APPLICATION OF MATHEMATICAL MODELS OF NEURAL NETWORKS IN FACE RECOGNITION

Nurjanov Furkatbek Reyimberganovich
Academy of the Armed Forces of the Republic of Uzbekistan
associate professor of the department of "Information technology and Cybersecurity" of the Armed Forces
https://doi.org/10.5281/zenodo.10473031

Abstract: In this article, solving the problem of neural networks in facial image recognition is becoming increasingly relevant issues in the field of science. In this regard, methods and advantages of using mathematical models of neural networks based on facial image recognition are considered. In addition, the proposed mathematical model and neural network algorithm are very suitable for facial image recognition applications.

Keywords. neural networks, face image recognition, methods, mathematical model, algorithm.

Introduction. In the world, special attention is paid to the study of complex objects in the process of developing biometric technologies, as well as to the improvement and development of identification and recognition systems based on facial images based on image processing using modern computer technologies. In this regard, the requirements for personal recognition and identification systems based on biometric technologies are significantly increasing, in particular, they are widely used in control at airports and subways, access to a building or program, video surveillance, forensics and in a number of other areas.

Human recognition from a facial image stands out among biometric systems in that, firstly, it does not require special expensive equipment. For most applications, a personal computer and a regular video camera are sufficient.

Secondly, there is no physical contact between a person and the devices. There is no need to touch anything or specially stop and wait for the system to respond.

Material and Methods. The disadvantages of recognizing a person from a facial image include the fact that such a system itself does not provide 100% reliability of identification. Where high reliability is required, a combination of several biometric methods is used [11].

1. The task of detecting a face in an image is the first step, preprocessing in the process of solving the problem of identifying a person from a facial image (for example, facial recognition, facial expression recognition).

2. The task of detecting a face in an image is more than simple for human vision, but when trying to build an automatic face detection system one has to face the following difficulties:

• highly variable facial appearance among different people;

• even a relatively small change in the orientation of the face relative to the camera entails a serious change in the image of the face;

• the presence of individual features (mustache, beard, glasses, wrinkles, etc.) significantly complicates automatic recognition;

• changes in facial expression can greatly affect how a face appears in an image;

• part of the face may be invisible (covered by other objects) in the image;

• shooting conditions (lighting, camera color balance, image distortions introduced by the system's optics, image quality) significantly affect the resulting facial image [2].

Currently, the following approaches are used to solve the problem of face detection [1]:

• principal components method;
• factor analysis;
• moment analysis;
• linear discriminant analysis;
• support vector machine;
• hidden Markov models;
• active appearance models (active appearance models);
• wavelet analysis;
• approaches based on artificial neural networks [3–8].

In this regard, recent research has been intensively carried out on the creation of artificial intelligence systems. The main goal was to develop mathematical and software tools for modeling human thinking processes for automatically solving various applied and theoretical problems.

Therefore, the task arises of increasing recognition accuracy, and hence the use of new approaches and image processing algorithms. To qualitatively improve the recognition system, consider the use of a neural network.

A neural network - is a mathematical model in the form of software and hardware implementation, built on the principles of functioning of biological neural networks. Today, such networks are actively used for practical purposes due to the possibility of not only development, but also training. They are used for prediction, pattern recognition, machine translation, audio recognition, etc.

Neural network methods - this offers a different approach to solving the problem of pattern recognition. Thus, the use of neural networks for the task of recognizing a person from a face image is a promising direction, which is the main focus of the preprint.

Results. This paper describes human recognition from a facial image and the possibility of using them in conjunction with neural network methods and mathematical models.

Despite the wide variety of neural network options, they all have common features. So, all of them, just like the human brain, consist of a large number of interconnected elements of the same type - neurons that imitate the neurons of the brain. In Fig 1. shows a diagram of a neuron.

