Modifications of the Correlation Method of Face Detection in Biometric Identification Systems

Bohdan Zhurakovskyi[1], Nataliia Fedorova[1], Oleksandr Pliushch[2], Mikhail Polishchuck[1], and Nataliia Korshun[3]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," 37 Peremogy ave., Kyiv, 03056, Ukraine*
[2] *Taras Shevchenko National University of Kyiv, 60 Volodymyrska str., Kyiv, 01601,Ukraine*
[3] *Borys Grinchenko Kyiv University, 18/2 Bulvarno-Kudriavska str., Kyiv, 04053, Ukraine*

Abstract

The accuracy of the functioning of modern neural networks both in the reproduction mode and in the learning mode remains insufficient for many practical tasks, therefore it is expedient to create new methods and algorithms for processing signals and data in neural environments. The purpose of the work is to create a pattern recognition method based on modified differential Hebb learning to increase the accuracy of the functioning of modern neural networks both in the playback mode and in the learning mode. Modifications of the correlation method of face detection were developed, which made it possible to reduce the total classification error by more than two times. A scheme of the parallel process of face recognition based on 2D and 3D images by the appropriate algorithm is proposed.

Keywords

Neural network, signal processing algorithm, correlation method of face detection, total classification error.

1 Introduction

When solving a number of problems, such as digital image processing, pattern recognition, meteorological data processing, etc., results must be obtained in real time. Significant progress has been made in the use of neural network technologies in scientific research, business, aviation, customs, etc.

In this work, the goal is to develop a neural network system for recognizing facial features.

The conducted analysis of the current state of neural network technologies allows us to formulate the conclusion that the feasibility of using a specific type of neural network (NM) should be determined based on a comparison of network characteristics with the conditions of the applied problem [1]. The specified characteristics and conditions include [2]: training data parameters; general limitations of the learning process; requirements for computing power; requirements for source information; limitations of the technical implementation of the neural network (NM); scope of application.

An artificial neuron is the basic module of neural networks. It simulates the main functions of a natural neuron [3].

When functioning, a neuron simultaneously receives many input signals. Each input has its own synaptic weight that gives the input the influence it needs for the adder function of the processing element. Weights are a measure of the strength of input connections and model various synaptic strengths of biological neurons.

Another property of neural networks is the huge number of connections that connect individual neurons [4].

According to the architecture of connections, most of the known neural networks can be grouped into two large classes:

1. Networks of direct distribution (with unidirectional serial connections).
2. Backpropagation networks (with recurrent connections).

CPITS-2022: Cybersecurity Providing in Information and Telecommunication Systems, October 13, 2022, Kyiv, Ukraine
EMAIL: zhurakovskiybyu@tk.kpi.ua (B. Zhurakovskyi); natasha_f@ukr.net (N. Fedorova); opliusch@yahoo.com (O. Pliushch); borchiv@ukr.net (M. Polishchuck); n.korshun@kubg.edu.ua (N. Korshun)
ORCID: 0000-0003-3990-5205 (B. Zhurakovskyi); 0000-0002-4548-4198 (N. Fedorova); 0000-0001-5310-0660 (O. Pliushch); 0000-0003-0824-7895 (M. Polishchuck); 0000-0003-2908-970X (N. Korshun)

Direct propagation networks are classified as static, here input signals are received at the inputs of neurons that do not depend on the previous state of the network.

Recurrent networks are considered dynamic, because due to feedback (loops), the inputs of neurons are modified in time, which leads to a change in the states of the network. [3].

2. Research Problem Statement

2.1. Learning Artificial Neural Network

The originality of neural networks, similar to the biological brain, lies in the ability to learn from certain examples that make up the training set. The process of learning neural networks consists in adjusting the architecture and weighting coefficients of synaptic connections in accordance with the data of the training set to effectively solve the given problem [5, 6].

For learning neural networks, it is possible in the case of learning with a teacher (supervised learning) and learning without a teacher (unsupervised learning).

