Hyperspectral satellite image classification using small training data from its samples

V A Fedoseev

1Samara National Research University, Moskovskoe Shosse 34, Samara, Russia, 443086
2Image Processing Systems Institute - Branch of the Federal Scientific Research Centre “Crystallography and Photonics” of Russian Academy of Sciences, Molodogvardeyskaya str. 151, Samara, Russia, 443001

e-mail: vicanfed@gmail.com

Abstract. The paper is aimed to develop an appropriate multi-stage algorithm to perform thematic classification of hyperspectral satellite images based on small training sets of data selected manually from localized image parts. This algorithm is needed in the case of the lack of sensing condition data and quite frequent in practice. The chosen technology includes the stages of pixel-wise and spatial pre-processing, image classification based on spatial and spectral factors, as well as spatial post-processing of the classification results. Experimental studies have shown a significant decrease in the classification quality under the conditions considered in the paper. As a result of the experiments, the highest results were achieved by the algorithm based on combining pixel-wise classification results and segmentation results obtained using k-means++ and connected components labeling, supplemented by nonlinear pre- and post-processing methods. The findings are supported by the results of comparative studies of different methods at each stage of the classification.

1. Introduction

1.1. Problem Statement

When solving problems of the thematic analysis of the territory, it is not always possible to use space images that are best suited for a particular task. This may be due to financial factors (we cannot always have an opportunity to buy the best shots), as well as organizational ones (i.e., no access to the snapshots for the analyzed period), weather (high cloudiness at the time of the ordered shooting) and some others. In such a situation, there is a need to use any available images related to the period of interest, including those for which there is no complete information about the shooting conditions and camera settings.

Let us consider the following example. Let it be necessary to perform thematic classification of a terrain which contains objects whose spectral reflectance is known from an available spectral library. However, we have the only hyperspectral image, which has not undergone an atmospheric correction, or the wavelengths of its channels are unknown or known inaccurately. As a result, the use of spectra from the library as training data will lead to false results.
In such cases, a researcher who knows at least some objects on the ground, could select class examples in the image and use them as a training sample to solve the classification problem on the whole image. Thus, the task is to choose the most appropriate classification algorithm in the following constraints:

- Hyperspectral data are considered "as is", i.e., without semantic information about channels, shooting parameters, etc.
- The algorithm should be effective both when working with raw data, and when working with data that has undergone an atmospheric correction.
- The training sample is formed from the image pixels localized in a small area for each class.
- A limited amount of training data (down to several dozens of examples for each class).

1.2. Review of current solutions

The very basic approach to hyperspectral remote sensing data classification is the scalar pixel-wise classification based on supervised learning. In this approach, each pixel is classified separately based on its spectral characteristics. Thus, spatial relationships between pixels are lost, but the problem is reduced to the simple problem of one-dimensional data classification. Apparently, the scalar classification results can be improved by using spatial information, as well as information on data distribution in the feature space. Most known algorithms that take into account these factors comprise image clustering or segmentation, followed by aggregation of the results obtained with the results of scalar classification [1, 2]. For instance, Borzov and Potaturkin in [1] compare four different methods of spectral-spatial classification based on segmentation and sliding window processing. Zimichev et al. in [2] apply k-means++ clustering with the subsequent search for connected components, and obtain the final result by majority voting.

In this paper, we propose an algorithm for hyperspectral image classification, which is compiled from a series of studies and consists of several stages. The choice of the techniques applied at each stage is made by comparative experimental studies of various methods under the conditions determined by the considered problem.

2. Experimental data preparation

2.1. Training data

This paper is rather empirical in the sense that we propose a multi-stage algorithm structure and then perform experimental studies to select the most appropriate methods to be used at each stage. Therefore, first we need to specify the key parameters used in the experiments.

