

# Intelligent system of forest area recognition for tasks of geographically distributed economic systems

A.A. Kuzmenko, D.A. Kondrashov, A.S. Sazonova, L.B. Filippova, R.A. Filippov  
alex-rf-32@yandex.ru | kuzmenko-alexandr@yandex.ru | asazonova@list.ru | libv88@mail.ru | redfil@mail.ru  
Bryansk State Technical University, Bryansk, Russia

For a long period, our country has been in the process of radical transformations of the state economic system, associated with the final transition to a market system of management, the development of local self-government and the independence of economic entities. In the new conditions of the emerging market, the issues of ensuring the sustainable development of territorial economic systems and sectors of the economy, which are the source and guarantor of social stability, employment, a high level and quality of life of the population of the regions, come to the fore. The paper deals with an intelligent system for recognizing the dynamics of changes in forest areas based on automatic pattern recognition methods. The existing methods of processing graphical information, classification and clustering methods that are of value within the framework of the problems being solved are considered, and several original algorithms are proposed. LTP and FFT algorithms were selected as feature extractors of which the simplest and most productive option is LTP. Histogram equalization algorithms, median and Gaussian filters to eliminate noise and remove small image details are chosen to pre-process the image. Euclidean and Mahalanobis distances were used as separability measures. Naive Bayes classifier is proposed to use for classification.

**Key words:** intelligent systems, pattern recognition, geographically distributed, economic systems

## 1. Introduction

Modern software market is able to offer a system for automation or solving almost any task. Problems of forest protection and forest management were also not ignored: there is a wide range of software that automates accounting activities, is integrated with GIS systems and provides forest planning capabilities, access to tax and cadastral maps, as well as acquisition and processing capabilities of remote sensing data.

There are not many systems focused on automatic processing of satellite images, and their functionality is unique compared to their analogues. For example, "ScanEx Image Processor" system is quite versatile and allows processing both the supplied database of images and images from its own sources, but the system is closed, provides a trial version only by agreement with the manufacturer, and does not allow modification of the algorithms used. "Forestry and land use" is focused only on processing the vendor's own database of images. "KEDR" system is available only to state structures of Amur and Primorye territories and does not even have open documents. Such introductory conditions complicate the search for the turnkey system.

## 2. Materials and methods

When image recognition is based on bitmap graphics, arrays of image pixels play the role of data arrays. Raw data sets have extra information, which in addition to increasing computational complexity can lead to the so-called retraining of classifiers. Also, feature extraction and classification algorithms are sensitive to the transformation of the data used to different extend.

So, to create a stable algorithm for recognizing the forest texture, it is necessary to set:

- image zoom in m/pixel (m/px);
- optimal image segment size suitable for classification;
- algorithm for reducing the amount of information in the image;
- algorithm for equalizing the color of photos.

The scale and size of the window can only be set experimentally, which will be done in the corresponding part of the work. Reducing the amount of information means applying filters to the image that suppress noise and unnecessary details. Color equalization involves equalizing the intensities in the channels used – the so-called equalization of the image histogram.

## 3. Image Filtering

Most of the image transformation methods used in this paper are based on convolution. Correlation and convolution are two closely related concepts. Correlation is the process of moving the filter mask over an image and calculating the sum of the products of the mask element values and the pixel values that the corresponding mask elements fall on. Convolution mechanisms are the same, except the filter mask is pre-rotated 180° [3,9]. Analytically, convolution is expressed as follows.

The filter, or convolution kernel, is a square or rectangular matrix with an odd number of rows and columns. The odd number is due to the fact that the convolution result is assigned to the pixel, the response center of the kernel (Fig. 1).

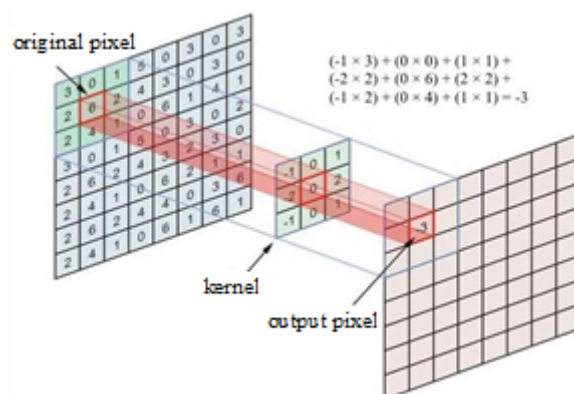


Fig. 1. Application of Sobel filter (edge detection)

Convolution cannot be used for extreme pixels. This problem is solved by creating an intermediate image with

the completed extreme rows and columns. Pixels can be either zero-intensive or copy the extreme ones. The second method is used in this work.

