Одним з найбільш ефективних способів підвищення точності і швидкодії алгоритмів розпізнавання є попереднє виділення зон інтересу на аналізованих зображеннях. Досліджено можливість застосування самоорганізуючих карт і нейронної мережі Кохонена для визначення зон інтересу на радіолокаційному або супутниковому зображенні підстильної поверхні. У знайдених зонах інтересу велика ймовірність виявлення об'єкта, що цікавить, для подальшого аналізу. Визначення зон інтересу необхідно перш за все для автоматизації та прискорення процесу пошуку і розпізнавання об'єктів, що цікавлять. Це, в силу постійно наростаючої кількості супутників, стає все більш доцільним. Представлено процес моделювання, аналіз і порівняння результатів застосування даних методів для визначення зон інтересу при розпізнаванні образів літальних апаратів на тлі підстильної поверхні. Також описано процес попередньої обробки вхідних даних. Освячено загальний підхід до побудови та навчання самоорганізуючої карти і нейронної мережі Кохонена. Застосування карт і нейронної мережі Кохонена дозволяє в 15–100 разів зменшити обсяг даних, що аналізуються. Це, відповідно, прискорює процес виявлення і розпізнавання об'єкта, що цікавить. Використання наведеного алгоритму істотно скорочує необхідну кількість навчальних образів для згортальної мережі, здійснює остаточне розпізнавання. Зменшення навчальної вибірки обумовлено тим, що розмір частин, що подаються на згортальну мережу вхідного зображення, прив'язаний до масштабу зображення і дорівнює розміру найбільшого об'єкта детектування. Нейронна мережа Кохонена показала себе більш ефективною відносно до даної задачі, так як рідше розміщує центри кластерів на підстильній поверхні в силу незалежності ваги нейронів від сусідніх центрів. Дані технічні рішення можуть застосовуватися при аналізі візуальних даних із супутників, літальних апаратів і безпілотних автомобілів, в медицині, робототехніці і т. д.

Ключові слова: розпізнавання образів, самоорганізуючі карти, нейронна мережа Кохонена, радіолокаційні та супутникові зображення, робототехника

UDC 004.931
DOI: 10.15587/1729-4061.2019.166887

APPLICATION OF KOHONEN NEURAL NETWORKS TO SEARCH FOR REGIONS OF INTEREST IN THE DETECTION AND RECOGNITION OF OBJECTS

V. Skuratov
Engineer, Researcher
Joint-stock company All-Russian Scientific Research Institute of Radio Engineering
Bol'shaya Pochtovaya str., 22, Moscow, Russian Federation, 105082
E-mail: viktor.skuratov@gmail.com

K. Kuzmin
Researcher
Department of Mathematical and Instrumental Methods in Economics
University of Russian Innovation Education
Krasnobogatyrskaya str., 10, Moscow, Russian Federation, 107061
E-mail: konstantin.alexandrovich@yahoo.com

I. Nelin
PhD, Associate Professor
Department of Radiolocation, Radio Navigation and On-Board Radio Electronic Equipment
Moscow Aviation Institute
Volokolamskoe highway, 4, Moscow, Russian Federation, 125993
E-mail: nelin.iv@yandex.ru

M. Sedankin
PhD, Senior Researcher
Main Research and Testing Robotics Centre of the Ministry of Defence of the Russian Federation
Seregina str., 5, Moscow, Russian Federation, 125167
State Research Center – Burnasyan Federal Medical Biophysical Center of Federal Medical Biological Agency
Marshala Novikova str., 23, Moscow, Russian Federation, 123098
E-mail: msedankin@yandex.ru

1. Introduction

At present, the low speed of image recognition is one of the main problems hampering the development of modern visual data processing systems. Increasing the speed of image recognition will lead to an increase in the performance of satellite and radar image analysis systems, medical images, data from robotic systems and unmanned vehicles, etc., having both technological and economic effects.

One of the ways to improve accuracy and speed of recognition algorithms is application of neural networks (NN). The most modern type of NN used in image recognition is the convolutional neural network (CNN), which got its name because of the presence of a convolution operation [1]. The size

Received date: 06.03.2019 Accepted date: 03.05.2019 Published date: 26.06.2019

of an object being recognized should be comparable to the size of objects in the training set for high-quality operation of a network of this type. Obviously, creation of a training set in which, in addition to the various search object types, there is also a variation in the scale of the object in the image, is extremely difficult. Training of such a network will take considerable time and computational resources. A number of errors in recognition will increase, and speed of an algorithm will be low. The algorithm presented in this paper allows us to reduce the number of data analyzed by convolutional NN when detecting objects in an image by 15–20 times and, accordingly, reduce the search time for necessary objects, increasing recognition accuracy and saving computational resources. This is achieved by using the network and or Kohonen map to determine regions of interest (ROIs) on the radar or satellite image of the underlying surface on the input image, in which the probability of detection of the desired object is high. This paper discusses the application of the developed algorithm for detecting and recognizing the type of aircraft (LA) on satellite images.