For the comparative analysis of different classification strategies, we used the following variants of per-class training set size: 15, 50, 100, and also 1, 5, 10 and 25 percent of all instances of each class. For comparative studies, we also used two ways of forming a training set: random and localized. In the first case, the training pixels are randomly selected from the whole image, while in the second case for each class the training pixels are taken from the single randomly chosen spatial cluster.

In this study, we used two hyperspectral images:

- ‘Indian Pines’ (made by the AVIRIS sensor): 145×145 pixels, 200 spectral channels, ground truth data contain 10250 examples of 16 classes,
- ‘Pavia University’ (made by the ROSIS sensor): 610×340 pixels, 102 spectral channels; ground truth data contain 42787 examples of 9 classes.

We conducted all the experiments using these two images but most results presented in Sections 4–6 are obtained for ‘Indian Pines’. This is done in order to save some space because most implications for different images coincide.

As a quality measure for classification algorithms, we used accuracy, equal to the fraction of correctly classified pixels from all ground truth data.
2.2. Atmospheric Distortion Model

To analyze the efficiency of classification algorithms for data without atmospheric correction, we need data before and after the correction with the marked ground truth. Unfortunately, in the open access, there are practically no such data sets, especially hyperspectral ones. Therefore, in this study, we performed modeling of atmospheric distortions for the selected ‘Indian Pines’ and ‘Pavia University’ images. The modeling was carried out for each pixel according to the standard radiation transfer equation of the MODTRAN model [3–7]:

\[ L = \frac{A \rho}{1 - \rho S} + \frac{B \rho e}{1 - \rho S} + C, \]

where \( L \) is detected radiance, \( \rho \) is reflectance at the target pixel, \( \rho e \) is the average reflectance in the neighborhood of the target pixel, \( S \) is the spherical albedo of the atmosphere, \( C \) is the radiance of the atmospheric. Finally, \( A \) and \( B \) are the coefficients that depend on atmospheric and geometry conditions. Thus, we can model atmospheric distortions by using image pixel values (the corrected ones) as \( \rho \) and the values smoothed by a spatial filter as \( \rho e \), and by setting other parameters to their typical values.

3. Proposed multi-stage classification algorithm

After the literature review, we selected the classification algorithm which consists of the following stages:
1. Pixel-wise preprocessing of the input data.
2. Spatial preprocessing of the input data.
3. Data classification technique based on the cluster structure of the data and its spatial properties.
Includes the following methods:
3.1. Method of single feature vector classification (applied either for all image pixels or for means or medians of pixel clusters).
3.2. Method of accounting for the data structure in feature space.
3.3. Method of accounting for spatial relationships.
4. Spatial post-processing of the obtained classification results.

In the next sections, we specify and investigate the algorithms used at each of these stages. We should only note that at the beginning of our study, we selected SVM-RBF model as the primary element-wise classification method (Stage 3.1). Such a choice is derived from the fact that SVM-RBF was successfully examined in some papers [1–2, 8] and is supported by our preliminary studies.

4. Pixel-wise preprocessing of the input data

For pixel-wise preprocessing of the input data (Stage 1), we considered the following methods: Principal Component Analysis (PCA), Minimum Noise Fraction Transform (MNF) [9], Noise-Adjusted Principal Components Transform (NAPC) [10] and consistent application of PCA and NAPC (NAPC+PCA). In all cases, 20 features were left to be used for further classification, which is a chosen as a compromise between the quality of the classification and computational efficiency.

Table 1 presents the results of the comparative study. It clearly shows the advantage of NAPC+PCA, that is why this method was chosen for the subsequent experiments. We can also note that the localized way of training sample formation significantly reduces accuracy. But according to the goals of this study (see Section 1.1) we should analyze the localized way primarily and try to improve the results shown in Table 1.