Filtering methods are distinguished in the spatial and frequency domains. Processing methods in the spatial domain contain approaches based on direct manipulation of image pixels. Spatial processing is characterized by the equation [4]:

$$g(x, y) = T[f(x, y)], \quad (1)$$

where:  $f(x, y)$  is an input image;  $g(x, y)$  is an output image;  $T$  is an operator on  $f$  in a certain point  $(x, y)$ .

The main approach to defining a neighborhood is to select a rectangular area around the original pixel  $(x, y)$ . To find  $g$  value at a certain point  $(x, y)$ ,  $f$ - function value is used inside a certain neighborhood of the point. This approach is based on the use of masks – two-dimensional arrays of function values. The most well-known methods in this category are linear and median filtering.

An averaging filter is used as a linear filter; its output value is the average value in its mask neighborhood. The same filter is used for removing image graininess caused by impulse noise.

Fig. 2 shows an example of processing a noisy image with median and linear filters [4,10].

Based on experiments [4], it is concluded that a median filter is more suitable for impulse noise, which preserves good element boundaries and has a high speed.

Filtering in the frequency domain is based on Fourier transformation. Transformation means that any function that periodically reproduces its values can be represented as the sum of sine/cosine of different frequencies multiplied by some coefficients. This sum is called Fourier series.

One of the most important properties of Fourier transformation is that its result can be restored to its original form without loss of information.

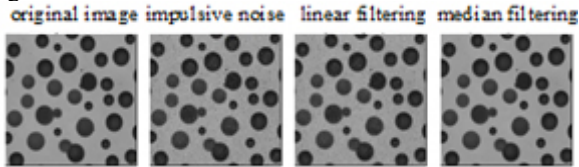


Fig. 2. Sample of median and linear filtering

#### 4. Feature extractors

As it is shown in [9], the use of images in their original form is ineffective within the classification task. The largest amount of data about the surface type in a photo is provided by patterns in its structure. To obtain these patterns special algorithms are used that is feature extractors [5]. The best-known feature extractors include artificial neural networks, algorithms based on Fourier transformations and so-called descriptors of key points.

Fourier transformation described above also has a discrete form that is suitable for digital image processing:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N}, \quad (2)$$

$$x = \frac{1}{N} \sum_{k=0}^{N-1} X \cdot e^{i2\pi kn/N}, \quad (3)$$

where:

- $X_k$  is the transformation result;

- $x_n$  is  $n$ -value of value vector;
- $i$  is unit imaginary number;
- $k$  is a complex sinusoid frequency.

The original data for the algorithm is a vector of function values with a specified step. The result of the algorithm is a vector of complex numbers, for which the index is the frequency value, and the real and imaginary parts are the coordinates of the radius vector point.

The frequency and amplitude components of the signal are vector arguments and the complex number module respectively. The module is defined as the length of the radius vector:

$$r = |a + b \cdot i| = \sqrt{a^2 + b^2}, \quad (4)$$

where  $a$  is the real part,  $b$  is the argument of the imaginary part.

The argument is defined as the angle between the radius vector and the plane:

$$\operatorname{tg}(\varphi) = \frac{b}{a}, \quad \cos(\varphi) = \frac{a}{\sqrt{a^2 + b^2}}, \quad \sin(\varphi) = \frac{b}{\sqrt{a^2 + b^2}}, \quad (5)$$

The image cannot be represented as a one-dimensional vector of numbers without losing important information. To obtain the spectrum of a two-dimensional vector of numbers, FFT is first applied to the columns, and then to the rows of the matrix formed (Fig. 3).