2. Literature review and problem statement

Today, there are a number of works [2–5], which investigated application of NN, a network (Kohonen neural network) and Kohonen self-organizing map (SOM) for image recognition. Some authors consider Kohonen networks and maps for preparation of input data with subsequent analysis by other NN, and others – for direct image recognition. Paper [2] presents the results of NT simulation for image recognition. Image segmentation by Kohonen maps occurs first, and then analysis in a hybrid NN takes place. Kohonen maps served to reduce a training set for a hybrid NN and to speed up the analysis. Work [3] presents Kohonen network for recognition of color images. A significant disadvantage [2, 3] is that an image for recognition is the entire image for classification and, the recognition will not occur if the scale of the required fragment will change in the input image. Paper [4] gives an algorithm to search a fragment in a series of color images using the Kohonen network. There are image fragments obtained by the «window», which passes through the analyzed image pixel by pixel, sequentially fed into the network for analysis. The size of the «window» corresponds to the size of the sought for fragment. Search for a fragment the size of 211×169 pixels in the image the size of 621×497 pixels lasted 122 seconds. There were 135219 histograms analyzed during this time. The time spent for analysis of one image does make it possible to apply the proposed algorithm for recognition of large amounts of data. In addition, the analysis of one image requires serious computational resources, because the algorithm divides an image into 135,219 fragments for analysis. Work [5] assesses effectiveness of application for classification of images of a hybrid network. Firstly, there goes the analysis by a Kohonen network, and then the data goes to a perceptron network. The work shows that the application of network configuration of such configuration give possibility to reduce computational complexity of algorithms used. We can reduce complexity, because the application of network configuration of such configuration reduces the required number of neurons by more than 2 times, which also makes it possible to reduce the classification error by 4–5 times.

Application of Kohonen network and Kohonen maps to highlight various characteristics of objects in images is of particular interest. Authors of work [6] used Kohonen maps to solve complex clustering problems with a large number of objects and to select objects with unusual characteristics among them. Authors of paper [7] used a Kohonen network to search for cluster centers by finding of the minimum Euclidean distance between points, which belong to a cluster.

The most modern type of NN used in image recognition is convolutional NN, which got its name because of the presence of a convolution operation. The essence of the convolution operation is that each image fragment is multiplied by a core of convolution by each element, and the result is summed up and written to the similar position of an output image. Then, the result goes to a subsampling layer and a fully connected neural network in the simplest case. A paper [8] presents a structure of a convolutional NN and its application for image recognition, and a paper [9] – for analysis of a video sequence.

Papers [10–14] describe development of the R-CNN (Region-based Convolutional Network) algorithm for detection of objects in images. Papers [10, 11] describe the initial version of the R-CNN algorithm. The algorithm highlights about 2,000 regions in an input image. It scales each of regions using an affine transformation. Then, it goes to the input of a convolutional network, which extracts a vector (map) of features. Article [11] describes a training of a linear regression to clarify coordinates of an object window additionally. Such a refinement of the object localization gives possibility to increase quality by 3–4 %. R-CNN has several drawbacks, mainly due to high time spent on network training, as well as on direct image processing by the algorithm. It is necessary to train a convolutional neural network in two stages and to train linear regressors for each class. Processing of one image takes about 47 seconds. Fast R-CNN article [12] is continuation of papers [10, 11]. The authors propose to supply a full image to the input of NN. There is a feature map generated for each selected image region in R-CNN. There is a feature map generated for the entire image, and then there are maps for each region determined in Fast R-CNN. This can significantly reduce detection time. The following Faster R-CNN article [13] proposes to replace the procedure for generation of ROIs with a separate convolutional NN, which extracts features from an image and determines the expected boundaries of ROIs. A sliding window bypasses the feature map of an input image, and there is a feature vector of small dimension extracted for each position of the window. The resulting vectors go to the input of two layers of NN. One of these layers serves to clarify boundaries of a region, and the other one serves to classify an object located within this region. There are several options of boundaries of regions, which have a different size or different side ratio, considered simultaneously for each position of a sliding window. Mask R-CNN [14] develops the Faster R-CNN architecture by addition of ability to predict a position of a mask, which covers a found object.

Article [15] describes the YOLO (You Only Look Once) algorithm. It gives a possibility to detect and recognize objects in images in 103 times faster than R-CNN and by 102 times faster than Fast R-CNN, but with lower accuracy. This algorithm imposes a grid on an input image and divides it into cells. The algorithm determines bounding frames of a zone of possible location of objects with an estimate of detection accuracy and probability of belonging to classes around each cell. Then the accuracy estimate for each zone is multiplied by the class probability and we get the final value of detection probability. The algorithm analyzes from several thousand to several tens of thousands parts of an image with restrictive frames of different sizes.

Work [16] presents SSD algorithm: Single Shot MultiBox Detector, which is comparable to YOLO in accuracy and speed. The algorithm overlaps the entire area of an input image with bounding frames. The size of frames varies within the established boundaries. It gives a possibility to detect and recognize objects of various sizes. There goes estimation of the probability of belonging to classes and correction of frame sizes for each part of an image highlighted by a frame. The algorithm performs analysis from several thousand to several tens of thousand parts of an image.