Before use, a neural network with supervised learning must be trained. The learning phase takes some time. Training is considered complete when the neural network reaches the user-defined level of efficiency and planned statistical accuracy [2].

If, after supervised training, the neural network effectively processes the data of the training set, its efficiency when working with data that was not used for training becomes important. If unsatisfactory results are obtained for the test set, training continues [7].

Unsupervised learning claims that computers can learn by themselves. Currently, unsupervised learning is used in networks known as self-organizing maps [8].

Let's build a mathematical model of the described process (Fig. 1).

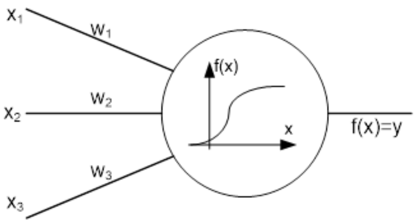

Figure 1: A simplified model of a neuron

The figure shows a model of a neuron with three inputs (dendrites), and the synapses of these dendrites have weights w_1, w_2, w_3. Let the synapses receive pulses of strength x_1, x_2, x_3, respectively, then after passing through the synapses and dendrites, impulses $w_1 x_1$, $w_2 x_2$, $w_3 x_3$ arrive at the neuron. The neuron will transform the received total impulse

$$x = w_1 x_1 + w_2 x_2 + w_3 x_3 \tag{1}$$

according to some transfer function $f(x)$. The strength of the output pulse is equal to

$$y = f(x) = f(w_1 x_1 + w_2 x_2 + w_3 x_3). \tag{2}$$

Thus, the neuron is fully described by its weights w_k and transfer function $f(x)$. Having received a set of numbers (vector) w_k as inputs, the neuron outputs some number y.

Consider the generalized neuron model (Fig. 2), associated with the first attempts to formalize the description of the functioning of the nerve cell.

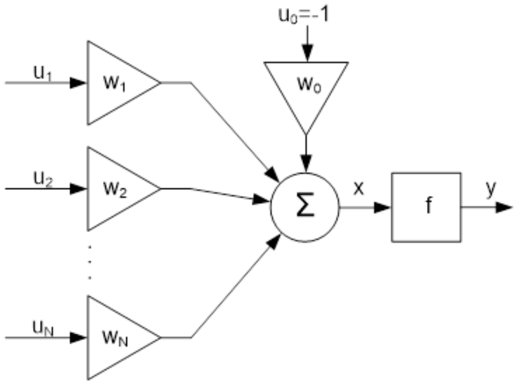

Figure 2: A generalized model of a neuron

Let's introduce the following notations $u_1 \ldots u_N$ are input signals of this neuron coming from other neurons; $w_1 \ldots w_N$ are synaptic weights; y is output signal of the neuron; v is the limiting value [9]. The formula describing the functioning of a neuron has the form

$$f(x) = \begin{cases} 1, \sum_{i=1}^{N} w_i u_i \geq v, \\ 0, \sum_{i=1}^{N} w_i u_i < v. \end{cases} \tag{3}$$

Model (3) can be represented in the form

$$y = f\left(\sum_{i=0}^{N} w_i u_i\right), \tag{4}$$

where

$$f(x) = \begin{cases} 1, x \geq 0, \\ 0, x < 0, \end{cases} \quad (5)$$

and $w_0 = v$, $u_0 = 1$.

