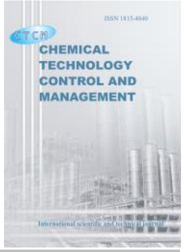
ISSN 1815-4840
Himičeskaâ tehnologiâ. Kontrol' i upravlenie
CHEMICAL TECHNOLOGY. CONTROL AND MANAGEMENT
2020, №2 (92) pp.69-81
International scientific and technical journal
journal homepage: https://uzjournals.edu.uz/ijctcm/



Since 2005

UDC: 004.932.4

APPLICATION OF A CONVOLUTIONAL NEURAL NETWORKS FOR IMAGES RECOGNITION

Avazbek Marakhimov[1], Lyudmila Varlamova[2]

[1]Doctor of Technical Sciences, Professor National University of Uzbekistan named after Mirzo Ulugbek, Tashkent sity, Republic of Uzbekistan
E-mail: avaz.marakhimov@yandex.ru;
[2]Ph.D., Associate Professor, Tashkent University of Information Technologies named after Muhammad al Kwarazmi, Tashkent sity, Republic of Uzbekistan
E-mail: vlp@bk.ru, phone (93)5535563.

Abstract: *In the problems of image recognition, various approaches used when the image is noisy and there is a small sample of observations. The paper discusses nonparametric recognition methods and methods based on deep neural networks. This neural network allows you to collapse images, to perform downsampling as many times as necessary. Moreover, the image recognition speed is quite high, and the data dimension is reduced by using convolutional layers. One of the most important elements of the application of convolutional neural networks is training. The article gives the results of work on the application of convolutional neural networks. The work was carried out in several stages. In the first stage was carried out the modeling of the convolutional neural network and was developed its architecture. In the second stage, the neural network was trained. The third phase produced Python software. The software health check and video processing speed were then performed.*
Key words: *nonparametric methods, small sampling, image recognition, convolutional neural networks, training algorithm.*

Аннотация: *Тасвирни таниб олиш муаммоларида, расм шовқинли бўлганда ва кичик кузатишлар намунаси мавжуд бўлганда, турли хил ёндашувлар қўлланилади. Мақолада параметрик бўлмаган аниқлаш усуллари ва конвулсион нейрон тармоқларига асосланган усуллар муҳокама қилинади. Ушбу турдаги нейрон тармоқ сизга тасвирларни сиқиб чиқаришга, зарур бўлганда схемаларни қисқартиришга имкон беради. Бундан ташқари, расмни аниқлаш тезлиги жуда юқори ва конвулсион қатламлардан фойдаланган ҳолда маълумотлар ҳажми камаяди. Конвулсион нейрон тармоқларни қўллашнинг энг муҳим элементларидан бири бу машғулотдир.*
Таянч сўзлар: *тасвир, таниб олиш, кичик танланма, сунъий нейрон тармоқлари, тасвирга ишлов бериш, нопараметрик усуллар.*

Аннотация: *В задачах распознавания изображений используются различные подходы, когда изображение зашумлено и имеется малая выборка наблюдений. В статье рассматриваются непараметрические методы распознавания и методы, основанные на сверточных нейронных сетях. Этот тип нейронных сетей позволяет сворачивать изображения, проводить субдискретизацию необходимое число раз. Причем скорость распознавания изображения достаточно высокая, а размерность данных понижается использованием сверточных слоев. Одним из важнейших элементов применения свёрточных нейронных сетей является обучение.*
Ключевые слова: *изображение, распознавание, малая выборка, искусственные нейронные сети, обработка изображения, непараметрические методы.*

Introduction

In the problems of image recognition in the conditions of small samples of observations and with uneven distribution, the presence of interference or noise, they face problems of eliminating interference and maximum recognition of image objects. The main indicators are the stability of recognition algorithms in the presence of applicative interference (image noise due to object shadowing, the presence of affected areas), when there is a small sample of observations and a priori data are not available, when applying the Laplace principle is difficult or impossible. For example, in [1-3], the problems of applying approaches to overcoming the problem of a small number of samples were considered using the methods of reducing dimensionality and adaptive nonparametric identification algorithms, and discriminant analysis methods [2-3].

The problems should be divided into tasks with severe restrictions for small samples, but with the presence of a sufficient number of reference images and the problem of classifying images with small samples, but with a large dimension and the smallest number of reference images.

The purpose of this work is to compare the use of nonparametric methods with convolutional neural networks in image recognition problems in the conditions of small observation samples.

I. Statement Of The Problem

The using of contours is common in the tasks of combining a pair of images or an image with a vector model (for example, a location map or a detail drawing), the description of the shape of objects or areas along their contours, for example, using mathematical morphology methods [4], to solve the problem of stereoscopic vision. Contour or spatial representations also serve as the basis for constructing structural descriptions of images.

When comparing contour methods such as the Otsu binarization method; the border detectors Roberts, Prewitt, Sobel, Kenny Laplace, Kirsh, Robinson, then in terms of processing speed they have differences. Thus, among these methods of constructing a gradient field of a halftone image, the Kenny operator algorithm performs a segmentation procedure in a record short time of 0.010sec., and segmentation based on the Otsu binarization method is carried out in 0.139 sec. On average, segmentation using the Roberts, Sobel, Laplace, Prewitt, Kenny operators is 0.0158sec. There are differences in the results of the segmentation carried out. Fig.1 shows the results of construction of gradient field of halftone image using specified operators.

Let the mathematical model of the source image be a two-dimensional discrete sequence $\overline{N}, j = \overline{1, M}$ of the form:

$$X_{i,j} = S_{i,j} + \eta_{i,j},\ i = \overline{1,N},\ \ j = \overline{1,M}, \tag{1}$$

where $S_{i,j}$ is a useful two-dimensional component (the original undistorted image); $\eta_{i,j}$ is the additive noise component; N is the number of rows; M is the number of columns of a two-dimensional image array. Image $X_{i,j}$ of height N with width M in pixels.

The task of constructing a segmented image allows you to remove part of the noise by averaging or smoothing the histogram, and obtaining brightness values at the boundaries of image objects. In this case, the method of randomly choosing brightness values using a 3 × 3 window was used.

