud3_training_samples.sav'
 restore,'hcloud4_training_samples.sav'

 restore,'hocean1_training_samples.sav'
 restore,'hocean2_training_samples.sav'
 restore,'hocean3_training_samples.sav'
 restore,'hocean4_training_samples.sav'

 restore,'hveg2_training_samples.sav'
 restore,'hveg3_training_samples.sav'
 restore,'hveg4_training_samples.sav'

 restore,'hsoil1_training_samples.sav'
 restore,'hsoil2_training_samples.sav'

 restore,'hsmoke1_training_samples.sav'
 restore,'hsmoke2_training_samples.sav'
 restore,'hsmoke3_training_samples.sav'
 restore,'hsmoke4_training_samples.sav'

hcloud=(hcloud1+hcloud2+hcloud3+hcloud4)/4.
hocean=(hocean1+hocean2+hocean3+hocean4)/4.
hveg=(hveg2+hveg3+hveg4)/3.
hsoil=(hsoil1+hsoil2)/2.
hsmoke=(hsmoke1+hsmoke2+hsmoke3)/3.


restore,'ru2_band.sav'

num=640*512L

sample=[[[hcloud]],[[hocean]],[[hsoil]],[[hveg]],[[hsmoke]]]

b=[ [[band[*,*,0]]],$
      [[band[*,*,1]]],$
      [[band[*,*,2]]],$
      [[band[*,*,4]]],$
      [[band[*,*,19]]],$
      [[band[*,*,30]]],$
      [[band[*,*,31]]] ]


datah=reform(b,num,(size(b))[3])
imgh=dblarr(640,512)
;DATA            FLOAT     = Array[327680, 10]


for i =0L,num-1 do begin
 imgh[i]=shay_mlc(datah[i,*],sample=sample)
endfor
save,imgh,file='mlch.sav'
```

```
;make background
device,decomposed=0,retain=2
loadct,13,bottom=0,ncolors=255
!p.background=255
!p.color=0

;colortable
; white         navy    olive green g-blue
r=[255,           007,  180,  057,  102];
g=[255,           080,  196,  193,  153];
b=[255,           170,  062,  093,  158];
color_ind=[0,1,2,3,4]

;colorbar
window,/free, xsize=640, ysize=580
tvlct,r,g,b
tv,imgh,0,68

ClassArr=['Cloud','Ocean','Soil','Vegetation','Smoke']
;The order should be the consistent with array sample.

shay_colorbar,ClassArr,color_ind

image3d = TVRD(TRUE=1)
WRITE_JPEG, 'MLCH.jpeg', image3d, TRUE=1, QUALITY=100

end
```