Formula (2) describes the neuron model presented in Fig. 1. This model was proposed in 1943 by McCulloch and Pitts. Not only the unit function (2) [9], but also other limiting functions of the form can be taken as a function

$$f(x) = \begin{cases} 1, x \geq 0, \\ -1, x < 0, \end{cases} \quad (6)$$

or

$$f(x) = \begin{cases} 1, x > 1, \\ -1, x < -1, \\ x, |x| \leq 1. \end{cases} \quad (7)$$

At the initial phase of modeling biological neural networks, boundary functions (3), (4), and (5) were used. Currently, the sigmoidal function, which is determined by the expression, is most often used

$$f(x) = \frac{1}{1+e^{-\beta x}} > 0. \quad (8)$$

Note that as $\beta \to \infty$ the characteristic (8) tends to the limiting unipolar function. As an alternative, the hyperbolic tangent function is used

$$f(x) = th\left(\frac{\alpha x}{2}\right) = \frac{1-e^{-\alpha x}}{1+e^{-\alpha x}} > 0. \quad (9)$$

In general, the task of training an artificial neural network is reduced to finding some functional dependence $Y=F(X)$ where X is the input vector and Y is the output vector. In general, such a problem, with a limited set of input data, has an infinite set of solutions [10]. To limit the search space during training, the task of minimizing the objective function of the ANN error, which is found by the method of least squares, is set:

$$E(w) = \frac{1}{2}\sum_{j=1}^{p}(y_j - d_j) \quad (10)$$

where y_j is value j^{th} output of the neuron networks, d_j is the target value of the j^{th} output, p is the number of neurons in the output layer.

$$\Delta w = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \quad (11)$$

where η is parameter that determines the speed of learning.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dS_j} \cdot \frac{\partial S_j}{\partial w_{ij}} \quad (12)$$

where y_j is the value of the j^{th} output of the neuron, S_j is the weighted sum of the input signals, while the multiplier

$$\frac{\partial S_j}{\partial w_{ij}} \equiv x_i \quad (13)$$

where x_i is value of the i^{th} input of the neuron.

Next, consider the definition of the first factor of the formula (12)

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{dS_k} \cdot \frac{\partial S_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{dS_k} \cdot w_{jk}^{(n+1)} \quad (14)$$

where k is the number of neurons in a layer $n + 1$.

1. Now consider the complete learning algorithm of a neural network [11].
2. Apply one of the required samples to the input of the ANN and determine the value of the outputs of the ANN neurons.
3. Calculate $\delta_j^{(N)}$ for the output layer of the ANN according to formula (14) and determine the weight changes $\Delta w_{ij}^{(N)}$ source layer N according to the formula (11).
4. Calculate according to formulas (14) and (11) respectively $\delta_j^{(N)}$ and $\Delta w_{ij}^{(N)}$ for other ANN layers, $N = N - 1..1$.
5. Adjust all the weights of the ANN

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t) \quad (15)$$

6. If the error is significant, then go to Step 1.

At the second stage of network training, vectors from the training sequence are selected alternately in a random order [12].

2.2. Implementation of System Functionality

Objects of different nature can act as images: text symbols, images, sound samples, etc. When training the network, various sample images are offered with an indication of which class they belong to. A sample is usually represented as a vector of its features. At the same time, the totality of all features must uniquely determine the class to which the sample belongs [13]. In case there are not enough features, the network may associate the same sample with several classes, which is incorrect. Upon completion of training, the network can be presented with previously unknown images and receive a response from it about belonging to a certain class [14]. When an image is presented to the network, a sign that the image belongs to this class should appear on one of its outputs. At the same time, other outputs should have a sign that the image does not belong to this class. If two or more outputs have an indication of belonging to a class, the network is considered to be "unsure" of its answer. Classification problems (such as letter recognition) are poorly algorithmized [15]. If in the case of letter recognition the correct answer is obvious to us in advance, then in more complex practical tasks a trained neural network acts as an expert with extensive experience and is able to answer a difficult question.

In fact, Hebb predicted that the synaptic connection between two neurons is strengthened if both neurons are excited [16]. This can be represented as the strengthening of a synapse according to the correlation of the levels of excited neurons connected by a given synapse. For this reason, Hebb's learning algorithm is sometimes called a correlation algorithm. The idea of the algorithm is expressed by the following equation:

$$w_{ij}(t+1) = w_{ij}(t) + NET_i\, NET_j,$$

where $w_{ij}(t)$ is synapse strength from neuron i to neuron j at time t; NET_i is level of excitation of the postsynaptic neuron; NET_j is the level of excitation of the postsynaptic neuron.