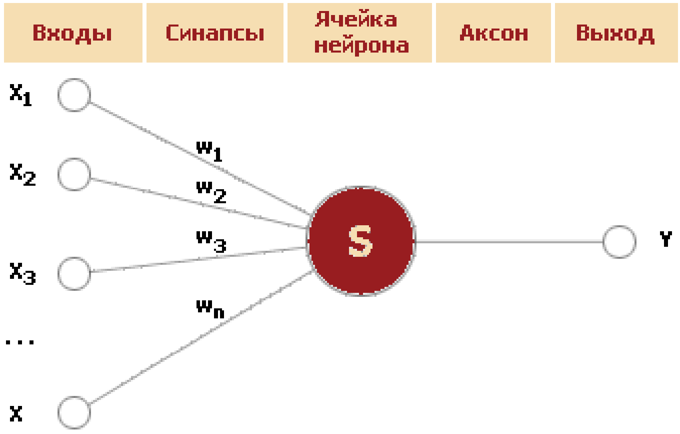


Fig. 1. Neuron diagram

Mathematical description. The figure shows that an artificial neuron, just like a living one, consists of synapses connecting the neuron's inputs to the nucleus; the nucleus of the neuron, which processes input signals and the axon, which connects the neuron with the neurons of the next layer. Each synapse has a weight that determines how much the corresponding neuron input affects its state. The state of the neuron is determined by the formula.

$$S = \sum_{i=1}^{n} x_i * w_i \qquad (1)$$

$$S = \sum_{k=1}^{N} k^{\,2}, \qquad (1.1)$$

where

n– number of neuron inputs.

x_i – meaning i- th neuron input.

w_i – weight i- th synapse.

Then the value of the neuron axon is determined by the formula

$$Y = f(S), \qquad (2)$$

Where f – some function called activation. The most commonly used activation function is the so-called sigmoid.

In the general case, the task of training a neural network comes down to finding a certain functional relationship Y=F(X) Where X – input, and Y – weekend vectors. In the general case, such a problem, with a limited set of input data, has an infinite number of solutions. To limit the search space during training, the task is to minimize the objective function of the NN error, which is found using the least squares method:

The neural network is trained using the gradient descent method, at each iteration the weight is changed according to the formula:

$$E(w) = \frac{1}{2}\sum_{j=1}^{p}(y_i - d_i)^2 \qquad (3)$$

where

y_i– meaning j- th output of the neural network,

d_i– target value j- th exit,

p – number of neurons in the output layer.

The neural network is trained using the gradient descent method, at each iteration the weight is changed according to the formula:

$$\Delta w_{ij} = -\mu \cdot \frac{dE}{dw_{ij}} \qquad (4)$$

Where, h – parameter that determines the learning rate.

$$\frac{dE}{dw_{ij}} = \frac{dE}{dy_{ij}} \cdot \frac{d_{yi}}{dS_j} \cdot \frac{dS_i}{d_{wij}}, \qquad (5)$$

where

y_i–value exit j- th neuron,

S_j– a weighted sum of input signals determined by the formula. (1)

In this case, the multiplier

$$\frac{dS_i}{d_{wij}} = x_i, \qquad (6)$$

where

x_i – value i- th input of the neuron.

Next, consider the definition of the first factor of formula (5)

$$\frac{dE}{dy_{ij}} = \sum k \frac{dE}{dy_k} \cdot \frac{d_{yk}}{dS_k} \cdot \frac{dS_k}{dy_i} = \sum k \frac{dE}{dy_k} \cdot \frac{d_{yk}}{dS_k} \cdot w_{jk}^{(n+1)}, \qquad (7)$$

where

k – number of neurons in a layer n+1.