All contour methods discussed above are spatial and can be described by expressions of the form [4]

$$I_m(x, y) = T[I_{i,j}(x, y)], \tag{2}$$

where $I_{i,j}(x, y)$ is the input image, $I_m(x, y)$ is the processed image, and T is the operator over $I_{i,j}$ defined in some neighborhood of the point (x, y). The T operator can be applied to a single image and is used to calculate the average brightness over a neighborhood of a point (pixel). The neighborhood of elements in the form of 3 × 3, or 4 × 4, was called the core or window, and the spatial filtering process itself. Since the smallest neighborhood is 1 × 1 in size, g depends only on the value of $I_{i,j}$ at the point (x, y), and T in equation (2) becomes a gradation transform function, also called a brightness transform function or a display function having view

$$s = T(r), \tag{3}$$

where r and s are variables that denote respectively the brightness values of the images $I_{i,j}(x, y)$ and $I_m(x, y)$ at each point (x, y). Based on (2) and (3), the images shown in Fig. 1 (a, b) were processed.

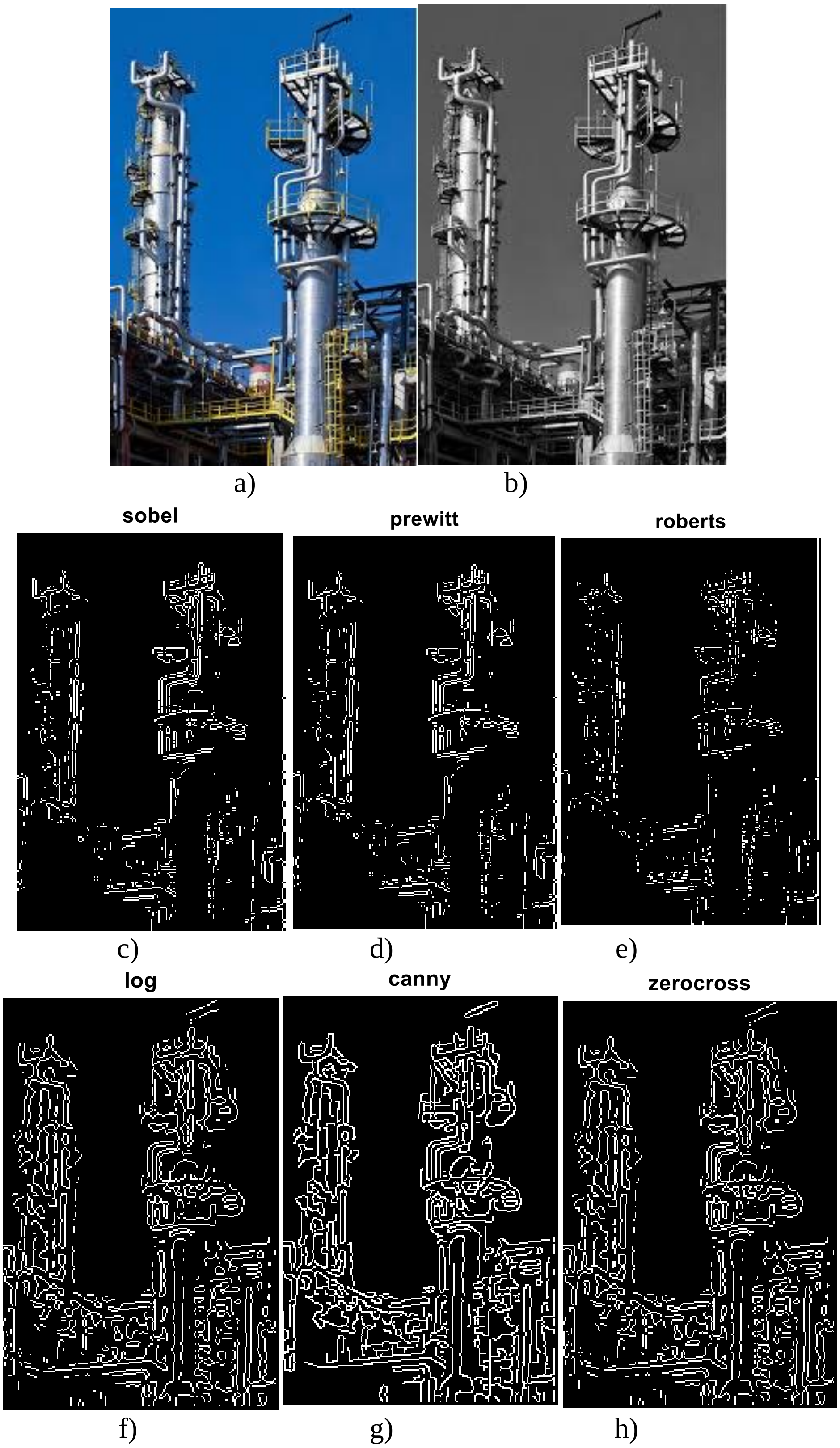


Fig. 1. ***Image processing using operators.***
a) and b) source images: color and gray scale; c) Sobel; d) Prewitt; e) Roberts; f) Laplacian-Gaussian; g) Kenny; h) Robinson.

In the process of spatial image processing, after the stage of brightness transformations, the selection of contours, as a rule, the filtering process follows. This implies the execution of operations on each element or pixel. The spatial filtering scheme can be represented as moving a mask or window across each image element (Fig. 2). It should be noted that spatial filters are more flexible than frequency filters.

Presenting the mask in the form of a matrix of size 3□3, each coefficient of the mask has the following form:

f(x-1, y-1)	f(x-1,y)	f(x-1, y+1)
f(x, y-1)	f(x,y)	f(x, y+1)
f(x+1, y-1)	f(x+1, y)	f(x+1, y+1)

$$\Rightarrow \begin{matrix} w(-1,-1) & w(-1,0) & w(-1,1) \\ w(0-1) & w(0,0) & w(0,1) \\ w(1,-1) & w(1,0) & w(1,1) \end{matrix}$$

*Fig. 2. **Image Area Elements***

The response g (x, y) at each point in the image is the sum of the products

$$g(x,y) = w(s,t) \times f(x,y) \tag{4}$$

$$g(x,y) = w(-1,-1)f(-1,-1) + w(-1,0)f(x-1,y) + w(-1,1)f(x-1,y+1) + w(0,-1)f(x,y-1) + w(0,0)f((x,y) + w(0,1)f(x,y+1) + w(1,-1)f(x+1,y-1) + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1)$$

The image the M×N size, a mask m×n or 3×3, filtration taking into account (4) has an appearance

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t), \tag{5}$$

w(s, t) - filter mask coefficients, $a = \frac{m-1}{2}, b = \frac{n-1}{2}$.