Hebb's concept answers the difficult question of how learning can be done without a teacher [17]. In Hebb's method, learning is exclusively local, covering only two neurons and the synapse connecting them; a global feedback system is not required for the development of neural networks.

Further use of Hebb's method for training neural networks led to great success, but at the same time showed the limitations of the method; some images simply cannot be used for teaching with this method. As a result, a large number of extensions and innovations appeared, most of which are largely based on Hebb's work [18].

Hebb's signal learning method. As we have seen, the NET output of a simple artificial neuron is a weighted sum of its inputs. This can be expressed as follows:

$$NET_j = \sum_i OUT_i W_{ij}$$

where NET_j is output NET neuron j; OUT_i is output of the neuron i; w_{ij} is the connection weight of a neuron i with a neuron j.

It can be shown that in this case a linear multilayer network is not more powerful than a single-layer network [19]; the capabilities of the considered network can be improved only by introducing nonlinearity into the transfer function of the neuron. A network using a sigmoidal activation function and a Hebb learning method is said to be trained by a Hebb signal method. In this case, the Hebb equation is modified as follows:

$$OUT = \frac{1}{1 + exp(-NET_i)} = F(NET_i)$$

$$w_{ij}(t+1) = w_{ij}(t) + OUT_i\, OUT_j$$

where $w_{ij}(t)$ is strength of a synapse from a neuron i to the neuron j at a moment in time t; OUT_i is the output level of the presynaptic neuron is equal $F(NET_i)$; OUT_j is the output level of the postsynaptic neuron is equal $F(NET)$.

Hebb's method of differential learning. Hebb's signal learning method involves computing the convolution of previous output changes to determine the weight change. The actual method, called Hebb's differential learning method, uses the following equality:

$$w_{ij}(t+1) = w_{ij}(t) + [OUT_i(t) - OUT_i(t-1)]\,[OUT_j(t) - OUT_j(t-1)],$$

where w_{ij}(t) is strength of a synapse from a neuron i to the neuron j at a moment in time t; $OUT_i(t)$ is the output level of the presynaptic neuron at a time point t; $OUT_j(t)$ is the output level of the postsynaptic neuron at a time point t.

1. Training is carried out as follows:
2. All weights of the network are randomized into small values.
3. An input training vector is applied to the input of the network X and the signal is calculated NET from each neuron using the standard expression

$$NET_j = \sum_i X_i W_{ij}.$$

4. The value of the activation threshold function for the signal is calculated NET from each neuron as follows [20].
5. $OUT_j = 1$, if NET_j more than the threshold θ_j, $OUT_j = 0$ otherwise. θ_j is threshold

corresponding to the neuron *j* (in the simplest case, all neurons have the same threshold). The error for each neuron is calculated by subtracting the obtained output from the required output:

$$error_j = target_j - OUT_j.$$

6. Each weight is modified as follows:

$$W_{ij}(t+1) = w_{ij}(t) + a\, x_i error_j.$$

7. Steps from the second to the fifth are repeated until the error becomes small enough.

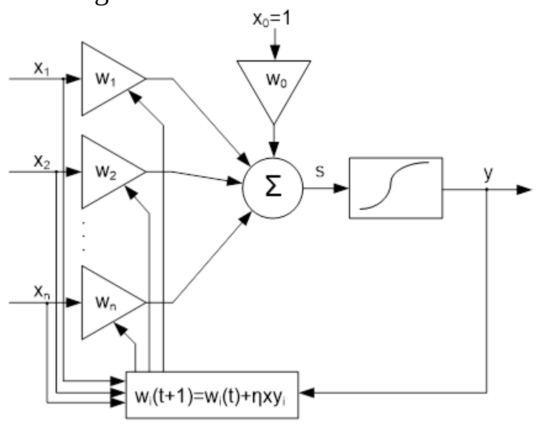

Figure 3: Structure of a Hebb neuron (simple method)