It is necessary to apply algorithms, which divide an input image into a set of images that are suitable in size for analysis in a convolutional NN for detection and recognition of objects on a radar or satellite image. The algorithm used should reduce an amount of data required for further analysis to save time and resources. It is possible to implement such an algorithm in the following ways:

1) sequential division of an input image into frames of the size required for a convolutional NN with a frame shift by a certain number of pixels relative to the previous one;

2) application of R-CNN, Fast and Faster R-CNN, Mask R-CNN, YOLO, and SSD algorithms.

Both presented methods require a serious investment of time and computational resources. In one case, each input image divides into several thousand frames which are submitted for analysis into the subsequent convolutional NN [4, 10, 11]. In the other case, the algorithm analyses an image entirely with enumeration of possible boundaries of ROIs, without first sifting out areas, which are not of interest [12–16].

Thus, there is a need for algorithms, which can reduce an amount of data analyzed, and thereby increase speed of recognition systems based on application of convolutional networks.

3. The aim and objectives of the study

The objective of the study is to create an algorithm for detection of ROIs in recognition of a type of aircraft using a convolutional neural network on radar and satellite images of the underlying surface.

We set the following tasks to achieve the objective:

– detection of boundaries of objects present in an image;

– search for centers of objects;

– verification of the selected algorithms on real input data, determination of ROIs on images with application of the algorithms and comparison of the results using a Kohonen self-organizing map and a neural network in search for object centers.

4. The algorithm for detection of boundaries of objects located on the underlying surface

It is necessary to perform preliminary processing before supplying input data into a convolutional NN for analysis.

There is always a boundary between the body of an aircraft and the underlying surface. It forms due to differences in brightness or illumination levels, since the reflectivity of an aircraft is usually higher than the reflectivity of the underlying surface. In addition, in most cases, the body of an aircraft casts a shadow on the underlying surface, which also forms a boundary. If we mark these boundaries around AC with dots in an input image, then the center of this cluster will actually coincide with the center of an aircraft.

The algorithm for detection of boundaries of objects located at the underlying surface is as follows:

1) conversion of an input color image into shades of gray;

2) application of the Sobel operator – a differential operator, which calculates an approximate value of the image brightness gradient;

3) conversion into a binary image using scissoring by the brightness threshold. The obtained brightness boundaries take values 1, all others – 0;

4) removal of objects of small size.

Table 1 presents a comparison of performance of the algorithm for detection of boundaries of objects depending on the resolution of an input image and a number of possible objects of the minimum size. We performed calculations in a proprietary program without using GPU (graphics processor), acceleration and optimization algorithms on a personal computer with the following parameters: Intel Core i7 processor – 6,700 K, 4 GHz, 32 GB RAM.

Table 1

Comparison of operation time of the algorithm for detection of object boundaries

Size of an input image, pixel	112×218	202×387	767×778
Number of objects sought for, pcs	3	10	64
Preliminary processing, s	0.002	0.003	0.02

As we can see from Table 1, the presented algorithm has a sufficiently high speed. It detects boundaries of objects of an input image in 0.002–0.02 seconds.

5. The methodology for finding the centers of possible ROIs at images

To reduce the time required for an analysis and necessary computational resources in detection of objects of interest at an input image, we propose to select zones of their possible location with further sequential analysis of found zones in the convolutional NN. Thus, we will save time and resources by reducing of a number of frames analyzed and, consequently, increase speed of the system as a whole.

We suggest using the self-organizing maps or a Kohonen neural network to search for centers of possible ROIs after the initial processing of the input image.

Kohonen neural networks are a class of self-organizing NNs based on a Kohonen layer, which consists of a number of parallel adaptive linear elements. After passing linear elements, the processing of signals occurs according to the «Winner Gets Everything» principle. The largest signal gets a single value, and the rest are reset.

NN training learning without a trainer is necessary (no training sample) to solve the problem of allocation of centers of ROIs in an input image, since processing of each input image should occur independently of the previous ones. The main types of Kohonen networks, which use unsupervised training, are [17, 18]:

– Kohonen network for vector quantization of signals;

– Kohonen self-organizing maps.

5. 1. Kohonen network for vector quantization of signals

The Kohonen layer consists of a number of n parallel linear elements, which have the same number of inputs and receive the same vector of input signals $\mathbf{x}=[x_1, x_2,..., x_i]$ at their inputs. We get the signal at the output of j-th linear element:

$$y_j = \omega_{j0} + \sum_{i=1}^{m} w_{ji} x_i, \tag{1}$$

where w_{ji} is the synaptic weight coefficient of i-th input of j-th neuron, i is the number of an input, j is the number of a neuron, ω_{j0} is the threshold coefficient.