The work carried out image processing and obtained values g (x, y) at each point of the image (Fig.1b) with size *Matrix 274 × 522*. A disadvantage of this technique is the occurrence of undesirable effects, i.e. incomplete image processing when the edges of the image remain untreated due to the nonlinear combination of mask weights. Adding zero elements at the edge of the image results in bands.

Carrying out the equalization of the histogram is similar to averaging the values of the elements along the vicinity of the mask-covered filter, the so-called sliding window method. It consists in determination of the size of a mask of the m×n filter for which the arithmetic average value of each pixel is calculated [5]

$$\bar{g}(x,y) = \frac{1}{m \cdot n} \cdot \sum_{s=-a}^{a} \sum_{t=-b}^{b} g(x,y) \tag{6}$$

If we look at the filtering result in the frequency domain, the set of weights is a two-dimensional impulse response. Such filter will be the FIR-filter with final pulse characteristic (finite impulse response) if area $\bar{g}(x,y)$ of course and pulse characteristic has final length. Otherwise, the impulse response has an infinite length and the IIR-filter is an infinite impulse response filter. However, in this work such filters will not be considered [5].

The correlation is calculated by window filtering, but if to rotate the filter 180 degrees, the image is convolved [4.6].

In the case where the image is very noisy, there is a small sample and the application of the above methods does not produce results for its processing, consider the application of nonparametric methods.

II. Image processing by non-parametric methods

It is necessary to distinguish objects in the presence of noise. Consider the selection using the Parzen window method. Since the distribution of objects in the image is highly uneven using the Parzen method, we will use a variable-width window with a decisive rule [1]

$$a(x, X^l, k, K) = \arg\max_{y \in Y} \lambda_y \sum_{i=1}^{l} [y_i \equiv y] K\left(\frac{\rho(x, x_i)}{h}\right), \tag{7}$$

$$h = \rho(x, x^{(k+1)}),$$

$X_{i,j}, i = \overline{1,N}$, with evaluation of species density

$$p_{y,h}(x) = \frac{1}{l_y V(h)} \sum_{i=1}^{l} [y_i \equiv y] K\left(\frac{\rho(x,x_i)}{h}\right), \quad (8)$$

$K(\theta)$ is an arbitrary even function of the kernel or window of width h, does not increase and positive on the interval [0,1] with weight

$$w(i,x) = K\left(\frac{\rho(x,x_i)}{h}\right), \quad (9)$$

$$K(\theta) = \frac{1}{2}[|\theta| < 1].$$

By small sample size, the matrix of two-dimensional distribution parameters becomes singular, and for small window widths this method reduces to the k-nearest neighbor's method [1], which has its own characteristics such as dependence on the selected step and instability to errors. In this case, it becomes necessary to impose conditions on the distribution density, the function $p_{y,h}(x)$ and the width of the window. Accordingly, the amount of data [2] in the image set is growing. However, a similar problem can be solved by methods of reducing the dimension or by methods of discriminant analysis [2]. Moreover, to reduce the volume of the data set, an external image used database. Then the task of constructing a classifier [1,6-8] is greatly simplified and the problem with a minimum sample and the least number of standards is reduced to a problem with a minimum sample, which is solved by nonparametric methods.

If we have large data sets, an effective mechanism is required to search for neighboring points closest to the query point, since it takes too much time to execute the method in which the distance to each point is calculated. The proposed methods to improve the efficiency of this stage were based on preliminary processing of training data. The whole problem is that the methods of maximum likelihood, k-nearest neighbors or minimum distance, do not scale well enough with an increase in the number of dimensions of space. Convolutional Neural Networks are an alternative approach to solving such problems. For training a convolutional neural network, for example, databases of photographs of individuals available on the Internet can be used [9].

III. Using a convolutional neural network in a minimum sampling image recognition task

Image processing with a deep neural network requires [10,11]:
- define the dimension of the input layer;
- determine the size of the output layer.
- computes the number of convolution layers.
- define the dimensions of the convolution layers.
- number of sub-sampling layers;
-define the dimensions of the subsampling layers.

The construction of the classifier with the help of a deep neural network, a neural network of direct propagation, begins with the first screwing layer. The architecture of the screw neural network is shown in Figure 1.

The CNN shown in Figure 1 consists of different types of layers: convolutional layers, subsampling layers. The convolution operation uses only a limited matrix of small weights (Figure 3), which moves over the entire processed layer (at the very beginning - directly on the input image), forming after each shift an activation signal for the neuron of the next layer with the same position [8.9].

A limited small scale matrix (Figure 4) called a kernel is used to perform the convolution operation. The kernel moves along the entire processed layer (at the very beginning - directly on the input image), after each shift an activation signal is generated for the neuron of the next layer with the same position [10,11].