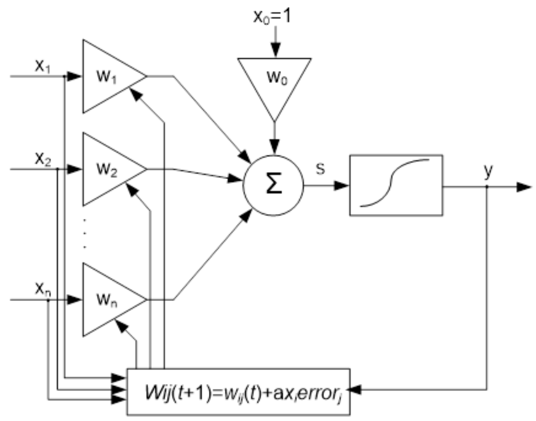

Figure 4: Structure of a Hebb neuron (modified method)

2.3. Testing of the Software Complex

Software implementation is carried out using the Delphi 7.0 environment.

We will conduct a comparative experiment of 10 experiments on the recognition of similar images using the simple and modified Hebb method.

Table 1
Results of experiments

	Digits for recognition	Recognition result	
		Simple Hebb method	Modified Hebb method
1	"3" and "9"	True/False	Faithful/Faithful
2	"3" and "8"	Faithful/Faithful	Faithful/Faithful
3	"3" and "8"	Faithful/Faithful	Faithful/Faithful
4	"3" and "8"	Faithful/Faithful	Faithful/Faithful
5	"3" and "8"	Faithful/Faithful	Faithful/Faithful
6	"2" and "7"	Faithful/Faithful	Faithful/Faithful
7	"0" and "6"	True/False	Faithful/Faithful
8	"0" and "9"	Faithful/Faithful	Faithful/Faithful
9	"0" and "5"	Faithful/Faithful	Faithful/Faithful
10	"8" and "9"	Incorrect/Incorrect	True/False

Thus, using Hebb's simple method, we got 16 correct answers out of 20 (80%).

Using the modified Hebb method, we got 19 correct answers out of 20 (95%).

Studies have shown that the modified Hebb method is more effective in recognizing similar images.

2.4. Development of an Identification Method

The decision on whether the input image belongs to the class of face images is made based on the comparison of the similarity of the image with faces and the threshold for the result of such a comparison. The similarity function is used in the form of the cosine of the angle between normalized vectors such that their means are equal to zero:

$$\begin{cases} S_0(\vec{I}_1, \vec{I}_2) = \dfrac{(\vec{I}_1, \vec{I}_2)}{|\vec{I}_1||\vec{I}_2|}, \\ \dfrac{1}{N}\sum\limits_{i=0}^{N-1} I_1^i = 0, \\ \dfrac{1}{N}\sum\limits_{i=0}^{N-1} I_2^i = 0, \end{cases} \qquad (16)$$

where $S_0(\overrightarrow{I_1}, \overrightarrow{I_2})$ is similarity function, $\overrightarrow{I_1}$ and $\overrightarrow{I_2}$ are compared images presented in vector form, *N* is dimensionality of vectors.

The image similarity function (16) is invariant to uniform changes in image brightness and contrast, which are caused by differences in image acquisition conditions and input equipment. To detect faces of different scales, a pyramid of images is built by scales, and a search is carried out on all levels of the pyramid. Thanks to this, the method allows not only to find the position of faces in the image, but also to determine their scale, which is related to the level of the pyramid that gave the best match [21]. To increase the speed of work and the invariance of the comparison to individual characteristics, the method is applied to reduced images of faces with a horizontal template size of 12 points. To reduce the influence of parts of face images that are subject to the most frequent changes - hairstyle, surrounding background, images of the central area of the face, including eyebrows, eyes, nose and mouth, are used to calculate the template. The quality of the described method is determined by the total error of separation of two classes of images: faces and background. To estimate such an error, the distribution densities of the result of the correlation of the face template with face images from the existing database and background images are used [22].

