E3S Web of Conferences **363**, 02025 (2022) https://doi.org/10.1051/e3sconf/202236302025
INTERAGROMASH 2022

Development of algorithms for localization of the areas depicted in the images of the Azov Sea, obtained using technical means of remote sensing, performing a comparison with the color scale and the GPS coordinate system

Elena Rakhimbaeva*, *Marcos* Porksheyan, and *Yulia* Belova

Don State Technical University, Rostov-on-Don, 34400, Russia

Abstract. The mathematical part of the algorithm for searching for objects in an image using additional classifying functions is presented. A software module has been developed in the Python programming language using the OpenCV library, designed to search for and highlight the boundaries of the coastal systems of southern Russia and build a condensing grid to the center from the boundaries. Experiments were carried out to test the performance of the software module under various weather conditions based on the Azov Sea.

1 Introduction

Until recently, the main source of data needed for the analysis and forecast of coastal and marine systems was the results of contact measurements of temperature, salinity, current velocities, etc. Their use far from the coast implies the use of a research fleet and large-scale expeditions. Along with this, with the development of satellite constellations, unmanned aerial vehicles, remote sensing methods have been actively used in the last 25-30 years. Modern space-based Earth remote sensing (ERS) systems occupy a significant place among space technologies. The needs of the national economy in the use of remote sensing images have recently increased significantly. They are used for meteorological observations, environmental monitoring, observations of the state of forest and agricultural land, and in many other tasks. The advantage of remote sensing systems is a wide coverage of the object of study, the ability to measure the parameters of the aquatic environment: the temperature rising of the near-surface layer, the distribution of salts in it for vast water areas almost simultaneously. Thus, it became possible to share remote sensing data, including satellite and mathematical modeling of hydrophysical and hydrobiological processes in coastal systems.

* Corresponding author: lena_rahimbaeva@mail.ru

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

2 Identification of the Azov Sea contour based on remote sensing data

Earth remote sensing (ERS) space systems make it possible to obtain images that represent cloud formations, land and water surface objects. To solve hydrometeorological problems, cloud formations are the subject of study, and for the tasks of mapping territories, studying natural resources, and others, clouds are interference and must be excluded from the process of processing video materials. In both cases, it becomes necessary to recognize and highlight cloud formations in images.

3 Mathematical description of the object search algorithm

The main part of the detection algorithm is the classifier, which is a statistical mathematical model that is automatically parameterized based on the training set through a process called machine learning [2].

The integral representation of an image is a matrix whose dimensions are the same as the dimensions of the image. We denote it by the formula (1):

$$A=a_{ij}. \tag{1}$$

The elements of this matrix are the sums of the brightnesses of all pixels located above and to the left of this element, Figure 1. The sum can be calculated using the formula (2):

$$a_{xy}=\sum_{i=1,\,j=1}^{i\le x,\,j\le y} f(i,j), \tag{2}$$

where the function $f(x,y)$ is defined by the formula $f(x_i\ ,y_i),x_i\in[1,n]\cap \mathrm{N},y_i\in[1,m]\cap \mathrm{N}$.

Thus a_{ij} – sum of pixels in a rectangle $[1,i]\times[1,j]$.

An example of amount calculating is the formula (3):

$$S=A(E)-A(D)-A(B)+A(C), \tag{3}$$

where $A(E)$, $A(D)$, $A(B)$, $A(C)$ – matrix A elements with coordinates E, D, B, C respectively.

Fig. 1. Rectangular area to demonstrate the formula for the sum of brightnesses

Thanks to this, it is possible to calculate the area of a window of any size in the same time.

The task of any simple (weak) classifier is that after the formation of feature sets, they must be assigned to some class. It divides all classified features and associates them with a specific class.

An example of such a classifier is shown in the figure 2.

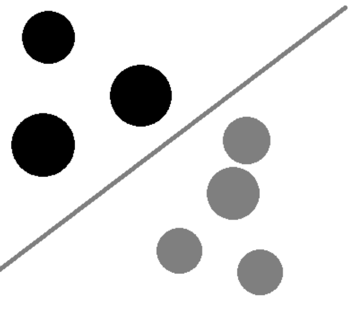

Fig. 2. Simple classifier

A straight line can serve as a simple example of a classifier, but with its help it is impossible to correctly split the set shown in Figure 3 into two classes [3].