Let's introduce an auxiliary variable.

$$\delta_j^{(n)} = \frac{dE}{d_{yi}} \cdot \frac{dy_i}{dS_j} \qquad (8)$$

Then we can define a recursive formula to determine the n-th layer if we know the next (n+1) layer.

$$\delta_j^{(n)} = \left[\sum k \delta_k^{(n+1)} \cdot \omega_{jk}^{(n+1)}\right] \cdot \frac{dy_i}{dS_j}, \qquad (9)$$

Finding the last layer of the NN is not difficult, since we know the target vector, the values that the NN should produce for a given set of input values [8].

$$\delta_j^{(N)} = \left(y_i^{(N)} - d_i\right) \cdot \frac{dy_i}{dS_j}, \qquad (10)$$

And finally, we write down formula (4) in its expanded form.

$$\Delta\omega_{ij}^{n} = -\mu \cdot \delta_j^{(n)} \cdot x_i^{n}, \qquad (11)$$

Let us now consider the complete neural network training algorithm:

1. Submit one of the required images to the input of the neural network and determine the values of the outputs of the neurons of the neural network.
2. Calculate the NS for the output layer using formula (10) and calculate changes in the weights of the output layer N using formula (11)
3. Calculate using formulas (9) and (11) respectively and $\Delta\omega_{ij}^{(N)}$ for the remaining layers of the neural network, $n = N - 1 \dots 1$
4. Adjust all NN weights

$$\Delta\omega_{ij}^{(n)} t = \omega_{ij}^{(n)}(t-1) + \Delta\omega_{ij}^{(n)}(t),$$

5. If the error is significant, then go to step 1.

Improving the efficiency of training backpropagation neural networks. The simplest gradient descent method discussed above is very inefficient in the case where the derivatives with respect to different weights are very different.

This corresponds to the situation when the value of the function S for some neurons is close in absolute value to 1 or when the absolute value of some weights is much greater than

1. In this case, to smoothly reduce the error, it is necessary to choose a very low learning rate, but in this case, learning can take an prohibitively long time [3].

The simplest method for improving gradient descent is to introduce a moment m, where the effect of the gradient on the change in weights changes over time. Then formula (11.1) will take the form.

$$\Delta\omega_{ij}^{(n)}t = -\mu \cdot \delta_j^{(n)} \cdot x_i^n + \mu\Delta\omega_{ij}^n(t-1), \ (11.1)$$

An additional benefit of introducing torque is the algorithm's ability to overcome small local minima.

In this regard, at the training stage of the neural network, a pre-trained set of images of a certain size is presented, containing both facial images and non-facial images (background, parts of faces, etc.). Some of these images are shown in Fig. 2.

Fig 2. Images from the training set.

Using this set, the neural network learns to distinguish facial images from non-facial images. The stage of detecting a face image is as follows: the image in which it is necessary to find a face is sequentially scaled by a certain factor, and the resulting set of images is presented to the neural network.

The presented image is scanned by the input layer of the neural network (that is, sections of the image with a certain step are fed to the input of the neural network), and candidates are identified in each of the images.

The main idea underlying neural networks is the sequential transformation of a signal by parallel working elementary functional elements. To solve the problem of detecting a face in an image, a large number of neural networks of various architectures were used. Let us briefly consider the main ones.

Multilayer neural networks, high-order neural networks and radial-basis neural networks are designed to solve these problems. Since such networks operate in the original space of images (features), the requirement of image preprocessing is critical for them. This is bringing the image to a standard form (position, scale, orientation, brightness equalization), reducing the dimension of data, and selecting key characteristics [5]. The next consequence of operating in the original space is the impossibility of taking into account image distortion (for example, when changing perspective, emotions), and therefore the training set must contain a representative set of examples, which are sets of images of objects in the range of angles and lighting conditions in which the recognition system is planned to be used fig. 2.