After passing through the layer of linear elements, there goes processing of signals according to the «Winner Gets Everything» rule. The search for the maximum $\mathbf{y}_j$ proceeds among output signals: its number is $i_{\max}=\mathrm{argmax}(\mathbf{y}_j)$. The signal with $i_{\max}$ number is equal to one at the output, the rest are equal to zero. If the maximum is reached simultaneously at outputs of several adders, then a single value is assigned to one of them according to the established rule, for example, the first one. Training of Kohonen network is a process of selection of weight values which minimize errors due to replacing of input vectors with a weight vector, which is close in the sense of the used metric. We call such an approach the vector quantization [19]. We can apply it, for example, in problems of compression of audio and video signals. The very idea of vector quantization consists in a compact representation of multidimensional input vectors using a limited set of support vectors of a lower dimension, which form a code table. Input vectors are encoded with numbers of winning neurons (cluster numbers) in a Kohonen network. Thus, all vectors from a certain area of an input space are replaced with one support vector, which is their nearest neighbor. The measure of closeness between objects is called distance. Kohonen networks usually use Euclidean distance:

$$d_E(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} = \|\mathbf{x}-\mathbf{y}\|. \tag{2}$$

The Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ vectors is the Euclidean norm of a difference of vectors, or a length of the segment, which connects x and y points. We apply competition mechanisms for training of Kohonen network «without a trainer». A neuron with the weight vector, which is less different from the input vector, wins when applied to the input of $\mathbf{x}$ vector. It is necessary to perform initialization of a network before the training process. It is the initial task of weights vectors. In the base case, we set random weight values. The training process of a Kohonen network consists of a cyclic repetition of the following steps:

1. Supplying of source data to network inputs.
2. Finding the output values for each neuron.
3. Definition of a winning neuron with weights, which are less different from corresponding components of the input vector.
4. Correction of weights of a winning neuron according to Kohonen rule:

$$w_j^{(k+1)} = w_J^k + \eta_j^k \left(\mathbf{x} - w_j^k\right), \tag{3}$$

where $\mathbf{x}$ is the input vector, k is the number of a training cycle, η_j^k is the coefficient of training rate for j-th neuron in k-th training cycle.

5. Going back to step 1 if training is not completed.

Consequently, the neuron with weights vector, which is closer to the input vector, is updated to become even closer. As a result, this neuron is likely to win the competition when applying a close vector to the input and loses when it is significantly different. After multiple supplies of training vectors to the network input, there will be a neuron that produces 1 when the vector belongs to the cluster, and 0 when the vector does not belong to it. Thus, the Kohonen network trains to classify input vectors.

5. 2. Kohonen self-organizing maps

The main purpose of self-organization maps is to transform vectors, which come to the input, with arbitrary dimensions into a one- or two-dimensional discrete map with a topologically ordered form [18]. The basis of maps, like the Kohonen network, is competitive training. Neurons of the output layer compete for the right to activate, and only one output neuron, the winning neuron, is active. One of the ways to organize this type of competition between neurons is to use negative feedbacks between them. In general, neurons are located at nodes of a two-dimensional grid with rectangular or hexagonal cells (Fig. 1) and interact with each other in SOM. r_n distance between neurons on a map determines a magnitude of this interaction. The distance between individual neurons coincides more with the Euclidean distance for a hexagonal grid, than for a quadrilateral one. The number of neurons in the grid determine a degree of details in the result of network operation and accuracy of the generalizing ability of a map depends on it.

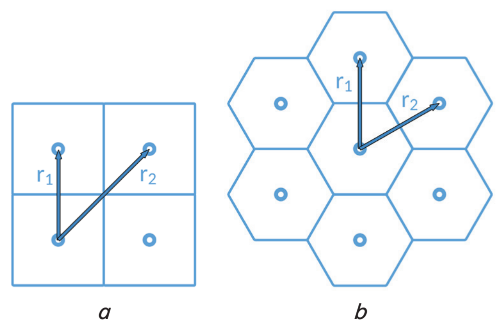

Fig. 1. The initial location of neurons in nodes of a two-dimensional grid with cells: *a* – quadrangular; *b* – hexagonal

Positions of neurons determined during the competitive process (that is, winning neurons) are ordered relatively to each other in such a way that a significant coordinate system appears on a grid. Thus, SOMs form topographic maps of input images, where spatial location (coordinates) of neurons of a grid is an indicator of statistical features contained in images. The spatial position of output neurons in a topographic map corresponds to a specific region of characteristics of data extracted out of the input space. Hence the very name – «self-organizing maps».

The attribute map has the following properties:

1. Approximation of the input space. The features map represented by a set of vectors of synaptic weights w_j performs approximation of the input space in the output space. The main objective of the SOM algorithm is to describe a large volume of input space vectors by definition of a small set of prototypes in the output space, which is an adequate approximation of the original input space.

2. Topological order. The feature map obtained after application of the SOM algorithm is topologically ordered, since the spatial position of neurons in a grid corresponds to a specific region or feature of an input image.

3. Compliance of density. A feature map reflects changes in statistics of distribution of the input signal. Areas of an input image, from which vectors are taken with a higher probability, are mapped to much larger areas of the output space and with a higher resolution than areas, from which vectors are taken with a lower probability.

4. Selection of features. SOM is able to extract a set of best features for approximation of distribution being studied for data from the input space with a non-linear distribution.

Each of input images usually consists of a localized area of activity located in a relatively quiet area. Location and nature of such an area varies depending on a type of input example. Operation of the SOM algorithm usually begins with the initialization of synaptic weights of a network. There are three basic processes launched to form a map after correct initialization of a network. These processes are: competition, cooperation, and synaptic adaptation.