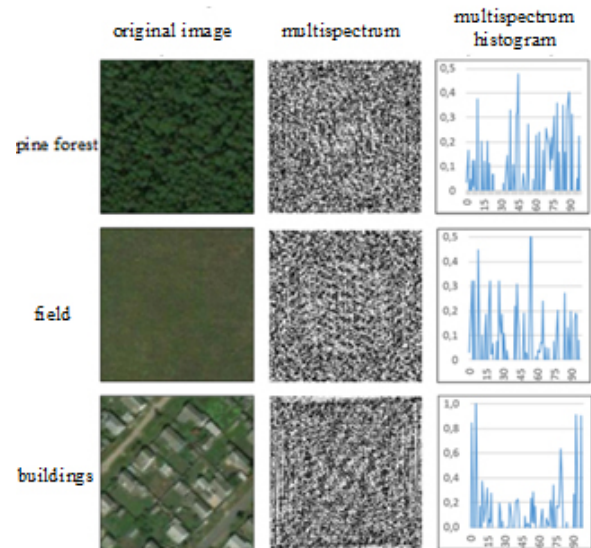


Fig. 3. Feature extracting by means of FFT

FFT use gives well-separable vectors of class features, but the described algorithm has a number of disadvantages. First, FFT calculating is quite resource-intensive, especially against LTP background. The second feature is the peculiarity of the obtained vector for an image segment – for example, it is sufficient to calculate LTP once for the entire image.

The result of applying LBP operator to the pixel matrix is a response matrix of values that characterize the brightness distribution in the neighborhood of the central pixel – the so-called bins. Based on the bin matrix, you can build an image histogram, which, unlike the brightness histogram, does not characterize the color distribution, but the image structure (Fig. 4).

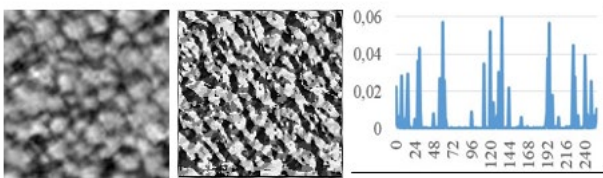


Fig. 4. Getting image histogram

Based on this, the task of searching for a forest on an image of the earth's surface can be done by calculating LBP histogram in the window mode and comparing it with the standard.

Another way to reduce the impact of noise, as well as to eliminate some of the texture details, is to introduce a threshold value  $T$  in the indicator condition. In this case, you can set three different values when building the code, taking into account the sign of the difference between the central pixel and neighboring ones. This method was presented under the name "local ternary pattern" (LTP).

In order to avoid an increase in the space of features, LTP is divided into two parts – the positive and negative patterns (Fig.5).

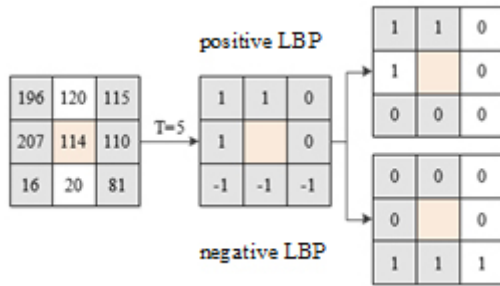


Fig. 5. Local ternary pattern

The dimension of basic LBP result can be reduced in two specific ways – using only so-called uniform patterns or patterns that are not sensitive to rotation of the pixel neighborhood.

Some binary codes have more information than others. Thus, a local binary pattern is called uniform if it contains no more than three series of "0" and "1" [7,11]. Uniform LBPs define only important local features of the image, such as line ends, faces, corners, and spots (Fig. 6), and also provide significant memory savings, i.e. the set of pattern  $s$  is reduced from  $2^p$  to  $(P(P-1)+2)$ .

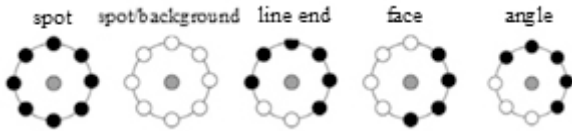


Fig. 6. Example of local features detected by LBP

## 5. Selection of algorithm characteristics

The maximum window scale was selected as  $1m/px$ , which corresponds to the capabilities of most types of modern satellite cameras [2] and makes it easier to link to the metric area.

The simplest way to determine the separability of forest texture classes is to cluster text images and then evaluate the result (Fig. 7).