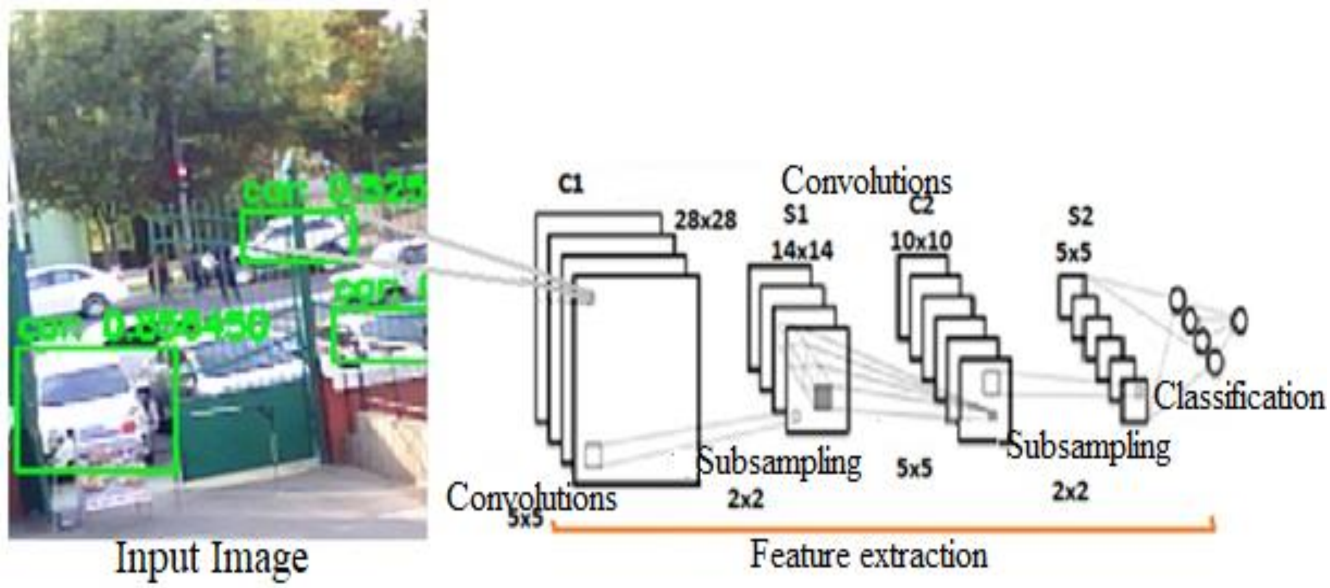


Fig. 3. ***Convolutional neural network architecture.***

The convolutional neural network architecture includes a cascade of convolution layers and sub-sampling layers (stacked convolutional and pooling layers), usually followed by several fully connected layers (FL), allowing local perception to be produced, layer weights to be separated at each step, and data to be filtered. When moving deep into the network, the filters (matrices w) work with a large perception field, which means that they are able to process information from a larger area of the original image, i.e. they are better adapted to the processing of a larger area of pixel space. The output layer of the convolutional network represents a feature map: each element of the output layer is obtained by applying the convolution operation between the input layer and the final sub band (receptive field) with the application of a certain filter (core) and the subsequent action of a non-linear activation function.

Pixel values are stored in a two-dimensional grid, that is, in an array of numbers (Figure 4) that is processed by the kernel and the value written to the next layer [12,13].

Each CNN layer converts the input set of weights into an output activation volume of neurons. Note that the system does not store redundant information, but stores the weight index instead of the weight itself. The direct passage in the convolution layer takes place in exactly the same way as in the full-knit layer - from the input layer to the output layer. At the same time, it is necessary to take into account that the weights of neurons are common [10,14].

Let the image be given in the form of the matrix X and W - the matrix of weights, called the convolution kernel with the central element-anchor.

The first layer is an inlet layer. It receives a three-dimensional array that specifies the parameters of the incoming image

$$F=m \times n \times 3,$$

where F is the dimension of the input data array, $m \times n$ is the size of the image in pixels, “3” is the dimension of the array encoding the color in RGB format. The input image is “collapsed” using the matrix W (Figure 4.) In layer C_1, and a feature map is formed.

The convolution operation is determined by the expression

$$y_{i,j} = \sum_{s=1}^{K} \sum_{t=1}^{K} (w_{s,t}, x_{((i-1)+s,(j-1)+t}), \tag{11}$$

where $w_{s,t}$ is the value of the convolution kernel element at the position *(s,t)*, $y_{i,j}$ is the pixel value of the output image, $x_{((i-1)+s,(j-1)+t}$ is the pixel value of the original image, K is the size of the convolution kernel.

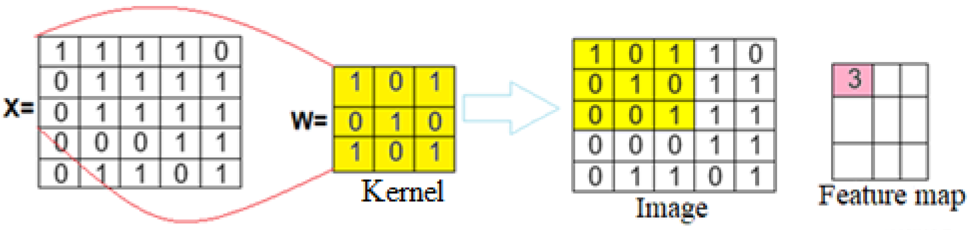


Fig. 4. ***Image convolution algorithm.***

After the first layer, we get a 28 × 28 × 1 matrix — an activation function or a feature map, that is, 784 values. Then, the matrix obtained in layer C_1 passes the operation of subsampling (pooling) using a window of size - $k \times k$. At the stage of subsampling, the signal has the form:

$$y_{i,j} = \max(x_{(ik+s,jk+t)}),$$

where $y_{(i,j)}$ is the pixel value of the output image, $(x_{(ik+s,jk+t)})$ is the pixel value of the initial image and so on to an output layer.

The pooling layer resembles the convolution layer in its structure. In it, as in the convolution layer, each neuron of the map is connected to a rectangular area on the previous one.

Neurons have a nonlinear activation function - a logistical or hyperbolic tangent. Only, unlike the convolution layer, the regions of neighboring neurons do not overlap. In the convolution layer, each neuron of the region has its own connection having a weight.

In the pooling layer, each neuron averages the outputs of the neurons of the region to which it is attached. It turns out that each card has only two adjustable weights: multiplicative (weight averaging neurons) and additive (threshold). The pooling layers perform a downsampling operation for a feature map (often by calculating a maximum within a certain finite area).

Parameters of CNN (the weight of communications the convolutional and full-coherent layers of network) as a rule are adjusted by application of a method of the return distribution of a mistake (backpropagation, BP) realized by means of classical gradient descent (stochastic gradient descent) [14-16]. Alternating layers of convolution and subsampling (pooling) are performed to ensure extraction of signs at sufficiently small number of trained parameters.

IV. Deep learning

The application of the artificial neural network training algorithm involves solving the problem of optimization search in the weight space. Stochastic and batch learning modes are distinguished. In stochastic learning mode, examples from the learning sample are provided to the neural network input one after the other.

After each example, the network weights are updated. In the packet training mode, a whole set of training examples is supplied to the input of the neural network, after which the weights of the network are updated. A network weight error accumulates within the set for subsequent updating.

The classic error measurement criterion is the sum of the mean square errors

$$E_n^p = \frac{1}{2}E_{rr}^2 = \frac{1}{2}\sum_{j=1}^{M}(x_j - d_j)^2 \rightarrow min, \quad (12)$$

where M is number of output layer neurons, j is number of output neuron, x_j is real value of neuron output signal, d_j is the expected value. To reduce the quadratic error, the neural network will be trained by gradient descent, calculating the frequency derivative of the E_n^p with respect to each weight. We get the following ratio:

$$\frac{\partial E_n^p}{\partial w_i} = (x_j - d_j) \times \frac{\partial(x_j - d_j)}{\partial w_i} = (x_j - d_j) \times \frac{\partial}{\partial w_i} g\left(Y - g\left(\sum_{j=0}^{n} w_j x_j\right)\right) = -(x_j - d_j) \times g'(in) \times x_j, \quad (13)$$

$$\frac{\partial E_n^p}{\partial w_i} = x_{n-1}^j \cdot \frac{\partial E_n^p}{\partial y_n^i} \quad (14)$$

$$\frac{\partial E_n^p}{\partial y_n^i} = g'(x_n^j) \cdot \frac{\partial E_n^p}{\partial x_n^i} \quad (15)$$

where g' – derivative activation function

$$\frac{\partial E_n^p}{\partial y_n^i} = x_n^i - d_n^i, \quad (16)$$

x_{n-1}^j is the output of the j-th neuron of the (n-1)th layer, y_n^i is the scalar product of all the outputs of the neurons of the *(n-1)* -th layer neurons and the corresponding weighting coefficients..