1. Competition. Neurons of a network calculate relative values of the discriminant function, which is the basis for the process of competition among neurons, for each input image.

It is necessary to compare $w_j^{\mathrm{T}}\mathbf{x}$ scalar products for each neuron and choose the largest value in order to find the best w_j vector, which corresponds to $\mathbf{x}$ input vector. In addition, each neuron has a certain saturation value equal to the threshold taken with the opposite sign. A choice of a neuron with the largest scalar product determines coordinates of a center of a topological area of an excited neuron.

The best matching criterion, based on maximization of $w_j^{\mathrm{T}}\mathbf{x}$ scalar product, is mathematically equivalent to minimization of the Euclidean distance between $\mathbf{x}$ and w_j vectors. If to use $p(\mathbf{x})$ index to identify a neuron, which corresponds to $\mathbf{x}$ input signal mostly, then we can determine this value using the following relationship:

$$p(\mathbf{x}) = \arg\min_j \left\| \mathbf{x} - w_j \right\|, \quad (4)$$

where $\| \ \|$ is the Euclidean norm.

A specific neuron, which satisfies this condition, is called the winner for $\mathbf{x}$ input vector.

2. Cooperation. The winning neuron determines location of a topological area of neurons of a network, which ensures cooperation between these neurons. The winning neuron is located in the center of a topological area of cooperating neurons. This condition leads to determination of the topological area of i winning neuron. It decreases gradually with an increase in a distance. Let us denote the topological area with a center in the winning neuron as h_{ji}. It consists of a set of excited (cooperating) neurons. An example of h_{ji}, which satisfies the mentioned conditions, is the Gauss function:

$$h_{ji(\mathbf{x})} = \exp\left(-\frac{d_{ji}^2}{2\sigma^2} \right), \quad (5)$$

where $d_{ji}^2 = \| r_j - r_i \|^2$ is the distance between j-th neuron and the current winning neuron on a plane, σ is the effective width: a numerical parameter, which sets a size of the area around the winning neuron where correction of weights occurs. The smaller is σ, the fewer are neurons from the winning's neighborhood influenced by $\mathbf{x}$ input vector.

Another feature of the SOM algorithm is that a size of the topological area decreases with time. This requirement is satisfied due to gradual decrease in σ effective width of the function of h_{ji} topological area. A popular variant of the dependence of σ on discrete values of n time is the exponential decrease described by the following formula:

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1} \right), \quad (6)$$

where σ_0 is the initial σ value in the SOM algorithm, τ_1 is the parameter responsible for the rate of decrease in $\sigma(n)$ function.

3. Synaptic adaptation. This mechanism makes it possible for excited neurons to increase their own values of discriminant functions with respect to the input vectors by adjusting of the synaptic weights. Corresponding adjustments are made to make the response of the winning neuron to the subsequent arrival of similar examples to strengthen. For discrete time of $w_j(n)$ vector of synaptic weights at n time, w_j, we can define $(n+1)$ updated vector at time $n+1$ as follows:

$$w_j(n+1) = w_j(n) + \eta(n) h_{ji(\mathbf{x})}(n) \left(\mathbf{x} - w_j(n) \right), \quad (7)$$

where $\eta(n)$ is the training rate parameter; $h_{ji(\mathbf{x})}(n)$ is the function of the neighborhood with a center in $i(\mathbf{x})$ winning neuron. Both of these parameters change dynamically during training in order to obtain the best result. This expression applies to all neurons of a network that lie in the topological neighborhood of the winning neuron. Expression (7) has the effect of shifting the vector of synaptic weights of the winning neuron towards the input vector. Consequently, vectors of synaptic weights will tend to follow distribution of the input vectors, due to adjusting in the neighborhood of the winning neuron.

Based on 5. 1 and 5. 2, we can conclude that it is possible to use the Kohonen neural network and SOM in search for centers of areas of ROIs, as they are able to determine centers of clusters of input data. When comparing results of the search for cluster centers by the described method with the algorithms of fuzzy centers and subtractive clustering, neural networks turn out to be significantly faster.

6. Description of the algorithm that searches for ROIs

There is a color image supplied to the input of the algorithm for analysis. Next, there goes its preliminary processing carried out in several stages:

1. Detection of boundaries of objects presented in the image. We applied the algorithm based on the use of the Sobel operator described in § 4 for this purpose.

2. Search for centers of objects. We can use a neural network or Kohonen self-organizing map for this.

3. Definition of ROIs. An expanded ROI forms around the found centers of objects. There goes selection of parts of the original image by a certain «window» in the expanded zone. The selected parts of the input image must contain the center point of the cluster and overlap the neighborhood of ROIs. Dimensions of the «window» correspond to the given dimensions of the largest detected object. They change when the image is zoomed.

4. Analysis in the convolution network. The parts of the input image obtained from ROIs go to the convolutional network for analysis. There goes determination of the presence of an object and its type.

Fig. 2 shows the operation scheme of searching for ROIs.