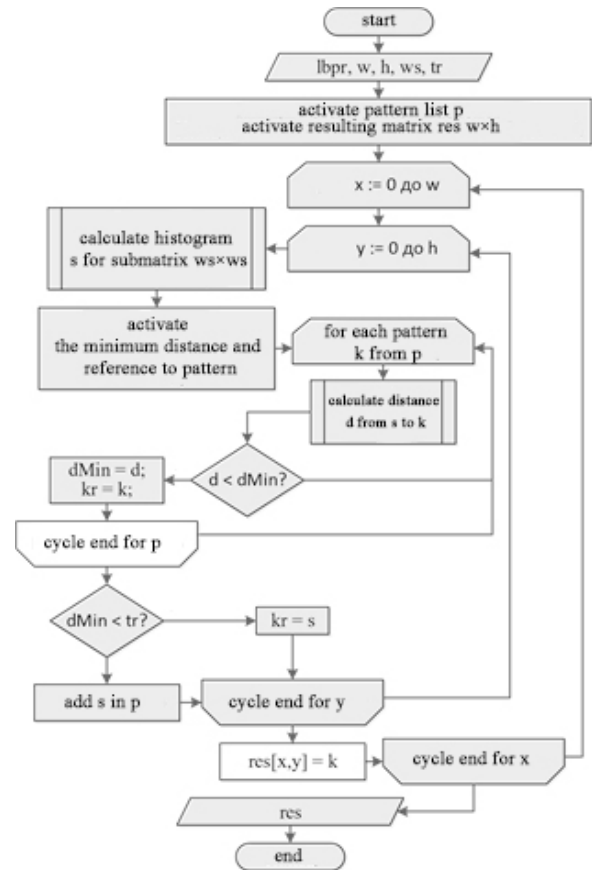


Fig. 7. Clustering algorithm, where lbpr is LBP histogram with the size  $w \times h$ ;  $ws$  is window size;  $tr$  is threshold

To preserve the possibility of comparing classification results with similar ones based on BPF coefficients, it was decided to process squares with a side length equal to number "2", which follows from the requirements of BDPF algorithm. According to the requirements, the window size range  $16 \times 16 - 32 \times 32$  was selected. If the window size is more than  $32 \times 32$ , the number of arithmetic operations per pixel becomes critical. Since the area of the common pine crown, on the basis of which some of the main comparisons are made, is 8-10 square meters,  $16 \times 16$  segment completely covers from 1 to 4 adult trees, which still allows to cover several trees in a sliding window.

The texture of the forest is heterogeneous, but the selection of multiple clusters for a forest area is the second condition for applying feature extractor, since one of the tasks being solved in the current work is the selection of forest stands of different species. The basic condition is a clear separation of the forest from other types of terrain. Since there is no need to allocate full-fledged clusters, so the simplest algorithm is used – clusters are allocated by a specified threshold, and the first pattern belonging to the cluster is used as the cluster kernel.

For comparison of texture patterns it is necessary to introduce a separability measure. There are many separability measures such as Euclidean distance, city block distance, divergence, and many others [2]. The most common measure in machine learning problems is Euclidean distance, i.e. the distance between two points in  $n$ -dimensional space [2,12]. Euclidean distance does not take into account the overlap of class distributions and is not applicable at a low level of separability (Fig.8), but

there are modifications of this measure that eliminate its disadvantages. One of these modifications is Mahalanobis measure.

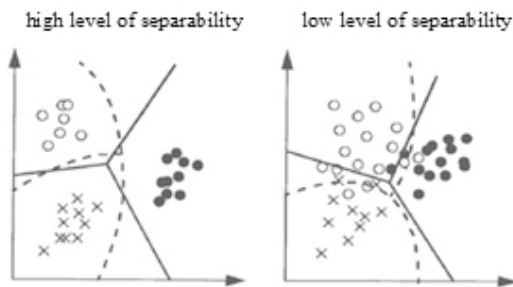


Fig. 8. Types of class separability [2]

Application of Mahalanobis measure makes sense for classification, but not for clustering, since the covariance matrix is calculated based on a training set for a class that does not yet exist.

One pixel is considered as a clustering unit. A certain area is captured around the pixel, for which a histogram is built and compared with standards. The cluster index is assigned to the center pixel.  $17 \times 17$  b  $33 \times 33$  are chosen as window sizes, according to the data provided above, approximately one and four trees per window.

In the course of checking the separability of classes, it was found that regardless of the algorithm characteristics for extracting features, structural features for different types of terrain can be almost indistinguishable. Fig.9 shows the result of selecting the threshold.