The gradient descent algorithm provides error propagation to the next layer and

$$\frac{\partial E_{n-1}^p}{\partial x_{n-1}^i} = \sum_i w_n^{ik} \cdot \frac{\partial E_n^p}{\partial y_n^i},$$

if we need to reduce E_n^p, then the weight is updated as follows

$$w_i \leftarrow w_i + \alpha \times (x_n^j - d_n^j) \times g'(in) \times x_i, \quad (17)$$

where α is training speed.

If the error $E_{rr} = (x_n^j - d_n^j)$ is positive, then the network output is too small and therefore the weights increase with positive input data and decrease with negative input data. With a negative error, the opposite happens. This error obtained in calculating the gradient can be considered as noise, which affects the correction of weights and can be useful in training.

Mathematically, the gradient is a partial derivative of the loss over each assimilable parameter, and one parameter update is formulated as follows [17]:

$$w_i := w_i - \alpha * \frac{\partial L}{\partial w_n}, \quad (18)$$

where – *L is the loss Function.*

The gradient of the loss function with respect to parameters is calculated using a subset of the learning dataset (Figure 5) called the mini-package applied to parameter updates.

This method is called mini-packet gradient descent, also often called stochastic gradient descent (SGD), and the size of the mini-lot is also a hyperparameter [15-17].

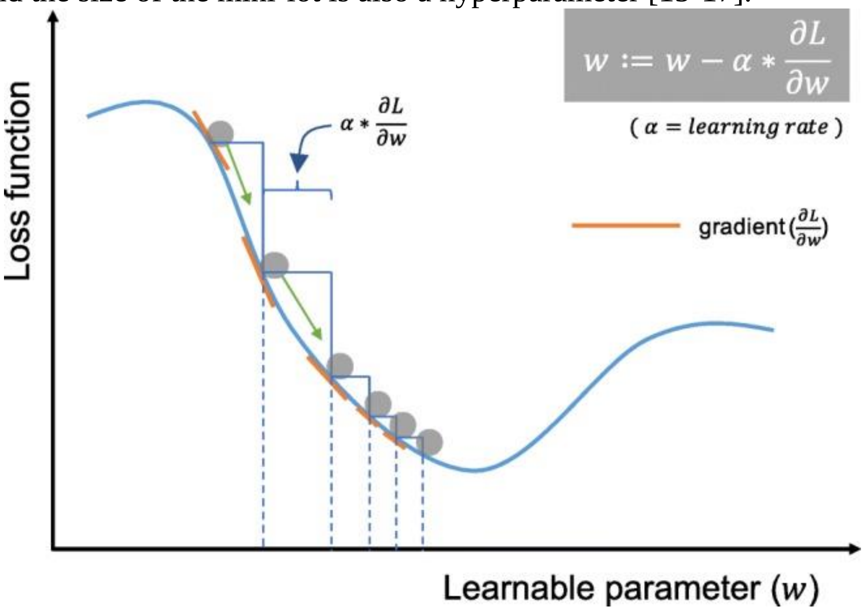


Fig. 5. ***Finding the Loss Function Gradient to a Calculated Parameter(weight) [18].***

Stochastic learning has some advantages over batch learning:
• in most cases, much faster than batch;
• can be used to track changes;
• often leads to better recognizers.

If the training sample size 500 consists of 10 identical sets of 50 examples, the average gradient across a thousand examples would produce the same result as a gradient calculation based on fifty examples. Thus, batch learning calculates the same value 10 times before updating the weights of the neural network. Stochastic learning, by contrast, will present an entire era as 10 iterations (eras) on a learning set of length 50. Typically, examples are rarely found more than once in a learning sample, but clusters of very similar examples may be found [19].

Nonlinear networks often have many local minima of different depths. The task of training is to hit the network in one of the lows. Batch training will lead to a minimum, in the vicinity of which weights are originally located. By stochastic learning, noise appearing when the weights are corrected causes the network to jump from one local minimum to another, possibly deeper [19].

Let’s now consider the advantages of the batch mode of learning over stochastic [19,20]:

• Convergence conditions are well studied;

• A large number of training acceleration techniques work only with batch mode;

• Theoretical analysis of the dynamics of changes in weights and convergence rate is simpler.

These benefits arise from the same noise factor that is present in stochastic learning. Such noises are removed by various methods. Despite some advantages of the batch mode, the stochastic method of training is used much more often, especially in those tasks when the training sample is large [21].

The learning process can be divided into several stages: training, verification and a set of tests (Figure 6).