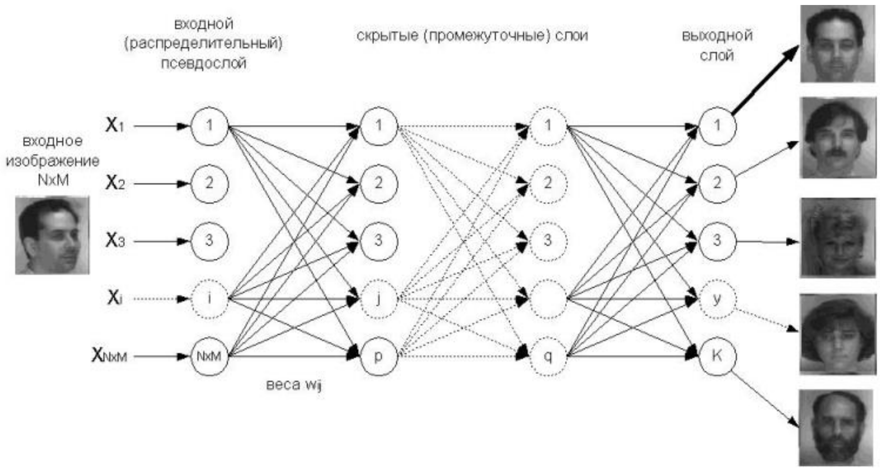


Fig 3. Multilayer neural network architecture and its application for image recognition.

A multilayer neural network is trained using the backpropagation algorithm, which is a type of gradient descent in the space of weights in order to minimize the total network error:

$\Delta W = -\alpha \frac{dE}{dW}, E(w) = \frac{1}{2}\sum_{j=1}^{p}(y_i - d_i)^2$, where t_j– reference value of network outputs. In this case, errors (more precisely, the correction values of the weights) propagate in the opposite direction from inputs to outputs, through the weights connecting neurons. The backpropagation algorithm is NP-hard, so the network training time increases exponentially with increasing data dimension [8-10].

To perform the algorithm, it is necessary to prepare in advance a test sample of l images containing the desired object, and n images not containing it. The total number of test images will be n=l+m. X={x_1, x_2, … , x_n}, where X – the set of all test images, where for each it is known in advance whether the desired object is present or not, and is reflected in the set Y: Y={y_1,,y_2,…,y_n}, where y_j={1, the object is present in the image x_i 0, otherwise under sign j we will understand the structure of the form J={mask, position, size}. The problem comes down to the formal formulation of recognition based on precedents [10].

Accepting a solution to a problem. Direct comparison of features was used in conjunction with the correlation method of comparing images with a sample. The input set of features is analyzed $\{F\} = \{C_{i,j}^{\max}, i = \overline{1,N}, j = \overline{1,M}\}$, where N – number of classes, M – number of class samples in the training set. The following classification options were considered [6-7].

1. Maximum correlation. The detection result is calculated as:

$$\{P_i\} = \{C_i^{\max}\} = \{\max_{j=\overline{1,M}} C_i^{\max}, i = \overline{1,N}\} \quad (12)$$

2. Based on the number of times a fixed threshold is exceeded across the entire sample. Let's introduce some threshold $T \in [0,1]$. Let's determine the detection result using the following formula:

$$\{P_i\} = \left\{\sum_{j=1}^{M}\begin{cases}1, C_{i,j}^{\max} \geq T \\ 0, C_{i,j}^{\max} < T\end{cases}, i = \overline{1,N}\right\}. \quad (13)$$

3. Integral criterion with a fixed threshold. We enter the threshold in the same way $T \in [0,1]$. The detection result is calculated using the formula:

$$\{P_i\} = \left\{ \sum_{j=1}^{M} \begin{cases} C_{i,j}^{\max} - T, C_{i,j}^{\max} \geq T \\ 0, C_{i,j}^{\max} < T \end{cases}, i = \overline{1, N} \right\} \quad (14)$$

4. Integral criterion with a floating threshold. Let's introduce two thresholds $T_{\min \in [0,1]}, T_{\max \in [0,1]}$. Let's calculate the maximum correlation over the entire set $C^{\max}$, If $C^{\max} < T_{\min}$, then we assume that the classification result is guaranteed to be unsuccessful and assume $\{P_i\} = \{0, 1 = \overline{1, N}\}$. Otherwise, we calculate the final threshold $T = C^{\max} \cdot T_{\max}$ and the detection result will be obtained using the formula (14).