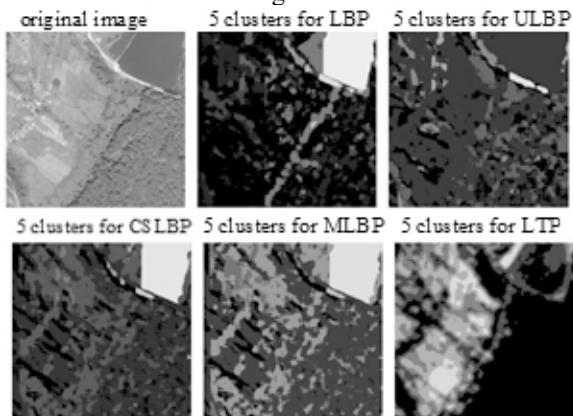


Fig. 9. Clustering results for window  $33 \times 33$

All the algorithms managed the task to some extent. LBP could not identify the forest, but it accurately identified the transitions between the main types of terrain. CSLBP and MLBP were able to separate the forest, while failing to separate the texture of the forest from that of the vegetable gardens. With the help of ULBP, it was possible to identify the main contours of forest stands, but the border lines (forest/field, forest/clearing) were excluded. The best result was achieved using LTP method, which accurately marked the contours of the forest and thickets near the road, while selecting them in one cluster with the buildings of the village.

Unlike other methods, LTP can be directly configured without using filters, binarization, etc. Fig. 10 shows the results of LTP allocation for various threshold values. At the threshold of 0, the terrain types are almost

indistinguishable. At threshold of 14, the forest, detached trees and buildings of the village are clearly distinguishable, but they are indistinguishable from trees. At the threshold of 50, only lake bridges and part of the road could be identified.

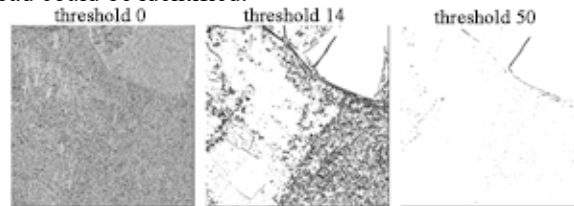


Fig. 10. LTP allocation for various threshold values

## 6. Classification tools

After pre-processing the image and subsequent selection of feature vectors, it is necessary to determine whether these vectors belong to any type of terrain, that is, to classify them. Classifying an object means specifying the number of the class that this object belongs to.

Previously described similarity measures - Mahalanobis distance and Euclidean distances - can be used to classify feature vectors based on standards, which was demonstrated when describing threshold clustering. This algorithm is easy to implement and to be scaled, but the linear dependence of the speed on the number of reference vectors makes it unacceptable within the framework of the described system.

There are many classification algorithms, and choosing a specific one is not an easy task. Determining the suitability of the classifier for working with the data formats used requires, at a minimum, the possibility to implement it for the selected development tools.

Making up training and test samples if there are no ready-made ones freely available is a long and time-consuming process.

Taking into account mentioned above, three classifiers with different specific features were selected based on the studied references. The first is a naive Bayes classifier for implementing a search based on a set of small classifiers. The second is a decision tree for optimizing classification based on similarity measures. The third is a multi-layer perceptron for processing large samples of data accumulated during the operation of the system. Since the perceptron was not fully introduced into the system, there is no description of it.

## 7. Naive Bayes classifier

Naive Bayes classifier (NBC) is a simple probability classifier based on Bayes theorem:

$$p(C_k \vee X) = \frac{p(X \vee C_k) \cdot p(C_k)}{p(X)}, \quad (6)$$

where:  $C_k$  -  $k$ -class;  $X = (x_1, x_2, \dots, x_n)$  is a size feature vector  $n$ ;  $p(C_k \vee X)$  is conditional (a posteriori) probability of belonging  $X$  to class  $C_k$ ;  $p(X \vee C_k)$  is conditional probability to find vector  $X$  in class  $C_k$ ;  $p(C_k)$  is unconditional (a priori) probability to meet class  $C_k$ ;  $p(X)$  is the probability of availability of vector  $X$  in the training sample.

The classifier is called "naive", because for an available set of features, it is assumed that the distribution of their values is independent of each other. Despite this

simplification, NBC in many cases shows itself no worse than more complex classifiers [5].

Since all vectors are represented in the sample with probability 1, the original formula is simplified to:

$$p(C_k \vee X) = p(X \vee C_k) \cdot p(C_k), \quad (7)$$

Given that the possible dependence of the probabilities of features occurring is not taken into account,  $(X \vee C_k)$  is calculated as the product of the probabilities of all features:

$$p(X \vee C_k) = \prod_{i=1}^n p(x_i \vee C_k), \quad (8)$$

To work with features-vectors of values, there is a modification of the classifier, that is, the so-called Gaussian naive Bayes classifier (GNBC).

Gaussian distribution is also called the normal distribution. The normal distribution graph is a bell-shaped curve that is symmetrical with regard to the average value (Fig. 11).