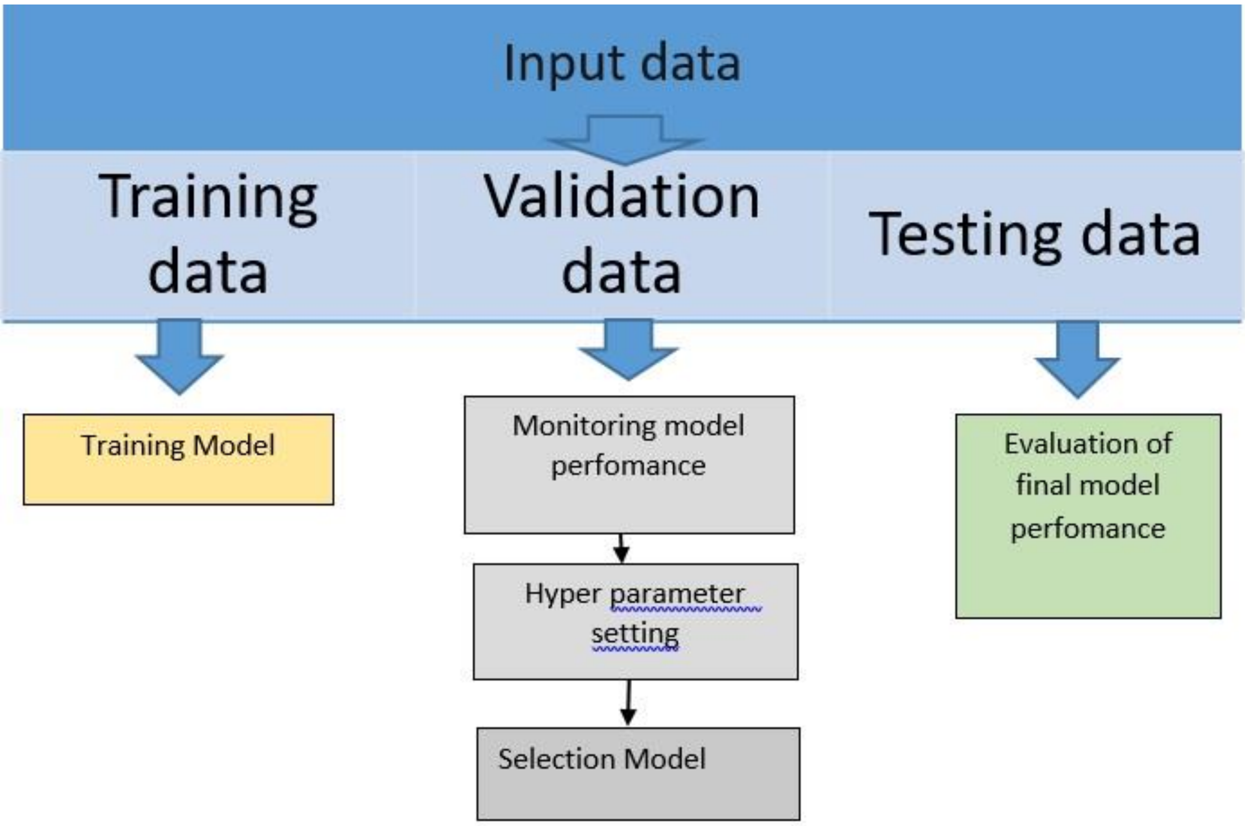


*Fig.6. **The Training Process.***

Learning data involves the use of a learning model with or without a teacher. To verify the correct model selection, performance monitoring is carried out, then the hyperparameter is set up and the model is finally selected. And to check the correct network settings, testing is carried out with an assessment of the final performance [22].

The process of recognition and extraction of signs, the formation of a database of objects is shown in Figure 7.

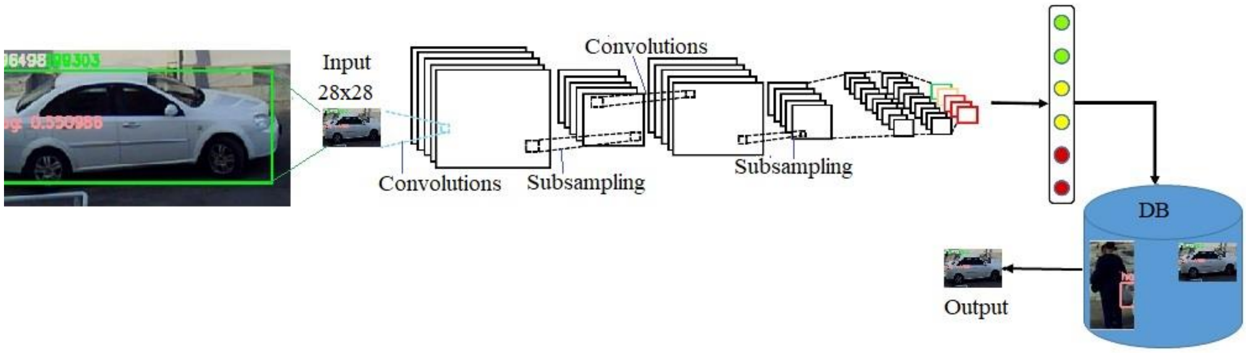


*Fig. 7. **The Recognition Process.***

In the training process a convolutional neural network, when forming a database of objects, steps of calculating a gradient based on the use of a subset of a training set of data are sequentially passed. A network trained on a larger data set generalizes better [23-25]. In the event of noise, the use of filtering methods is shown depending on the type of noise [26].

When creating software for recognizing and classifying objects, we used the teaching method with a teacher, i.e. first, linear regression was calculated (10-14), then by the least squares method, the loss function (18) [22, 27-29].

V. Presence of small observations samples

In the case when the task of training a model for a smaller data set is considered: increasing data and teaching transfer, it is advisable to use methods of deep learning [18,30]. Let's stop on transfer training because it allows to adapt the selected model, for example, on an ImageNet or *lasagne* data sets [9,30-32]. In transfer learning, a model trained on one dataset adapts to another dataset. The main assumption about transfer training is that the general characteristics studied on a sufficiently large data set can be divided between seemingly disparate data sets [33,34]. This portability of studied datasets is a unique benefit of deep learning, which makes itself useful in various tasks with small datasets. The learning algorithm is presented in Figure 8: extraction of fixed functions and fine tuning [35].

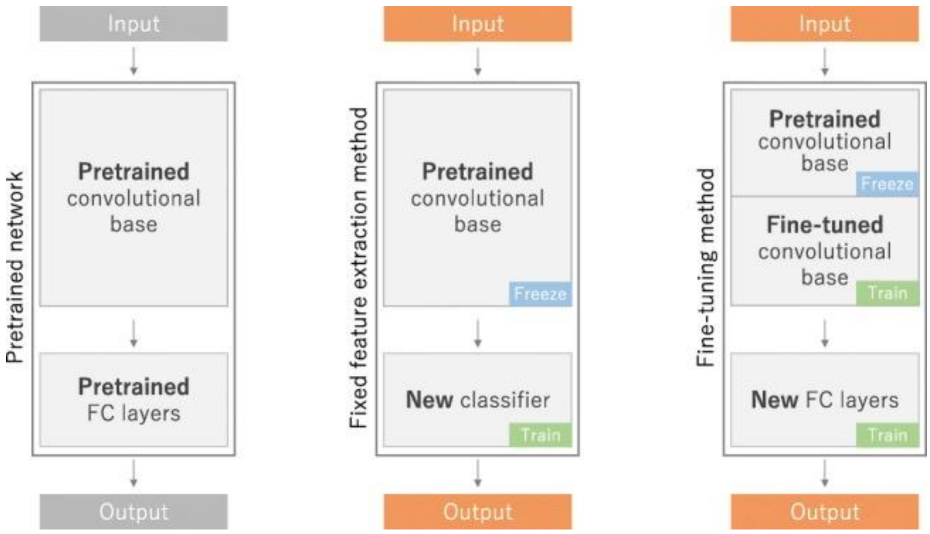


*Fig. 8. **Transfer training.***

A method of extracting fixed features is the process of removing fully connected layers from a network, a pre-trained network, while maintaining the remaining network, which consists of a series of convolutional and combining layers, called a convolutional base, as an extractor of fixed features. In this case, the machine learning classifier adds random weights on top of the extractor of fixed functions in ordinary fully connected convolutional neural networks. As a result, training is limited to the added classifier for a given dataset.