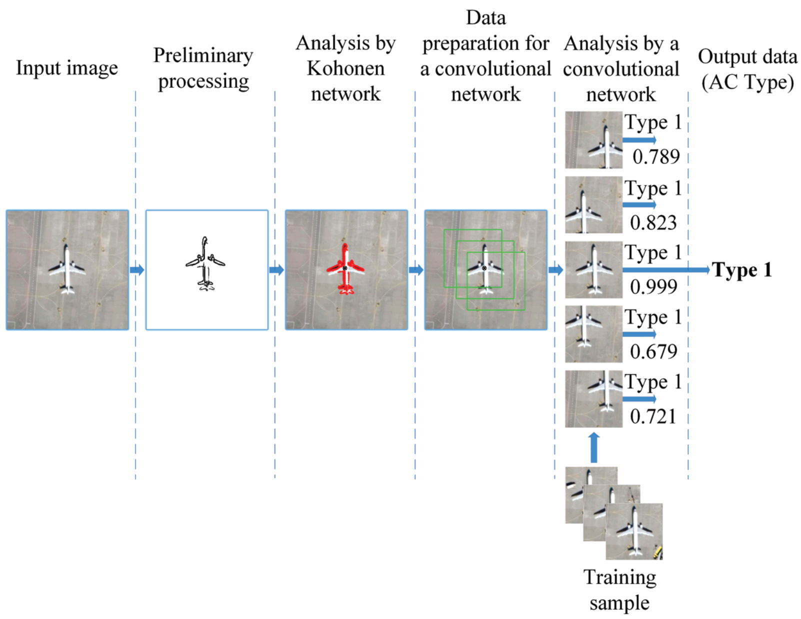

Fig. 2. Operation scheme of the algorithm for searching the ROIs and determination of a type of AC

network. If centers of the clusters are located relatively close to each other, then it is possible to combine them into one, located in the center between them. The same applies to the «windows» around the neighboring cluster centers, if they overlap in more than 90 % of the area. We can also exclude centers located in the immediate vicinity of the edge of an image from the analysis, since they cannot be the center of an aircraft that the convolutional network could recognize in further processing.

As we can see in Fig. 3, Kohonen network and maps determined centers of clusters in such a way that all aircrafts will be highlighted by a «window» and will be used for further analysis in a convolutional network. The obvious disadvantage of a map is the dependence of the position of centers of clusters on neighboring centers, which leads to the presence of centers in areas with no brightness differences. The Kohonen network has no such disadvantage. It locates centers within clusters of points with a difference in brightness almost always.

This algorithm can be used to search and recognize in the input image not only different types of AC, but also other objects of interest, such as robotics, for example, when analyzing the visual data of unmanned vehicles or when testing robotic complexes procedure. To do this, we need to set the maximum and minimum possible sizes of objects for operation of the algorithm for the search of ROIs, to make a training sample of these objects and to train a convolutional NN.

7. Demonstration of operation of the algorithm on actual input data and comparison of the results of application of the self-organizing map and the Kohonen neural network for the search for centers of objects

It is necessary to take into account the real scale of an image and the relative dimensions of the smallest AC to detect several AC in the input image.

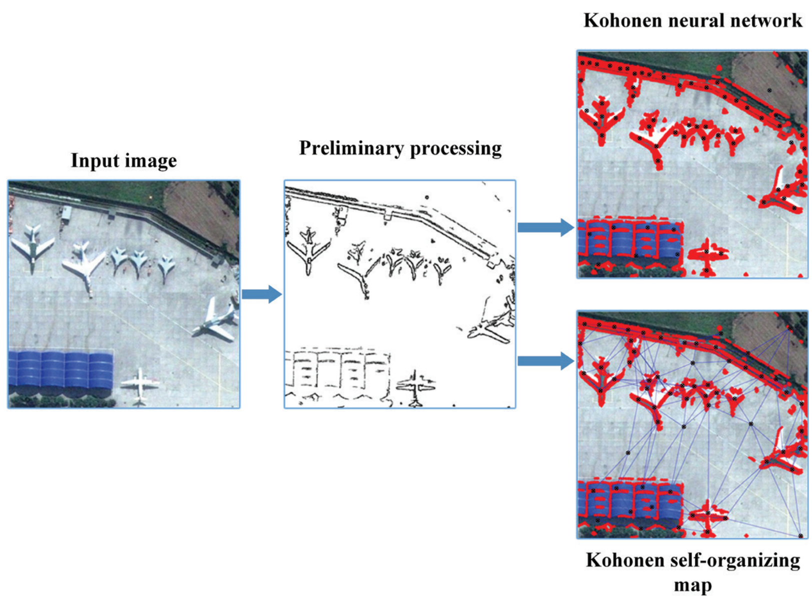

Fig. 3. Initial processing of the input image with further analysis by Kohonen network and self-organizing map