5. Complex criterion. Introducing additional thresholds $T_{abs} \in [0,1], T_{mid} \in [0,1]$, so $T_{\min} < T_{mid} < T_{\max} < T_{abs}$. If $C^{\max} < T_{\min}$, then we assume the classification result is guaranteed to be unsuccessful and assume $\{P_i\} = \{0, 1 = \overline{1, N}\}$. Otherwise, we use the following algorithm: If $C^{\max} > T_{abs}$ then we calculate the result using the criterion 1, otherwise if $C^{\max} > T_{\max}$ then we calculate the result according to the criterion 2, having accepted $T = T_{\max}$, otherwise if $C^{\max} > T_{mid}$ then we calculate the result using the criterion 3, having accepted $T = T_{mid}$ otherwise if $C^{\max} > T_{mid}$ then we calculate the result according to the criterion 3, having accepted $T = T_{\min}$.

6. Complex criterion with averaging. We calculate the minimum correlation over the entire set $C^{\min}$. If $C^{\max} > T_{\min}$, then we assume the classification result is guaranteed to be unsuccessful and assume $\{P_i\} = \{0, 1 = \overline{1, N}\}$. Otherwise, we apply the following algorithm and calculate partial results for each of the qualifying conditions: If $C^{\max} > T_{abs}$ then we calculate the partial result using the criterion 1.

If $C^{\max} > T_{\max}$ then we calculate the result using the criterion 2, having accepted $T = T_{\max}$. If $C^{\max} > T_{mid}$ then we calculate the result using the criterion 4, having accepted $T_{\max} = T_{mid}$ in the criteria. If $C^{\max} > T_{\min}$ then we calculate the result according to the criterion 3, having accepted $T = C^{\min}$. For the final $\{P_i\}$ the average of partial results based on the number of fulfilled conditions is taken. To simplify the integration of recognition results of various classifiers $\{P_i\}$ [6-7] scaled to the range [0,1] using the following formula:

$$\{P_i^S\} = \left\{ \frac{P_i - P^{\min}}{P^{\max} - P^{\min}}, i = \overline{1, N} \right\}, \quad (15)$$

where $P^{\max} = \max_{j=\overline{1,N}} P_j, P^{\min} = \min_{j=\overline{1,N}} P_j,$

In this regard, solving a mathematical model for facial recognition based on a neural network, the surveillance system takes a photograph of a person. The neural network searches for the facial area, selects it, and optimizes the scale of the resulting image and its brightness. This image is sent to a second neural network for comparison with the database.

The output is a selection of portraits that are most similar to the incoming image, which are stored in the database, after which the neural network carries out a detailed comparison of the incoming portrait with a selection of images similar to it based on key features. In the case when identification by a feature vector is carried out simultaneously by several classifiers, the resulting output vector [10] is obtained as the component-wise average of the output vectors of all classifiers.

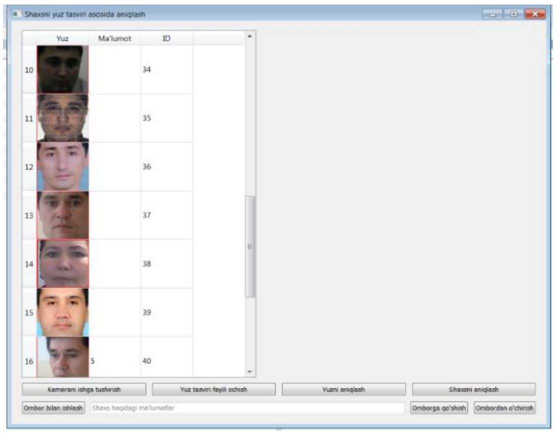
The ORL database contains 400 images: 40 people, 10 images each. For each experiment, the database was randomly divided into two parts, training and testing, with five images of one person in each part. The same training set was used to train the network and compare it with an unknown image.

The recognition capabilities were studied based on the obtained principal components and the possibility of image reconstruction depending on the following factors:

- number of training cycles;
- number of hidden neurons;
- image resolutions: ORL/1 (92x112 pixels, original size), ORL/2 (46x56), ORL/4 (23x28);
- various random breakdowns into test and training parts.

The developed software was tested on an Intel compatible computer running Microsoft Windows Vista Home Premium. To work correctly with video files, K-Lite Mega Codec Pack was installed, with manual settings of all filters in Direct Show mode (Fig. 3 and Fig. 4) are presented.