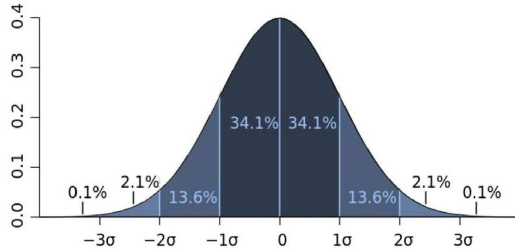


Fig. 11. Normal distribution graph

Due to the fact that to calculate the standard deviation, it is necessary to recalculate the mathematical expectations of features again (the mathematical expectation can be calculated based on the previous value, as opposed to  $\sigma$ ), NBC cannot be further trained in the course of work. Given the method of determining a priori probability, an important condition for correct NBC training is the statistical correspondence of the training sample composition to the composition of the studied data.

## 8. Decision tree

The task of monitoring the dynamics of changes in the forest area involves processing large amounts of information over a long period of time. This process actively uses classification tools, and it may be necessary to adjust the classifiers for different tasks. Training a classifier is a rather time-consuming process, since the main criterion for its success is the quality and volume of the training sample, which must be collected and provided with appropriate markers. To simplify this task, the system saves vectors of reference features and their source images to the database. This approach allows not only to reuse prepared class maps, but also organize classification based on the database without training. Classification based on the feature vector library belongs to the group of classification methods based on comparison with the standard [8]. The method of comparison with the standard involves the construction of a graph of feature vectors, while the classification process means finding the shortest path, which is based on the concept of edit distance – the minimum number of changes, inserts and losses required to change the image of A to the image of B [8]. The feature extractors described earlier give vectors of real numbers,

on the basis of which the edit distance cannot be calculated. However, they can also be reduced to binary attributes by setting requirements for the values of features – if  $a > n$ , then class A, and so on. In this case, the vector is simplified to a binary tree and comes into compliance with another common classification algorithm – the decision tree.

The decision tree training consists of selecting nodes based on a training sample, each of which is characterized by a feature vector attribute that most affects the outcome of the classification stage [6]. Node splitting occurs until the threshold probability is reached when the output value will take the required value.

In general, the condition for reaching these aims at  $i$ -level can be represented as follows [6]:

$$C_i = (Q_{i1} \vee Q_{i2} \vee \dots \vee Q_{ik}) \wedge (Q_{i1} \vee Q_{i2} \vee \dots \vee Q_{ik}) \wedge \dots \wedge (Q_{i1} \vee Q_{i2} \vee \dots \vee Q_{ik}), \quad (9)$$

where:  $Q_{ik}$  is logical required condition;  $i$  is a node level;  $k$  is the number of conditions.

Since the process is organized on the basis of reference feature vectors, the last node may hide a set of such vectors. At the same time, passing the tree to the end does not guarantee that the sample belongs to the described classes. At the final stage it is compared with the standards using Mahalanobis distance described earlier, which is used to make a conclusion about (not)belonging to the class. Covariance matrix is calculated for each class separately.

Fig. 12 shows the tree structure.

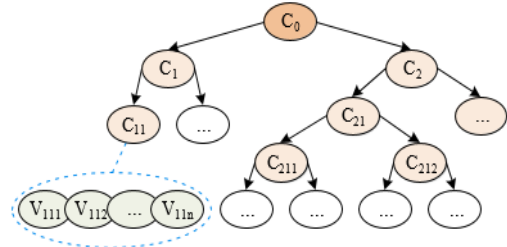


Fig. 12. Structure of hybrid decision tree

The advantage of the described algorithm is its high speed of relatively simple searching [6]. It is important to note that using a covariance matrix makes it impossible to update instantly during operation – features are added to the tree, but the matrix can only be recalculated in the background process.

## 9. Classification scenario

Previously, the advantage of using multiple algorithms for distinguishing features or classifying them together has been demonstrated. Not for all classification algorithms a non-uniform feature vector can be created. For example, when classifying by similarity, it is not possible to use LBP and the average values of RGB spectrum together, because LBP will give two hundred features, and the spectrum will give three features, which will have negligible effect on the result. To solve such problems, the concept of a classification/search scenario was formed (Fig. 13).