In order not to miss AC, it is necessary that the determined number of cluster centers in the image coincide with the number of the smallest ACs that can fit on the input image with a known scale. Fig. 3 shows an example of operation of the algorithm for determination of ROIs in a real satellite image 7 the size of 767×778 pixels, where several types of AC, various underlying surfaces and structures are present. After preliminary processing, the input image went to the single-layer network and Kohonen self-organizing map with the initial arrangement of neurons at nodes of a two-dimensional grid with hexagonal cells for analysis for comparison. 150 epochs of training are needed for SOM, and 1 is enough for Kohonen network. Based on the input image scale, the number of smallest aircraft that can fit on the image is 8x8, therefore, 64 cluster centers should be identified at the output of the neural

This feature excludes empty areas from further analysis and provides a gain in analysis time and cost of computational resources. In this case, after selection of ROIs around cluster centers by the «window», the Kohonen network sends about 80 cut frames of a size of the maximum AC for further analysis, Kohonen self-organizing map – sends about 120. This is 1,000 times less than when using a sequential division of an input image into frames or 15–100 times less than when using algorithms similar to RCNN, YOLO and SSD. In addition, application of this algorithm reduces the required number of training images of AC for a convolutional network significantly, since a size of the «window» is tied to the image scale and is equal to a size of the largest detected aircraft. This means that there is no need for variations in the size of an aircraft in the training images. This solution gives possibility to reduce the size of a training sample in several times, to speed up network training and to increase recognition accuracy.

Table 2 presents a comparison of performance of Kohonen network and map depending on the resolution of an input image and a number of possible objects of minimal size. We performed calculations in a proprietary program without use of GPU (graphics processor), acceleration and optimization algorithms on a personal computer with the parameters specified in Section 4.

Table 2

Comparison of performance of Kohonen network and map

Size of an input image, pixel	112×218	202×387	767×778
Number of sought for objects, pcs	3	10	64
Processing by Kohonen network, s	0.6	1.1	5.3
Processing by Kohonen maps, s	0.1	0.3	8.5

As we can see from Table 2, the rate of image processing by Kohonen network and Kohonen map has a different dependence on the image size and the number of sought for objects. The self-organizing map shows 3–6 times higher processing rate than the Kohonen network for small and medium-sized images and the number of sought for objects of the order of 1–30 pcs. With further increase in the size of an image and the number of sought for objects, the Kohonen map lags behind the network in processing rate gradually.

8. Discussion of results of the created algorithm for the search for ROIs at the underlying surface

Existing algorithms for detection and recognition of objects allocate several thousand or even hundreds of thousands of parts in an input image and submit them to a convolutional network for analysis, which significantly increases the analysis time. The algorithm proposed in this study makes it possible to reduce a number of analyzed parts of an input image and reduce the size of a training sample for a convolutional network, which can significantly reduce time for search and recognition of necessary objects. We achieve this due to a preliminary search for ROIs in an image with further analysis of the areas found only. Determination of a size of the «window» scanning certain ROIs, based on the scale of an image and the specified dimensions of the largest detected object, eliminates the need for variations in the size of an aircraft in training images. Detection of boundaries of objects and application of Kohonen network and self-organizing map make it possible to determine centers of clusters in an image and create ROIs around them. These solutions give possibility to eliminate variations in scale and center the sought for object, so a training sample size decreases in several times, network training rate and recognition accuracy increase.

The disadvantage of the created algorithm is the fact that a significant increase in recognition rate manifests itself only when analyzing images, where boundaries of objects do not occupy most of space. For example, in satellite and radar images, when the sought for objects are located on a homogeneous underlying surface of great length. The algorithm will determine an entire image as a ROI in the analysis of image occupied by small objects. It will still reduce a number of analyzed parts of an input image. There will be 300–500 times less analyzed parts than when using sequential division or 3–10 times less than when using algorithms similar to RCNN, YOLO and SSD. In the future, we plan to identify unique features of shapes of boundaries of sought for objects, which will help to distinguish these objects against the background of boundaries of the underlying surface and boundaries of objects of no interest.

The preliminary processing algorithm showed a high rate of selection of boundaries of objects in an input image. The search for cluster centers using Kohonen self-organizing map and network makes it possible to process images up to 500×500 pixels in size and a number of search objects is not more than 40 pcs at high rate. In the future, we plan to use algorithms for acceleration and optimization of Kohonen networks, as well as to consider other options for finding of cluster centers for carrying out GPU calculations.

It is possible to apply the presented methodology not only for recognition of a type of aircraft, but also for other objects, which are distinguishable against the background of the underlying surface and are of interest. In the further work, we plan improvement of the presented algorithm due to a better selection of boundaries of objects and search for their centers.

We also plan to create and optimize a convolutional neural network for recognition of objects of interest, which will create a complete system for search and recognition of objects on a radar or satellite image of the underlying surface.

9. Conclusions

1. We applied the algorithm based on the use of the Sobel operator was used to define boundaries of objects, which are present in an image. The algorithm has high rate, it highlights boundaries of objects in 0.002–0.02 sec in an input image.

2. We used Kohonen neural network and self-organizing map to search for object centers, since they are one of the fastest algorithms for determination of centers of clusters of input data.