The software only supports video sources with a resolution of 640x480 pixels. Video cameras used as a source must support Direct Show and the ability to set the RGB 640x480x24 or RGB 640x480x32 mode.



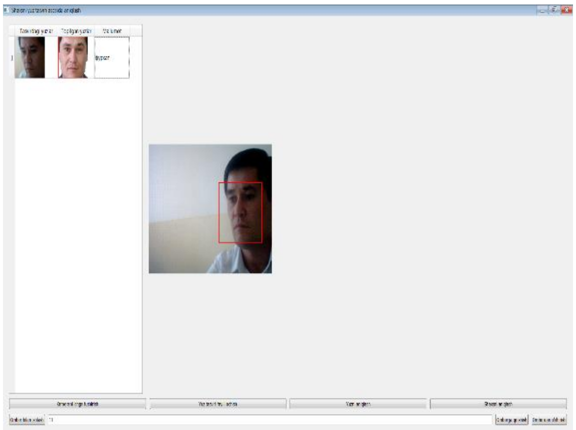


Fig 4. Personality face interface software product

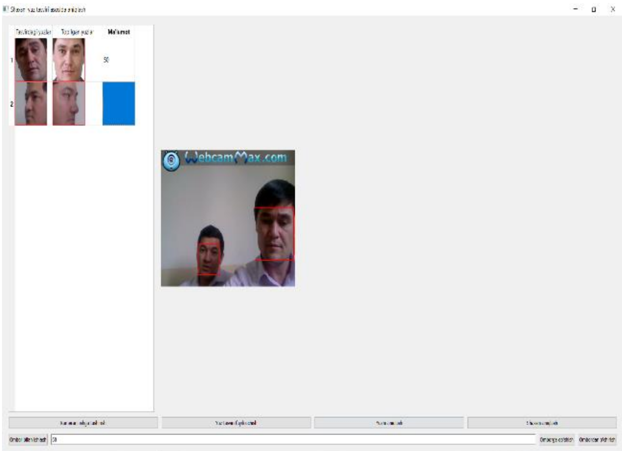


Fig 5. Face image identification

Discussion. A training set of 110 images of faces of 5 people (5x22 images of faces) was formed. Each image was pre-processed with grayscale conversion, histogram equalization, contrasting and scaling to a size of 128 x 128 pixels for the algorithm and 48 x 48, 32 x32 pixels for various areas of interest of the method.

The results obtained can be characterized as follows. The reconstruction error quickly decreases during the first 10-20 steps and then remains virtually unchanged (Fig. 10). Recognition error follows a similar trend. Both of these values fluctuate slightly due to the random order of the training images.

Testing was carried out in 6 zones in the face area: a rectangular eye area, separate zones for each eye, nose, mouth, and the whole face image. Similar tests with similar results were carried out for expansions in significantly larger and smaller numbers of eigenvectors. This allowed us to conclude that this method is of little suitability for constructing features in real conditions of the presence of geometric distortions and instability of lighting for face identification, despite its high performance. For individuals present in the training set, the results are presented in Table 1. Recognition results for users represented in the database

Table 1

Username	**Number of perations**	**Number of access denied**	**Number of recognition errors**	**Successful identification percentage**
Alexander	193	1	0	90.5
Bultakov	187	0	0	88.3
Furkat	175	41	0	92.8
Khurshid	112	9	0	76.6
Sardor	91	9	0	90.3

Successful identification percentage: 92.8%
Recognition errors: 0.0%
Access denied: 7.2%

Here, the number of alarms is the number of times the system issued information about the successful identification of the user; successful identity recognition; number of access denials - the number of frames in which a person was detected, but could not be assigned to any of the classes; number of errors - the number of times the user was mistakenly identified as another person from the database. In the table 2 shows the identification results for unauthorized persons. Recognition results for faces not in the database.