The fine tuning method is not only to replace fully connected layers of a pre-prepared model with a new set of fully connected layers to retrain a given dataset, but also to fine tune all or part of the cores in a pre-trained convolutional basis using reverse propagation (Figures 3,4,7).

All layers of the convolutional layer can be fine-tuned as an alternative, and some earlier layers can be fixed by fine-tuning the remaining deeper layers [18,36,37].

VI. Example of application of a convolutional neural network

In the work, the CNN was chosen with one input layer, two convolutional and two layers of subsampling. Dimension of an entrance layer 1×28×28, the first convolutional layer 32×24×24, the first layer of subsample 32×12×12. The given layers consist of 10 feature cards. The second convolution layer has a dimension of 10 × 10, the subsampling layer is -5 × 5. The network structure is shown in Figure 9.

By training, varying, and testing the selected network, the optimal number of epochs (iterations) was determined. As a result, the loss function L amounted to 3-12, and the recognition rate at various objects ranged from 56 to 97. The database of program objects includes about 80 objects, including people, animals, plants, automobile transport, etc. (Figures 9-10).

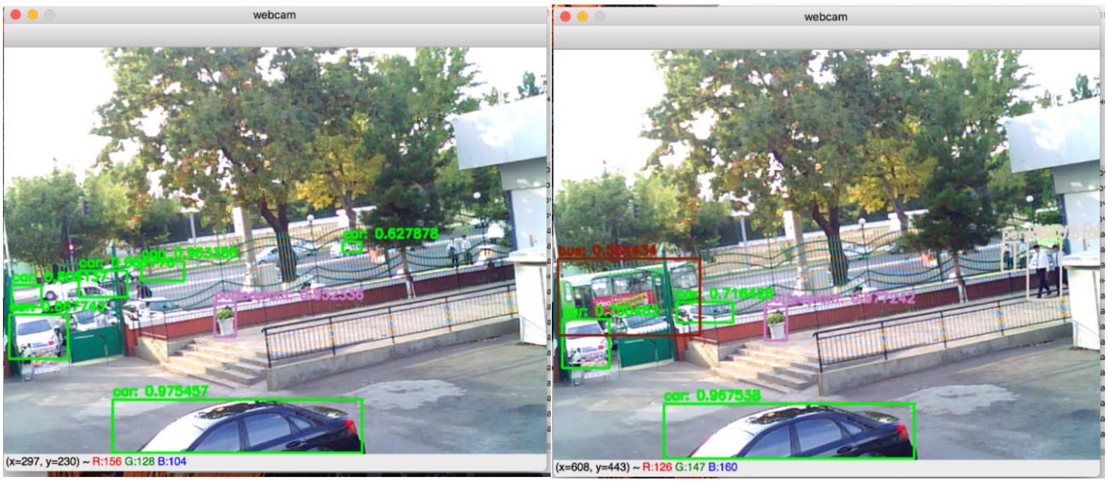


Fig. 9. ***The function of software built on the CNN.***

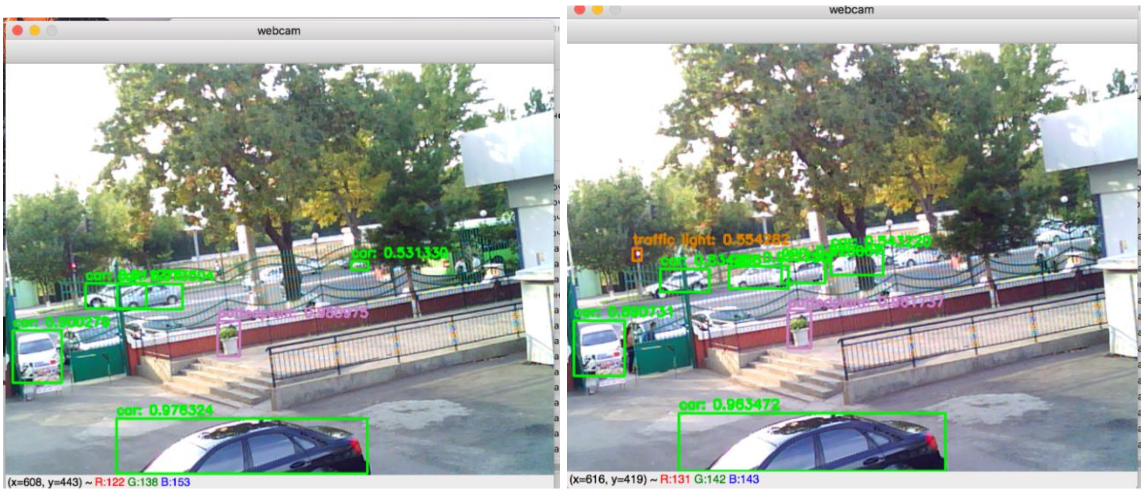


Fig. 10. ***The function of software built on the CNN.***

At the CNN output, the probabilities of matching the object in the database are obtained. A frame is selected with maximum probability and is taken as the final one at the moment. The number of errors was 3-12%. In each frame, the recognized object is highlighted by a rectangular frame, above which the coincidence or recognition coefficient is indicated (Fig. 9.10).

VII. Conclusion

In this article proposed using of nonparametric analysis algorithms in comparison with convolutional neural network algorithms, which showed good results and significantly smaller data size. Regardless of the number of objects that appear in the frame, they were all recognized and the class specified.

However, despite the results obtained, the recognition coefficient for some classes was low. Therefore, need to pay attention to the process of normalizing the input data for training and verification images.