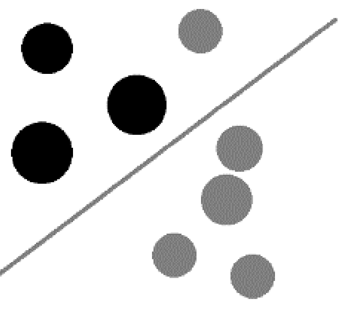

Fig. 3. Set of two classes

Therefore, another task is posed - to produce a partition with a minimum error.

To achieve maximum accuracy, it is necessary to combine the functions into a single unit, called a "strong classifier". This procedure is called classifier boosting and is based on the AdaBoost algorithm.

Mathematically, this procedure can be formulated as follows: it is necessary to construct a classifying function F: X → Y, where X is the set of feature vectors, and Y is the set of labels.

Let there be a training sample $(x_1, y_1),\ldots,(x_N, y_N)$, where $x_i\ (i=1..N)$ – vector - attribute, a $y_i\ (i=1..N)$ – the label of the class to which it belongs $x_i \in X$.

Let there also be a family of simple classifying functions, where $H=\{F : F : X \to Y\}$ − classifying function [4].

We will look for a classifier in the form of a formula (4).

$$F(x) = sign\left[\sum_{i=0}^{M} a_i h_i(x)\right], \tag{4}$$

where $a_i \in \mathrm{R}, h_i(x) \in H, M$ – number of weak classifiers.

Let us construct an iterative process in which at each step we will add a new term. The iterative process can be constructed according to the formula (5).

 https://doi.org/10.1051/e3sconf/202236302025

$$f_i = a_i h_i(x). \tag{5}$$

So $(x_1, y_1)\dots(x_N, y_M)$ – training set, $D_0(m) = \frac{1}{N}$ – initial «weight». For each step $i = 1\dots M$:

- the best "weak classifier" for the current D_i represented by the formula (6):

$$h_i = \arg\min_{h\in H}\left(e_i = \sum_{m=1}^{N} D_i(m) * \left(h(x_m) \neq y_m\right)\right), \tag{6}$$

where e_i – weighted classifier error.

- calculate $a_i = \frac{1}{2}\log\frac{1-e_i}{e_i}$;
- memorize $f_i = a_i h_i(x)$;
- update the formula (6) and get a new result, which is visible in the formula (7):

$$D_{i+1}(m) = \frac{D_i * e^{-y_m f_i(x_m)}}{Z_i}, \tag{7}$$

where Z_i – normalizing coefficient, so we replace our sum with the formula (8):

$$\sum_{m=1}^{N} D_{i+1}(m) = 1. \tag{8}$$

Compiling $F(x)$ according to the formula (9):

$$R = w(-1,-1)f(x-1, y-1) + w(-1,0)f(x-1, y) + \dots +$$
$$w(0,0)f(x,y) + \dots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1). \tag{9}$$

$D_i(m)$ at each step sets the significance of the m-th sample for the next step of learning the algorithm. The larger this value, the more the algorithm will try to classify the example correctly at this stage. The more accurately an example is recognized in the previous steps, the less weight it acquires. The meaning of these actions is that we vary the weights in such a way that the classifier included in this step concentrates on examples that were poorly handled in previous steps. [5].

To improve the accuracy of recognition, additional classifying functions are added, combined into a single whole, called a cascade house. Such a cascade will consider a region eligible and containing an object if each function matches the region with a class denoting the presence of the object, otherwise the region is rejected. [6].

4 Algorithm for detecting objects on an image

An image is being read from a file. Next, the image needs to be converted to grayscale (Figure 4). A so-called window moves across the image - a smaller scan, a classifier is applied for each position of the window and fixes the position and size of the object. The output of the classifier is an array of coordinates and sizes, (x,y,w,h), where (x,y) are the coordinates of the upper left corner of the frame, and (w,h) are the width and height of the frame, respectively. Further in the cycle, the frame is drawn along the contour of the Sea of Azov. The image is converted to color mode and displayed on the screen [7].