5. Accounting for spatial relationships and data structure in feature space in image classification

5.1. Methods description

To take into account data relationships in spatial and feature domains during the classification stage, we examined two clustering methods (Mean Shift and k-means++) and their simple segmenting enhancement.
Table 1. Performance of different data preprocessing methods (in terms of resulting scalar classification accuracy, 'Indian Pines').

| Method          | Training pixels per class | Localized training set | | | | | Randomly generated training set | | | |
|-----------------|---------------------------|------------------------|---|---|---|---|---|---|---|---|
|                 |                           | 15 | 50 | 100 | 1% | 5% | 10% | 25% | 15 | 50 | 100 | 1% | 5% | 10% | 25% |
| PCA             |                           | 0.445 | 0.528 | 0.605 | 0.469 | 0.521 | 0.596 | 0.708 | 0.596 | 0.688 | 0.742 | 0.547 | 0.733 | 0.763 | 0.846 |
| MNF             |                           | 0.456 | 0.497 | 0.598 | 0.466 | 0.512 | 0.564 | 0.708 | 0.603 | 0.694 | 0.775 | 0.601 | 0.722 | 0.768 | 0.848 |
| NAPC            |                           | 0.481 | 0.536 | 0.610 | 0.491 | 0.538 | 0.585 | 0.711 | 0.612 | 0.697 | 0.750 | 0.590 | 0.727 | 0.765 | 0.841 |
| NAPC+PCA        |                           | 0.494 | 0.571 | 0.618 | 0.498 | 0.570 | 0.612 | 0.737 | 0.603 | 0.730 | 0.788 | 0.598 | 0.759 | 0.812 | 0.867 |

Mean shift is a clustering algorithm proposed in [11]. In this algorithm, local maxima of the kernel density estimation are sought, based on the assumption that they correspond to the centers of the clusters. Essential advantages of this algorithm include the absence of assumptions about cluster shapes and the distribution law, as well as the automatic way to determine the number of clusters. The k-means++ algorithm [12] is an extension of the standard k-means algorithm with an improved initial approximation step. Unlike Mean Shift, this algorithm takes the number of clusters as an input. We also considered both algorithms in the form of segmentation. To do this, we performed the connected components labeling on the clustering results. As a result, the data were further partitioned by spatial dimension.

In [1, 2] two ways of aggregating classification results with clustering (or segmentation) results are considered. Both are based on the principle that all the pixels belonging to one region (segment or cluster) should be attributed to one class. The first method is the majority voting: for each region \( P \), we calculate the statistics of scalar classification results and select the most frequent class to be set for the entire region \( P \). In the second method, we calculate a single feature vector \( u \) from all the feature vectors \( y_i \in P \) of each region \( P \) as their median, and then we classify the only one feature vector and apply the results for the whole region.

5.2. Experimental results

In this subsection, we present the results of the experimental study on the methods specified above. Table 2 shows the results of this study in the same format as Table 1 (but only for the localized training set). To save space in the table, we showed only the best results obtained by one of two aggregation methods (majority voting and median classification). In Table 2 and the next ones, we highlight in italics the scalar classification results which are given for comparison. We also mark in bold the highest results for the column and underline the results which are very close to the highest.

Let us list the main conclusions from the results obtained. The considered methods allow improving classification quality, although the difference if not sufficient for the localized training set. However, clustering without segmentation does not lead to significant improvements. Moreover, often scalar classification wins it as Table 2 shows. We can also note that k-means++ achieves higher scores more often than Mean Shift does. Combined with the higher speed of k-means++, we can give preference to this method. Concerning two aggregation methods, we cannot come to an unambiguous conclusion because different methods work better in different cases.

An important issue that needs to be clarified is the influence of the atmospheric correction on classification accuracy. Table 3 shows the comparative results of scalar classification and the best configuration from Table 2 for distorted and distortion-compensated data. Surprisingly, the results are slightly higher for distorted data. We can explain it with the spatial averaging operation used in the
MODTRAN model [3–7], which implicitly provides an additional accounting for spatial information. The obtained results lead to the conclusion that we can use the distortion-compensated data to find the best algorithm. Therefore, we shall not give the results for distorted data in next sections.