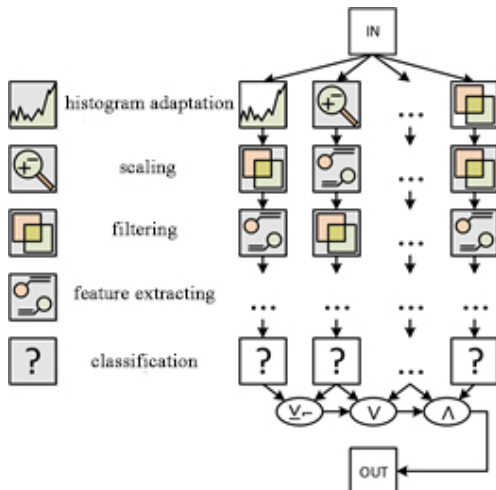


Fig. 13. Scheme of classification scenario

Classification scenario is a data structure that specifies the order in which images are processed by multiple algorithms. The resulting class maps are combined using logical operations. The scenario can also be used to describe one-dimensional algorithms. For example, the following selection of trees according to the scenario "median filter" - "spot selection (LBP)" - "center filtering".

### 10. Prediction of changes in the boundaries

Changes in forest boundaries can be caused by many factors, many of which are random. Events such as fires, deforestation, and disease outbreaks lead to rapid changes in the structure of plantings, with no pronounced periodicity.

Fig. 14 shows the boundary changes that need to be taken into account when developing the algorithm.

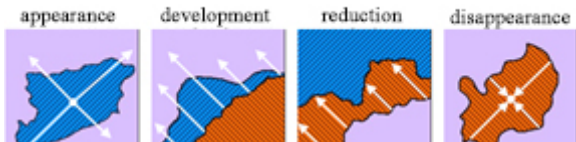


Fig. 14. Basic scenarios of boundary changes

Predicting function values with reference to a time interval requires the use of one-factor forecasting functions [8], but in the absence of a large sample, debugging such a solution is not possible. If we reduce the complexity of requirements for the forecast, methods that are easier to implement and debug become available, such as step-by-step extrapolation, where the time interval is the interval between sample events.

Under the assumption that the average level of the series has little tendency to change, we can assume that the predicted level is equal to the average value of the levels in the past [8].

The confidence limits for the average with a small number of observations are defined as follows:

$$\hat{y}_{t+j} = \bar{Y} \pm t_a S_{\bar{y}}, \quad (10)$$

where  $t_a$  is the table value  $t$  of Student statistic with  $n-1$  degree and probability level  $S_y$ .

The total variance associated with both the fluctuation of the sample average and the variation of individual

values around the average will be  $S^2 + (S^2/n)$ , where  $S$  is the standard deviation.

To predict borders, outlines are initially selected – the image pixels are bypassed in the cycle, the border pixels are found, and the array is saved. Then the array is bypassed and a segment is added for pixels whose distance is greater than the threshold (Fig. 15).

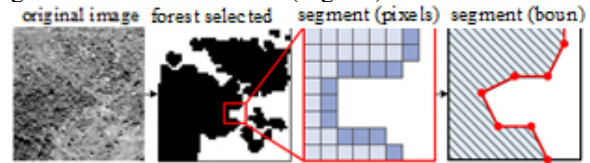


Fig. 15. Separating boundaries

After selection, the formula described is applied to the obtained points (junctions of segments). The shortest of the three segments is chosen as the direction of extrapolation -two to the two nearest previous points, and the third is the median of the resulting triangle (Fig. 16).

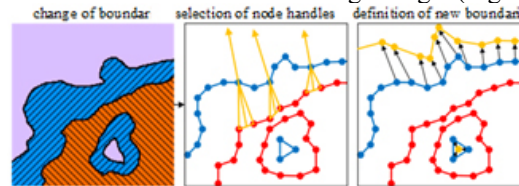


Fig. 16. Prediction of changes in boundaries by one step

### 11. Conclusions

Within the framework of this paper, a number of algorithms for processing and classifying graphical information were proposed to solve the tasks of studying forest stands based on images of the earth's surface.

LTP and FFT algorithms were selected as feature extractors, of which the simplest and most productive option is LTP, and the most complete and at the same time resource-intensive is FFT.

To pre-process the image, histogram equalization algorithms, median and Gaussian filters to eliminate noise and remove small image details were selected.

Euclidean distance was used as a measure of separability, Mahalanobis measure - for the purpose of classification. Czekanowski's quantitative index is also available in the system, which gives results similar to Euclidean distance, but with a different distribution of output quantities.