To represent the properties of faces when solving the identification problem, convolution coefficients of the original image with Gabor functions of various scales and rotation angles are used. Gabor functions are localized in the spatial and frequency domain and have the form of a plane wave with a wave vector $\vec{k}$, on which a Gaussian envelope function of width is superimposed σ/*k*, where σ = 2π:

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2}\exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right)\left[\exp\left(i\vec{k}_j\vec{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right]$$

where

$$\vec{x} = (x, y),\quad \vec{k}_j = (k_{jx}, k_{jy}),\quad k_j^2 = \left|\vec{k}_j\right|^2,\quad x^2 = |\vec{x}|^2,$$

$$\vec{k}_j\vec{x} = k_{jx}x + k_{jy}y.$$

The normalization coefficient, the second exponent in square brackets, is obtained from the condition that the integral of the Gabor functions is equal to zero over the entire domain of definition, which gives the invariance of the convolution of an arbitrary image with the Gabor functions with respect to a constant shift of the image on the brightness scale. Gabor functions of five different scales were used in this work, $\nu = \{0,\ldots,4\}$, and eight turning angles, $\mu = \{0,\ldots,7\}$. Each function was defined by a characteristic wave vector:

$$\vec{k}_j = \begin{pmatrix} k_\nu \cos\phi_\mu \\ k_\nu \sin\phi_\mu \end{pmatrix},\quad k_\nu = 2^{-\frac{\nu+2}{2}}\pi,\quad \phi_\mu = \mu\frac{\pi}{8},\quad (17)$$

where is the index $j = \mu + 8\nu$. The choice of such a set of Gabor functions is due to the best approximation of the original image area by Gabor wavelets. Full wavelet transformation

$$J_j(x, y) = \iint I(x', y')\psi_j(x - x', y - y')dx'dy' \quad (18)$$

gives 40 complex coefficients $J_j(x,y)$ at each point of the image $I(x,y)$ (five scales and eight angles). To determine the proximity of face images, the corresponding face graphs are compared using the comparison function:

$$S_G(G^1, G^2) = \frac{1}{N}\sum_{n=0}^{N-1} S_0(G_n^1, G_n^2),$$

where S_G is graph comparison result; N is the number of nodes in the graph; n is the index that determines the number of the node; G_n^1 and G_n^2 are corresponding feature vectors the n^{th} node of the graph; $S_0(G_n^1, G_n^2)$ is feature vector comparison function (16). The decision about whether the image belongs to the class of images of the faces of a given person is made using a threshold based on the result of comparing the graph of the image being tested and the graph of the face of a known person contained in the database. When analyzing a face image, the configuration of the graph is adjusted to its proportions in order to achieve correspondence between the compared points of the face images.

To combine the stages of face search and recognition into a single complex, an approach is proposed that takes into account the influence of deviations in determining the scales found during the search on the result of identification [23]. To reduce this effect, it is proposed to determine the discretization step by scale when constructing the image pyramid in the search method based on a given limit on the recognition error.

Ways to assess the quality of the recognition method depend on the problem for which it is supposed to be used. As a rule, algorithms and methods of automatic identification of a person based on a facial image are developed to solve the problems of access control and searching in the database. To assess the quality of the method when solving the access control problem, distribution densities are used for the results of comparing images of the faces of different people and one person [24]. At the same time, the recognition threshold is determined based on the importance for specific conditions of errors associated with incorrect identification and refusal of recognition. When evaluating the quality of the method for solving the search problem in the database, the percentage of finding the correct match is used. At the same time, equivalent databases are used, that is, the image of a person's face entered into the test database must also be contained in the database in which the search is carried out. At the same time, these images should not be identical. Sometimes a more complex evaluation of the quality of the method is used, used, for example, during testing within the FERET program. The testing methodology consists in calculating the dependence of the percentage of correct matches in a certain number of found matches with the largest weights on this number. In this work, such a technique is used to compare the developed identification methods with analogues. It should be noted that the objective assessment of automatic identification methods based on face images is extremely difficult, because the test result strongly depends on the database used [25]. In these conditions, it is convenient to use the relative performance indicators of the methods obtained for previously used approaches and developed modifications on the same database of face images.