Table 2. Performance of different clustering and segmentation methods (in terms of resulting classification accuracy, ‘Indian Pines’, localized training set).

| Method | Type         | Aggregation | Regions | Training pixels per class |
|--------|--------------|-------------|---------|--------------------------|
|        |              |             | 15      | 50 | 100 | 1%  | 5%  | 10% | 25% |
| No (scalar classification) | – | – | – | 0.494 | 0.571 | 0.618 | 0.498 | 0.570 | 0.612 | 0.737 |
| k-means++ | Clustering | Voting | 100 | 0.449 | 0.487 | 0.515 | 0.474 | 0.498 | 0.495 | 0.582 |
| k-means++ | Clustering | Voting | 400 | 0.486 | 0.552 | 0.571 | 0.502 | 0.548 | 0.571 | 0.645 |
| Mean Shift | Clustering | Median | 3550 | 0.496 | 0.565 | 0.584 | 0.488 | 0.563 | 0.573 | 0.667 |
| Mean Shift | Clustering | Voting | 450 | 0.437 | 0.452 | 0.446 | 0.454 | 0.456 | 0.455 | 0.491 |
| k-means++ | Segmentation | Voting | 3000 | 0.511 | 0.594 | 0.601 | 0.493 | 0.591 | 0.597 | 0.759 |
| k-means++ | Segmentation | Median | 6000 | 0.502 | 0.593 | 0.635 | 0.501 | 0.580 | 0.618 | 0.747 |
| Mean Shift | Segmentation | Median | 8400 | 0.491 | 0.577 | 0.637 | 0.497 | 0.570 | 0.621 | 0.750 |
| Mean Shift | Segmentation | Voting | 4500 | 0.434 | 0.521 | 0.550 | 0.453 | 0.509 | 0.515 | 0.639 |

Let us consider the same experiment as shown in Table 2 obtained for the random training set. The results shown in Table 4 prove that random sampling significantly improves accuracy. Therefore, if it is possible for a user to mark out more training areas in spatially separated fragments of the analyzed image, he should not underestimate the importance of this work. Notice that the cluster methods, in this case, work unequivocally worse than the scalar classification, so their application, in this case, does not make sense. However, the use of segmentation methods still allows to achieve better results, and the same algorithms as for the localized training set proved to be the best.

6. Pre- and post-processing methods

Paper [1] notes that the spatial preprocessing of input data (in particular, median filtering) can also improve scalar classification accuracy and can be used in place of segmentation. We can also add that nothing prevents from using this procedure not instead of, but in conjunction with segmentation. Partly the validity of the theses of the paper [1] was confirmed in the experiment with distorted data (see Table 3) since these data are also subjected to low-frequency filtering. Thus, it is advisable to investigate the influence of the low-frequency filtering of the initial data on the classification quality.

In addition to preprocessing, spatial post-processing of classification results also plays an important role. As a rule, post-processing is performed using a non-linear sliding window procedure. For example, in [2], for each window position, the most common class of pixels within the window is determined, as well as the number of such pixels $N$. If the class at the window center is not assigned to the majority class and $N$ exceeds a certain threshold $T$, then the label of the majority class is assigned to the center point.

In addition to this procedure, the reclassification of small segments is also often used in practice. The minimum allowable size of the area is set parametrically and depends on the spatial resolution of a particular image and its content. Therefore, despite its unconditional importance, we do not investigate this procedure in the paper.
Table 3. Influence of atmospheric correction on the classification accuracy (’Indian Pines’, localized training set).

| Method                  | Type       | Aggregation | Regions | 15   | 50   | 100  | 1%  | 5%  | 10% | 25% |
|-------------------------|------------|-------------|---------|------|------|------|-----|-----|-----|-----|
| Distortion-compensated data |
| No (scalar classification) | –          | –           | –       | 0.494| 0.571| 0.618| 0.498| 0.570| 0.612| 0.737 |
| k-means++               | Segmentation| Voting      | 3000    | 0.511| 0.594| 0.601| 0.493| 0.591| 0.597| 0.759 |
| Distorted data |
| No (scalar classification) | –          | –           | –       | 0.500| 0.591| 0.642| 0.514| 0.570| 0.631| 0.765 |
| k-means++               | Segmentation| Voting      | 4500    | 0.509| 0.600| 0.651| 0.522| 0.577| 0.629| 0.773 |