For classification, it was proposed to use a naive Bayes classifier, a simple but effective statistical classifier based on Bayes theorem. As a less specialized classifier that works without training on the basis of features stored in the database, the decision tree algorithm was proposed, an algorithm that significantly speeds up classification based on comparison with the standard by organizing feature vectors into a binary tree. A three-layer perceptron was also proposed as a test solution for working with large samples, but it was not possible to test it fully due to the large amount of training sample required.

These algorithms were described and tested. On their basis a set of libraries in C# language was developed, which form the described system together. MongoDB was chosen as the database, which is easy to develop and quite high – performance database that uses BSON documents as a storage format. A web service based on Asp.Net.Core

was developed to provide shared access to the system's tools. Its organization features are described in the project part.

## References

- [1] Moiseev N.A. On the state of forest use and the need to improve forest management. *Forestry Bulletin*, 2011, no.7.
- [2] Farutgin I.N. Technological solutions of ScanEx for receiving and processing satellite information. *Interexpo Geo-Sybir*, 2011. pp. 3-5.
- [3] Forests and land use: solution description. Planet Labs Inc., 2019. Available at: <https://www.planet.com/markets/forestry/>.
- [4] Strugailo V.V. Overview of digital image filtering and segmentation methods. Moscow Automobile and Road Construction State Technical University. pp. 270–281.
- [5] Kolesenkov A.N. Monitoring of subsurface use processes based on processing of aerospace images. *Izvestiya TulGU Technical Sciences*, 2018, no. 2.
- [6] Evdokimova N.I. Local patterns in the duplicate detection task. *Computer Optics*, 2017, vol. 41, no. 1. pp. 79-87 .
- [7] Polovinkin P.N. Detectors and descriptors of key points. *Image classification algorithms. The problem of detecting objects in images and methods for solving it*. Lobachevsky State University of Nizhni Novgorod. 30 p.
- [8] Olsen, R.C., S. Bergman, and R.G. Resmini. Target detection in a forest environment using spectral imagery. *SPIE 3118:4b*, 1997.
- [9] Ed. By Prasard S. Thenkabail, John G.Lyon, Alfredo Huete. *Hiperspectral remote sensing of vegetation*. – CRC Press, 2012.
- [10] Filippov R.A. *Internet of things: basic concepts: Tutorial*. Bryansk, BSTU, 2016. 112 p. ISBN 978-5-906967-62-6.
- [11] Averchenkov A.V. Development of a mathematical model of an information system for inventory and monitoring of software and hardware based on fuzzy logic methods. *Kachestvo. Innovatsii. Obrazovaniye.*, 2018, no.7. pp. 105-112. ISSN: 1999-513X
- [12] Leonov, Yu.A. Selection of rational schemes automation based on working synthesis instruments for technological processes/ YU.A. Leonov, E.A. Leonov, A.A. Kuzmenko, A.A. Martynenko , E.E. Averchenkova, R.A. Filippov - Yelm, WA, USA: Science Book Publishing House LLC, 2019 - 192 p. - ISBN: 978-5-9765-4023-1 - Text : unmediated.
- [13] Leonov E.A., Intellectual subsystems for collecting information from the internet to create knowledge bases for self-learning systems / E.A. Leonov, Y.A. Leonov, Y.M. Kazakov, L.B. Filippova/ In: Abraham A., Kovalev S., Tarassov V., Snael V., Vasileva M., Sukhanov A. (eds) - Text : electronic // *Proceedings of the Second International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’17)*. IITI 2017. *Advances in Intelligent Systems and Computing*. - 2017 - vol 679. - Springer, Cham, p. 95-103 - DOI:10.1007/978-3-319-68321-8\_10

## About the authors

Alexandr A. Kuzmenko, Bryansk State Technical University, Bryansk, Russia. E-mail: [alex-rf-32@yandex.ru](mailto:alex-rf-32@yandex.ru)

Dmitriy A. Kondrashov, Bryansk State Technical University, Bryansk, Russia. E-mail: [kuzmenko-alexandr@yandex.ru](mailto:kuzmenko-alexandr@yandex.ru)

Anna S. Sazonova, Bryansk State Technical University, Bryansk, Russia. E-mail: [asazonova@list.ru](mailto:asazonova@list.ru)

Ludmila B. Filippova, Bryansk State Technical University, Bryansk, Russia. E-mail: [libv88@mail.ru](mailto:libv88@mail.ru)

Rodion A. Filippov, Bryansk State Technical University, Bryansk, Russia. E-mail: [redfil@mail.ru](mailto:redfil@mail.ru)