The first modification consists in dividing the images into separate areas and comparing them individually, which gives the invariance of the comparison result to local changes in the brightness and contrast of the images in the specified areas. In this case, the result of comparing two images is calculated by the formula:

$$S_1 = \sum_{i=1}^{N} S_0^i ,$$

where S_0^i is the result of the correlation calculated by the area with the index $\{i\}$ based on formula (16), N is number of regions. This method of comparison allows more flexible assessment of the similarity of images obtained in conditions with different lighting directions.

The second modification of the correlation method of face area search is based on the use of several templates calculated by the method of own faces. Eigenfaces were calculated as eigenvectors of the covariance matrix obtained on the basis of a training database consisting of 376 face images:

$$S_X = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T ,$$

$$W_{opt} = \arg\max_{W} \left(\left| W^T S_X W \right| \right) ,$$

where S_X is covariance matrix, N is the number of faces in the database, μ is database average vector (average face), x_k k^{th} vector from the database, W_{opt} is matrix of eigenvectors (eigenfaces).

The pattern of decreasing eigenvalues of the matrix showed the possibility of using the first three eigenvectors reflecting the most significant changes in the space of face images. The surface dividing the classes of face and background images is built on the basis of the minimization of the function of the total classification error by the gradient method. It is assumed that the dividing surface is a second-order surface.

The comparative characteristics of the correlation search method that was described and the proposed modifications, obtained on one test set of face and background images (517 face images and 10,000 background images were used), are shown in Figure 5. During testing, faces were searched for 8 scales, the horizontal size of which was changed from 34 to 91 points.

Thus, the proposed modifications of the correlation method of face detection made it possible to reduce the total classification error by more than two times. The speed and quality of work of the developed methods allows their use in various systems that perform automatic analysis of face images as pre-processing methods.

Table 2

Comparative characteristics of the developed methods for detecting the face area in the image

Characteristics	The usual method	Method with division of the template into sections
Total classification error, %	17.7	8.6
Missing face errors, %	10.9	4.9
False detection, %	6.8	3.7

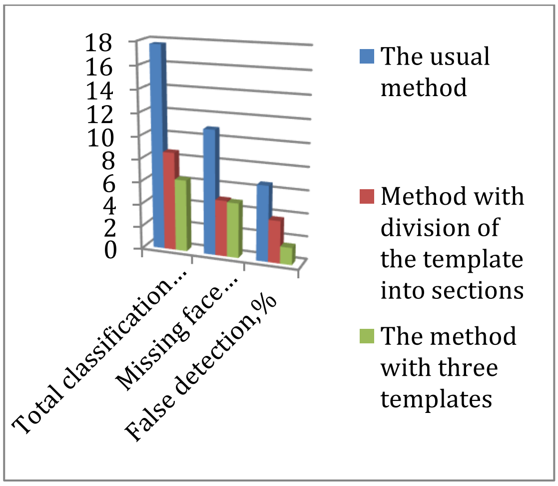

Figure 5: Comparative characteristics of the developed methods for detecting the face region in the image

According to the selected recognition methods for 2D and 3D images, the following scheme of the recognition process has been developed for the face recognition software service using the PrimeSense Carmine 1.08 3D sensor (Fig. 6).