Table 4. Performance of different clustering and segmentation methods (in terms of the resulting classification accuracy, ’Indian Pines’, random training set).

| Method                  | Type       | Aggregation | Regions | 15   | 50   | 100  | 1%  | 5%  | 10% | 25% |
|-------------------------|------------|-------------|---------|------|------|------|-----|-----|-----|-----|
| No (scalar classification) | –          | –           | –       | 0.603| 0.730| 0.788| 0.598| 0.759| 0.812| 0.867 |
| k-means++               | Clustering | Voting      | 100     | 0.460| 0.540| 0.560| 0.538| 0.599| 0.595| 0.590 |
| k-means++               | Clustering | Voting      | 400     | 0.553| 0.667| 0.683| 0.582| 0.675| 0.692| 0.705 |
| Mean Shift              | Clustering | Median      | 3550    | 0.564| 0.670| 0.679| 0.529| 0.688| 0.706| 0.728 |
| Mean Shift              | Clustering | Voting      | 450     | 0.394| 0.478| 0.494| 0.409| 0.523| 0.545| 0.546 |
| k-means++               | Segmentation| Voting      | 3000    | 0.634| 0.757| 0.821| 0.624| 0.791| 0.830| 0.880 |
| k-means++               | Segmentation| Median      | 6000    | 0.645| 0.739| 0.835| 0.622| 0.780| 0.827| 0.886 |
| Mean Shift              | Segmentation| Median      | 8400    | 0.629| 0.752| 0.815| 0.600| 0.767| 0.822| 0.880 |
| Mean Shift              | Segmentation| Voting      | 4500    | 0.598| 0.627| 0.659| 0.521| 0.704| 0.763| 0.808 |

Table 5. Impact of pre- and post-processing procedures on the classification accuracy (’Indian Pines’, localized training set).

| Classification method | Window size at pre-processing | Post-processing | Training pixels per class |
|-----------------------|-------------------------------|-----------------|--------------------------|
|                       |                               |                 | 15   | 50   | 100  | 1%  | 5%  | 10% | 25% |
| Scalar classification  |                               |                 | 1×1  | No   | 0.494| 0.571| 0.618| 0.498| 0.570| 0.612| 0.737 |
|                       |                               |                 | 3×3  | No   | 0.501| 0.570| 0.649| 0.516| 0.582| 0.637| 0.756 |
|                       |                               |                 | 5×5  | No   | 0.469| 0.574| 0.656| 0.463| 0.578| 0.655| 0.774 |
|                       |                               |                 | 1×1  | Yes  | 0.517| 0.600| 0.642| 0.520| 0.607| 0.640| 0.776 |
|                       |                               |                 | 3×3  | Yes  | 0.510| 0.582| 0.664| 0.526| 0.591| 0.647| 0.781 |
|                       |                               |                 | 5×5  | Yes  | 0.474| 0.586| 0.665| 0.468| 0.583| **0.662**| 0.784 |
### Table 6. Impact of pre- and post-processing techniques on the classification accuracy ('Indian Pines', random training set).