References

1. Duda R., Hart P. Pattern recognition and scene analysis. Translation from English. G. G. Vajeshtejnv, A. M. Vaskovski, edited by V. L. Stefanyuk, "MIR" Publishing House, Moskow 1976. – 509 p.
2. Mokeev V.V., Tomilov S.V. On the solution of the problem of small sample size when using linear discriminant analysis in face recognition problems // Business Informatics. – 2013. №1(23). – pp. 37-43.
3. Lapko A.V., Chencov S.V., Lapko V.A. Non-parametric patterns of pattern recognition in small samples // Autometry. - 1999. -№6. – pp. 105-113.
4. Authorized translation from the English language edition, entitled DIGITAL IMAGE PROCESSING: International Version 3rd Edition; ISBN 0132345633, by GONZALEZ, RAFAEL C; WOODS, RICHARD E., published by Pearson Education, Inc,publishing as Prentice Hall. Copyright ©2008 by Pearson Education, Inc.
5. Voronin, V.V. Methods and algorithms of image recovery in conditions of incomplete a priori information: monograph/V.V.Voronin, V.I.Marchuk - Shakhts: VPO "JURGUES," 2010. – 89 p.
6. Tan, X. Face recognition from a single image per person: a survey / X. Tan, S. Chen, Z.H. Zhou, F. Zhang // Pattern Recognition. – 2006. – Vol. 39, Issue 9. – pp. 1725-1745. – DOI: 10.1016/j.patcog. 2006.03.013.
7. Bertinetto, L. Learning feed-forward one-shot learners / L. Bertinetto, J.F. Henriques, J. Valmadre, P. Torr, A. Vedaldi // Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016. – 2016. – pp. 523-531.
8. Parkhi, O.M. Deep face recognition / O.M. Parkhi, A. Vedaldi, A. Zisserman // Proceedings of the British Machine Vision. – 2015. – pp. 6-17.
9. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, MaS, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Li FF. Imagenet large scale visual recognition challenge. International Journal of Computer Vision 2015; 115(3): 211-252. DOI: 10.1007/s11263-015-0816-y.
10. Lempitsky V. Convolutional neural network. Available at: [https://postnauka.ru/ video/66872].
11. Fukushima K. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern. 1980;36:193–202. doi: 10.1007/BF00344251. [CrossRef] [Google Scholar].
12. Hubel DH, Wiesel TN. Receptive fields and functional architecture of monkey striate cortex. J Physiol. 1968;195:215–243. doi: 10.1113/jphysiol.1968.sp008455. [CrossRef] [Google Scholar].
13. Russell Stewart, Norvig Peter. Artificial Intelligence: A Modern Approach, 2nd ed.: Transl. From English. –M .: Williams Publishing House, 2006. – 1408p.
14. Qian N. On the momentum term in gradient descent learning algorithms. Neural Netw. 1999;12:145–151. doi: 10.1016/S0893-6080(98)00116-6. [PubMed] [CrossRef] [Google Scholar].
15. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv. Available online at: https://arxiv.org/pdf/1412.6980.pdf. Accessed 23 Jan 2018.
16. Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv. Available online at: https://arxiv.org/pdf/1609.04747.pdf. Accessed 23 Jan 2018.
17. Yoshua Bengio, Y. Le Cun, and D. Henderson. Globally trained handwritten word recognizer using spatial representation, space displacement neural networks and hidden Markov models. In Advances in Neural Information Processing Systems 6, San Mateo CA, 1994. Morgan Kaufmann.
18. Convolutional neural networks: an overview and application in radiology. Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, Kaori Togashi. Insights Imaging. 2018 Aug; 9(4): 611–629. Published online 2018 Jun 22. doi: 10.1007/s13244-018-0639-9. Available online at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6108980/.
19. Clark K, Vendt B, Smith K, et al. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. J Digit Imaging. 2013;26:1045–1057. doi: 10.1007/s10278-013-9622-7. [PMC free article] [PubMed] [CrossRef] [Google Scholar].

20. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM (2017) ChestX-ray8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3462–3471. 10.1109/CVPR.2017.369.
21. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436–444. doi: 10.1038/nature14539. [PubMed] [CrossRef] [Google Scholar].
22. Marakhimov, Avazjon and Khudaybergenov, Kabul (2019) "Convergence analysis of feedforward neural networks with backpropagation," Bulletin of National University of Uzbekistan: Mathematics and Natural Sciences: Vol. 2 : Iss. 2, Article 1. Available at: https://uzjournals.edu.uz/mns_nuu/vol2/iss2/1.
23. Tomasi C., Manduchi R. Bilateral filtering for gray and color images // Proc. IEEE Int. Conf. Computer Vision, 1998, – pp. 839-846.
24. Overton K., Weymouth T. A noise reducing preprocessing algorithm // Proc. IEEE Computer Science Conf. Pattern Recognition and Image Processing, Chicago, IL, 1979, –pp. 498-507.
25. C. Chui and Chen Guanrong. Kalman Filtering with Real-Time Applications. Fifth edition (e-Book). Springer Series in Information Sciences. 2017, – 245 P.
26. Marakhimov A.R., Varlamova L.P. Block Form of Kalman Filter in Processing Images with Low Resolution// Chemical Technology. Control and Management/ -2019, №3. Pp.57-72.
27. Marakhimov, Avazjon and Khudaybergenov, Kabul (2018) "Neuro-fuzzy identification of nonlinear dependencies,"Bulletin of National University of Uzbekistan: Mathematics and Natural Sciences: Vol. 1 : Iss. 3, Article 1. Available at: https://uzjournals.edu. uz/mns_nuu/vol1/iss3/1.
28. Varlamova L.P., Salakhova K.N., Tillakhodzhaeva R.S., Neural network approach in the task of data processing.// Young scientist No. 202, 2018. Part 1, –pp. 99-101.
29. Marakhimov A.R., Khudaybergenov K.K., “A fuzzy MLP approach for identification of nonlinear systems”, Contemporary problems in mathematics and physics, CMFD, 65, no. 1, Peoples' Friendship University of Russia, M., 2019, –pp 44–53.
30. Jason Brownlee. A Gentle Introduction to Transfer Learning for Deep Learning.//Available at: [https://machinelearningmastery.com/transfer-learning-for-deep-learning/].
31. Lee DH (2013) Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: Proceedings of the ICML 2013 Workshop: Challenges in Representation Learning. Available online at: http://deeplearning.net/wp-content/uploads/ 2013/ 03/pseudo_label_final.pdf. Accessed 23 Jan 2018.
32. https://pythonhosted.org/nolearn/lasagne.html.
33. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 10.1109/CVPR.2016.90.
34. Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 10.1109/CVPR.2015.7298594.
35. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 10.1109/CVPR.2017.243.
36. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Proceedings of Computer Vision – ECCV 2014, vol 8689. – pp 818–833.
37. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? arXiv. Available online at: https://arxiv.org/pdf/1411.1792.pdf. Accessed 25 Jan 2018.