3. We tested the developed algorithm for determination of ROIs and their centers on real satellite images of the underlying surface. The use of the algorithm makes it possible to reduce a number of parts of an input image analyzed by a convolutional network by 15–100 times, which consequently reduces time required for search and recognition of sought for objects. In addition, application of this algorithm reduces the required number of training images for a convolutional network, since the size of the «window» is tied to the scale of an image and is equal to the size of the largest detected object. This fact, as well as the centering of an object on training images, make it possible to accelerate network training in more than 5 times and to increase recognition accuracy by at least 10 %.

The Kohonen network showed itself more efficient than the self-organizing map in relation to this task, since it places cluster centers 3 times less frequently on the underlying surface. However, the Kohonen map gives a gain in analysis rate for an image of up to 500×500 pixels and a number of sought for objects not more than 40 pcs. As the above parameters increase, SOM begins to yield the processing rate of the Kohonen network. We can apply the given technical solutions to control test operations procedure of robotic complexes and military technology.

References

1. Simard P. Y., Steinkraus D., Platt J. C. Best practices for convolutional neural networks applied to visual document analysis // Proceedings of the Seventh International Conference on Document Analysis and Recognition. 2003. URL: http://cognitivemedium.com/assets/rmnist/Simard.pdf
2. Novikova N. M., Dudenkov V. M. Modelirovanie neyronnoy seti dlya raspoznavaniya izobrazheniy na osnove gibridnoy seti i samoorganizuyuschihsya kart Kohonena // Aspirant. 2015. Issue 2. P. 31–34.
3. Agayan K. Yu., Hanzhin V. G. Neyronnaya set' s arhitekturoy Kohonena dlya raspoznavaniya izobrazheniy // Prochnost' neodnorodnyh struktur – PROST 2018. Sbornik trudov IX-oy Evraziyskoy nauchno-prakticheskoy konferencii. 2018. P. 153.
4. Gerasimova N. I., Verhoturova A. E. Poisk fragmenta izobrazheniya s ispol'zovaniem neyronnoy seti Kohonena // Informacionnye tekhnologii v nauke, upravlenii, social'noy sfere i medicine: sbornik nauchnyh trudov Mezhdunarodnoy konferencii s mezhdunarodnym uchastiem. Ch. 1. Tomsk, 2014. P. 68–70.
5. Soldatova O. P., Chayka P. D. Efficiency analysis of solution of classification using hybrid kohonen neural networks // Izvestiya Samarskogo nauchnogo centra Rossiyskoy akademii nauk. 2015. Vol. 17, Issue 2. P. 1147–1152.
6. Narushev I. R. Neural network on the basis of the self-organizing kochonen card as a means of detecting anomalous behavior // Ohrana, bezopasnost', svyaz'. 2018. Vol. 2, Issue 3 (3). P. 194–197.
7. Kajan S., Sekaj I., Lajtman M. Cluster Analysis Aplications in Matlab Using Kohonen Network. URL: https://pdfs.semanticscholar.org/a8ba/6977dce4bdbeec3dd370eb614de2c6f56514.pdf
8. LeCun Y., Bengio Y. Convolutional networks for images, speech, and time series // The handbook of brain theory and neural networks. 1998. P. 255–258.
9. LeCun Y., Kavukcuoglu K., Farabet C. Convolutional networks and applications in vision // Proceedings of 2010 IEEE International Symposium on Circuits and Systems. 2010. doi: https://doi.org/10.1109/iscas.2010.5537907
10. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation / Girshick R., Donahue J., Darrell T., Malik J. // 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014. doi: https://doi.org/10.1109/cvpr.2014.81
11. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation / Girshick R., Donahue J., Darrell T., Malik J. // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016. Vol. 38, Issue 1. P. 142–158. doi: https://doi.org/10.1109/tpami.2015.2437384
12. Girshick R. Fast R-CNN // 2015 IEEE International Conference on Computer Vision (ICCV). 2015. doi: https://doi.org/10.1109/iccv.2015.169
13. Faster R-CNN: Towards real-time object detection with region proposal networks / Ren S. et. al. // Advances in neural information processing systems. 2015. P. 91–99.
14. Mask R-CNN / He K., Gkioxari G., Dollar P., Girshick R. // 2017 IEEE International Conference on Computer Vision (ICCV). 2017. doi: https://doi.org/10.1109/iccv.2017.322
15. You Only Look Once: Unified, Real-Time Object Detection / Redmon J., Divvala S., Girshick R., Farhadi A. // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. doi: https://doi.org/10.1109/cvpr.2016.91
16. SSD: Single Shot MultiBox Detector / Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A. C. // Computer Vision – ECCV 2016. 2016. P. 21–37. doi: https://doi.org/10.1007/978-3-319-46448-0_2
17. Haykin S. Neyronnye seti: polnyy kurs. 2-e izd. Moscow: Izdatel'skiy dom Vil'yams, 2008.
18. Kohonen T. Self-organizing maps. Vol. 30. Springer Science & Business Media, 2001. 502 p. doi: https://doi.org/10.1007/978-3-642-56927-2
19. Gersho A., Gray R. M. Vector quantization and signal compression. Vol. 159. Springer Science & Business Media, 1992. 732 p. doi: https://doi.org/10.1007/978-1-4615-3626-0