| Classification method | Window size at preprocessing | Post-processing | Training pixels per class |
|------------------------|-----------------------------|-----------------|--------------------------|
|                        | 1×1                         | No              | 15  | 50  | 100 | 1%  | 5%  | 10% | 25% |
| Scalar classification  | 1×1                         | Yes             | 0.706 | 0.825 | 0.886 | 0.683 | 0.829 | 0.879 | 0.943 |
|                        | 7×7                         | No              | 0.706 | 0.825 | 0.886 | 0.683 | 0.829 | 0.879 | 0.943 |
|                        | 9×9                         | No              | 0.706 | 0.825 | 0.886 | 0.683 | 0.829 | 0.879 | 0.943 |
|                        | 1×1                         | Yes             | 0.830 | 0.866 | 0.915 | 0.740 | 0.885 | 0.918 | 0.930 |
|                        | 7×7                         | No              | 0.830 | 0.866 | 0.915 | 0.740 | 0.885 | 0.918 | 0.930 |
|                        | 9×9                         | No              | 0.830 | 0.866 | 0.915 | 0.740 | 0.885 | 0.918 | 0.930 |
|                        | 1×1                         | Yes             | 0.838 | 0.880 | 0.916 | 0.772 | 0.895 | 0.921 | 0.932 |
|                        | 7×7                         | Yes             | 0.838 | 0.880 | 0.916 | 0.772 | 0.895 | 0.921 | 0.932 |
|                        | 9×9                         | Yes             | 0.838 | 0.880 | 0.916 | 0.772 | 0.895 | 0.921 | 0.932 |
| Highest gain, relative to scalar classification results, % | 35.38 | 26.14 | 20.91 | 42.52 | 25.49 | 20.35 | 13.83 |

Tables 5-6 show the results of a study of pre- and post-processing methods. They were used in conjunction with the following classification algorithms (used at Stage 3):
- Scalar classification.
- Segmentation based on k-means++ with majority voting.
- Segmentation based on k-means++ with median classification.

For median pre-processing we used window sizes from 3×3 to 9×9 (Tables 5-6 contain some of the results obtained). At the post-processing stage, we used fixed parameters selected after preliminary tests: a 3×3 window and \( N = 5 \) (out of 8 possible).

The obtained results lead us to the following consequences. The post-processing procedure improves the classification quality significantly since all the best results were obtained with this procedure. The local pre-processing procedure does not always improve classification accuracy for the localized training set. However, in the random case, its positive effect on the classification accuracy is clear. The results on the highest increase show that the procedures considered allow to reach the gain of up to 11% (for the localized training set), and up to 42.5% (for the random training set). Of the two aggregation methods, majority voting is more often leads to the better results.

Table 7 highlights the main results obtained for the second image, ‘Pavia University’. The table shows that the highest gain is 7.4% (for the localized training set), and 33.9% (for the random training set). Other conclusions coincide with those made for the first test image.

### Table 7. Impact of pre- and post-processing techniques on the classification accuracy (‘Pavia University’, total results).

| Classification method | Training set  | Training pixels per class |
|-----------------------|---------------|----------------------------|
|                       |               | 15 | 50 | 100 | 1% | 5% | 10% | 25% |
| Scalar classification  | Localized     | 0.519 | 0.619 | 0.628 | 0.695 | 0.718 | 0.720 | 0.718 |
| Our best result        | Localized     | 0.526 | 0.659 | 0.675 | 0.716 | 0.729 | 0.743 | 0.744 |
| Highest gain, %        | Localized     | 1.337 | 6.458 | 7.363 | 3.043 | 1.429 | 3.189 | 3.629 |
| Scalar classification  | Random        | 0.715 | 0.832 | 0.857 | 0.864 | 0.917 | 0.932 | 0.950 |
| Our best result        | Random        | 0.957 | 0.982 | 0.987 | 0.981 | 0.993 | 0.995 | 0.996 |
| Highest gain, %        | Random        | 33.90 | 18.06 | 15.10 | 13.55 | 8.28  | 6.78  | 4.86  |

### 7. Conclusion

We can draw the following conclusions from all the experiments conducted.

1. When selecting reference image areas used for training, it is important to collect samples from distant parts of the image for each class, because such practice improves sufficiently the classification quality. If the data have not passed the preliminary procedure of atmospheric correction, this is not a significant obstacle to conduct classification.