Table 2

Name	**Number activations**	**Number of errors according to statistics**	**Number of access denied**	**Error percentage**
Vladimir	0	0	61	0
Elena	0	0	173	0
Vitalik	0	0	55	0

Mikhail	10	0	205	4.7

Recognition errors: 4.7%

Incorrect access permissions: 0.0%

Here, the number of positives is the number of times an outsider was incorrectly detected as a user from the database; number of errors according to statistics - the number of times when a user was mistakenly admitted to the system as another person from the database (the number of erroneous detections reached the number of accumulations of results required for decision-making). The number of access denials is the number of frames in which a face was detected, but could not be assigned to any of the classes.

Based on the results, it can be seen that, despite the existing false positives, due to statistical processing, access to the system was closed in all cases. Source material – raster image (linear size of the face image – no less than 150x100 pixels, no more than 1000x800 pixels). Scale transformations relative to the standard - reduction/increase by 6 times. Lighting variation is within 10 – 15% of the average brightness level of the reference image. The description is invariant to the background. Multi-standard support (beard/no beard, glasses/no glasses).

The probability of the desired photo appearing in the top five of the rating (with a base of 500 standards) is ~ 97%. Description generation time ~ 1 s. Comparison time ~ 3 s (for a base of 5000 people). Web interface (password protected) – yes.

Conclusion. The paper examines the main classes of human recognition problems from a face image. The advantages and prospects of neural network methods are indicated. The architectures of neural networks that are promising for this task are noted. Particular attention is paid to the choice of the initial image representation and taking into account its properties. Promising directions of research in this area to achieve the intended goal are noted.

The paper presents a mathematical model for post-processing a face image and considers its application to the problem of identifying key points on a person's face. The practical significance of the study lies in the fact that the proposed software algorithms can be used to design face and detail image processing systems, which are a component of modern modeling systems for identifying and comparing face recognition identifiers. The stages of designing the development of a new algorithm for identifying facial images in the process of forming a mathematical model for identifying facial images of an individual have been determined. The significance of the research results is due to the promising development of the theoretical foundations of personality, identifying the biometric technologies of the facial image identification system. A description and results of preliminary experiments on creating an access control system based on the analysis of a human face image are given.

References:

1.Самаль Д.И., Старовойтов В.В. Подходы и методы распознавания людей по фотопортретам. – Минск: Ин-т техн. кибернетики НАН Беларуси, 1998. – 54с.
2.Зайнидинов Х.Н., Нуржанов Ф.Р., Даулетов А.Ю. Параллельная обработка изображения лица личности // Научно-технический журнал «Вестник Туринского политехнического университета в городе Ташкенте». Выпуск 4/2018. -С. 28-33. 2.

3.Алексеева В. И., Коробка Ж. И. Искусственные нейронные сети: рекомендательный библиографический указатель. Краматорск: ДГМА, 2014. С. 25.
4.Gutta S., Wechsler H. Face recognition using hybrid classifiers // Pattern Recognition. – 1997. – Vol. 30. – P. 539-553.
5.Скурихин А.Н. Генетические алгоритмы //Новости искусственного интеллекта. – 1995. – №4. – С. 6-46.
6.Nurjanov F.R. Mathematical models image face of the personality IJARSET: International journal of Advanced Research in Science Engineering and Technology Volume 6 Issue 7, July 2019, India 06/07/2019, 9 pages.
7.Nurjanov F.R. Method and algorithm for identifying the parameters of the image face person // International Journal of Science and Research (IJSR). Volume 8 Issue 6, June 2019, India 08/06/2019. 7 pages. Scientific Journal Impact Factor, № 23; Impact Factor 7,426.
8.Потапов А. С. Искусственный интеллект и универсальное мышление. СПб.: Политехника, 2012. С. 711.
9.Друки А. А. Система поиска, выделения и распознавания лиц на изображениях // Средства и системы обработки и анализа данных. 2011. С.16–25.
10.Sadykhov R., Vatkin M. New training algorithm of neural network “Neokognitron” for a recognition of hand-written symbols // Pattern Recognition and Image Processing. – 2001. – P. 203-212.