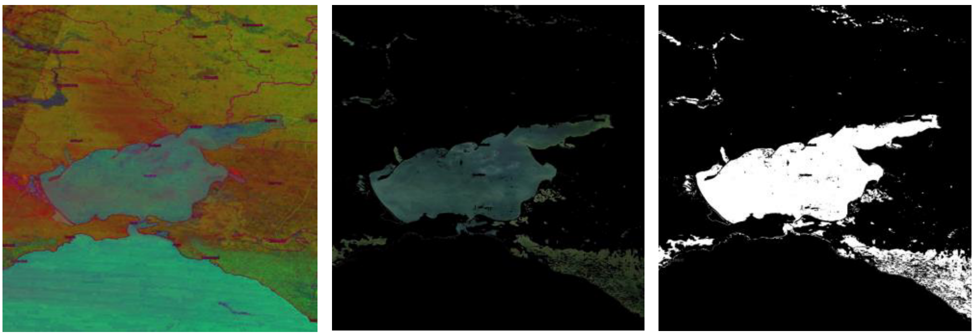

Fig. 4. Running the software before outputting the desired result

5 Results of the software tool

A software tool was developed to identify the boundaries of the coastal systems of the South of Russia (Azov Sea) and build a denser grid [8] to the center from the boundaries.

Several experiments were carried out for the Sea of Azov. All satellite images date from different days and are taken from the website https://worldview.earthdata.nasa.gov/ .

Fig. 5. Operation of the program module under favorable weather conditions

Experiments were carried out to test the performance of the software module under various weather conditions based on the Sea of Azov.

The first situation is when the sea area is still clearly visible, however, there is a slight cloud cover and a significant presence of phytoplankton masses (Figure 5).

The second situation considered, when the sea area in some places is practically invisible, there are significant clouds and a significant presence of biomass (Figure 6).

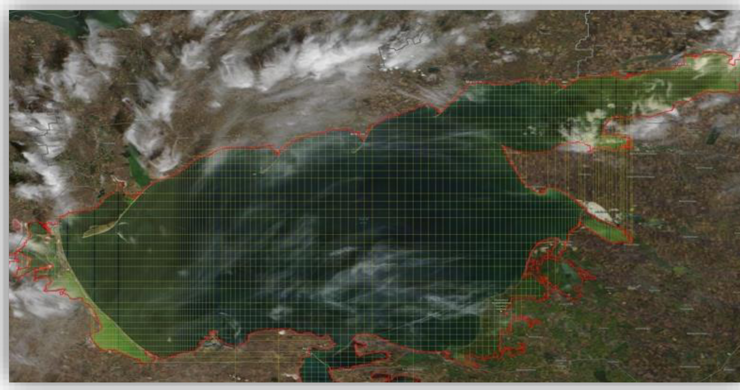

Fig. 6. The operation of the software module with pronounced interference in the form of clouds and a significant presence of biomass

6 Conclusion

The main direction of the development of the work can be called the improvement of image processing methods for determining the boundaries of the Azov Sea. In the future, it is planned to develop a software tool based on mathematical modeling of the dynamics of the development of phytoplankton populations using remote sensing of the coastal systems of the Azov Sea.

Acknowledgments

The study was supported by the Russian Science Foundation grant No. 22-71-10102, https://rscf.ru/project/22-71-10102/

References

1. P. Viola, M.K. Jones, Rapid Object Detection using a Boosted Cascade of Simple Features. Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2001)
2. R.S. Gonzalez, Digital Image Processing (Technosphere M., 2012)
3. L.M. Mestetsky, Mathematical methods of pattern recognition (MGU VMiK, M., 2004)
4. J. Sochman, Matas J. AdaBoost. Center for Machine Perception (Czech Technical University, Prague, 2010)
5. Y. Freund, R.E. Schapire, A Short Introduction to Boosting (Shannon Laboratory, USA, 1999)
6. T. Ojala, M. Pietikäinen, D.A. Harwood, Pattern Recognition, **29** (1996)
7. T. Ahonen, A. Hadid, M. Pietikäinen, IEEE Trans. Pattern Analysis and Machine Intelligence, **28(12)**, (1996)
8. A.A. Samarsky, Introduction to the theory of difference schemes (Nauka, M., 1989)
9. B.J. Williams, Hydrobiological modelling (University of Newcastle, NSW, (2006)