2. For preliminary pixel-wise data processing, we suggest to use the combination of NAPC and PCA transforms.

3. The SVM-RBF classification model has been found acceptable for the problem considered. During the learning, it is desirable to test a set of regularization parameters and window widths.

4. In addition to pixel-wise pre-processing, we found it worthwhile to apply spatial pre-processing. Specifically, we recommend using median filtering with windows from 3×3 to 5×5. However, the greatest positive effect of such a procedure is achieved for random training samples. If the training sample is localized, it is better to use a small window size – 3×3, or completely skip this step.

5. To take into account the spectral and spatial relationships between the image points, one should apply any segmentation algorithm that provides a large number of segments. For example, k-means++ clustering with subsequent connected components labeling can be a good choice. To aggregate segmentation and classification results, we give the preference to the majority voting method.

6. We also found helpful to apply non-linear post-processing of the obtained classification results. The acceptable parameter values are the 3×3 window size and \( N = 5 \).

7. As a final step, we recommend reclassifying small areas according to the neighboring pixels. The size of such areas should be set from a priori knowledge and depends on the spatial resolution.
8. Our studies have shown that the synthesized algorithm together with the selected methods used on its stages can provide an increase in classification accuracy by up to 11% (for the localized training set), and by up to 42.5% (for the random training set) on the two test images.

8. References

[1] Zimichev E A, Kazanskiy N L and Serafimovich P G 2014 Spectral-spatial classification with k-means++ particional clustering *Computer Optics* **38**(2) 281-286

[2] Borzov S M and Potaturkin O I 2017 Efficiency of the spectral-spatial classification of hyperspectral imaging data *Optoelectron. Instrument. Proc.* **53** 26-34

[3] Xu Y, Wang R, Liu S, Yang S and Yan B 2008 Atmospheric correction of hyperspectral data using MODTRAN model in *Remote Sensing of the Environment: 16th National Symposium on Remote Sensing of China* **7123** (International Society for Optics and Photonics) p. 712306

[4] Belov A M and Myasnikov V V 2014 Atmospheric correction of hyperspectral images using approximate solution of MODTRAN transmittance equation *Computer Optics* **38**(3) 489-493

[5] Belov A M and Myasnikov V V 2015 Atmospheric correction of hyperspectral images based on approximate solution of transmittance equation *Seventh International Conference on Machine Vision (ICMV 2014)* **9445** (International Society for Optics and Photonics) p. 94450S

[6] Denisova A Y and Myasnikov V V 2016 Atmospheric correction of hyperspectral images using small volume of the verified data *Computer Optics* **40**(4) 526-534 DOI: 10.18287/2412-6179-2016-40-1-526-534

[7] Denisova A and Myasnikov V 2017 Atmospheric correction of hyperspectral images using qualitative information about registered scene in *Ninth International Conference on Machine Vision (ICMV 2016)* **10341** (International Society for Optics and Photonics) p. 1034125

[8] Kuznetsov A V and Myasnikov V V 2014 A comparison of algorithms for supervised classification using hyperspectral data *Computer Optics* **38**(3) 494-502

[9] Green A A, Berman M, Switzer P and Craig M D 1988 A transformation for ordering multispectral data in terms of image quality with implications for noise removal *IEEE Transactions on Geoscience and Remote Sensing* **26** 65-74

[10] Lee J B, Woodyatt A S and Berman M 1990 Enhancement of high spectral resolution remote-sensing data by a noise-adjusted principal components transform *IEEE Transactions on Geoscience and Remote Sensing* **28** 295-304

[11] Comaniciu D and Meer P 2002 Mean shift: a robust approach toward feature space analysis *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** 603-619

[12] Arthur D and Vassilvitskii S 2007 k-means++: The advantages of careful seeding in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (Society for Industrial and Applied Mathematics) 1027-1035

Acknowledgments

This work was supported by the Russian Foundation for Basic Research (grants 16-29-09494, 16-41-630676), by Project 5-100, and by the Federal Agency for Scientific Organizations (Agreement 007-GZ/43363/26).