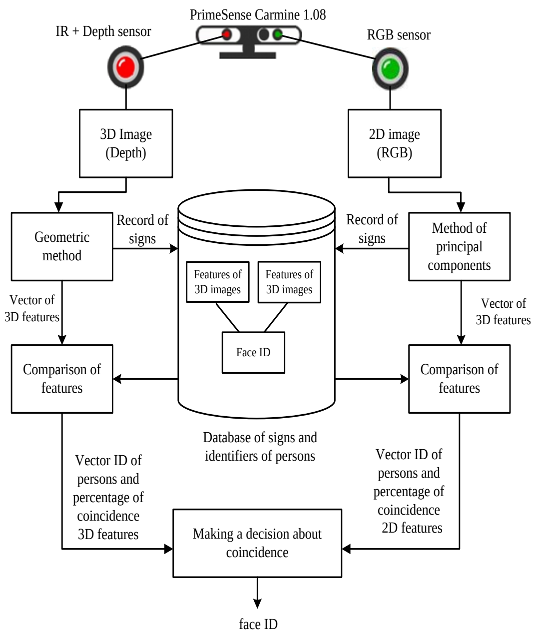

Figure 6: Schematic of the face recognition process using 2D and 3D images

The main idea is a parallel process of face recognition based on 2D and 3D images using an appropriate algorithm. Decision-making in this case takes into account the recognition results of each of the methods [26]. This variant of combined 2D and 3D recognition significantly increases the efficiency and accuracy of recognition, while increasing the reliability of the software service [27].

3. Conclusions

In this work, the Hebb neural network (simple and modified) for recognition of similar images is investigated and implemented. A comparative analysis of the simple and modified Hebb method was carried out. On the basis of the modified Hebb method, modifications of the correlation method of face detection were developed, which made it possible to reduce the total classification error by more than 2 times. A scheme of the parallel process of face recognition based on 2D and 3D images using the PrimeSense Carmine 1.08 3D sensor was built.

4. References

[1] R. M. Golden, Mathematical Methods for Neural Network Analysis and Design, Cambridge, Massachusetts: The MIT Press, 1996.

[2] J. P. Campbell, Speaker Recognition: A Tutorial, vol. 85, no. 9, 1997, pp. 1437–1462.

[3] V. V. Hrytsyk, et al., The Neural and Neural-Like Networks: Synthesis, Realization, Application and Future Information Technologies and Systems, vol. 1, no. 1/2, 1998, pp. 15–55.

[4] M. TajDini, et al., Wireless Sensors for Brain Activity—A Survey, Electronics, vol. 9, iss. 12, 2020. doi: 10.3390/electronics9122092.

[5] S. Chetlur, et al., cuDNN: Efficient Primitives for Deep Learning, 2014. arXiv:1410.0759.

[6] M. Nazarkevych, et al., Research of AtebGabor Filter in Biometric Protection Systems, in International Scientific and Technical Conference on Computer Sciences and Information Technologies, vol. 1, 2018, pp. 310-313.

[7] G. Huang, Z. Liu, K. Q. Weinberger, Densely Connected Convolutional Networks, 2016. arXiv: 1608.06993.

[8] D. Rumelhart, G. Hinton, R. Williams, Learning Internal Representations by Error Propagation, in: Parallel Distributed Processing, Cambridge, vol. 1, 1986, pp. 318—362.

[9] T. Brown, E. Kairiss, C. Keenan, Hebian Synapses: Biophysical Mechanisms and

Algorithms, Annual Review of Neuro-scicnce, vol. 13, 1990, pp. 475–511.
[10] O. Shevchenko, et al., Methods of the Objects Identification and Recognition Research in the Networks with the IoT Concept Support, in Cybersecurity Providing in Information and Telecommunication Systems, vol. 2923, 2021, pp. 277–282.
[11] B. Zhurakovskyi, et al., Coding for Information Systems Security and Viability, in CEUR Workshop Proceedings, vol. 2859, 2021, pp. 71–84.
[12] N. McLaughlin, J. M. D. Rincon, P. Miller, Data Augmentation for Reducing Dataset Bias in Person Re-Identification, in International Conference on Advanced Video and Signal Based Surveillance (AVSS), no. 12, 2015, pp. 1–6. doi: 10.1109/AVSS. 2015.7301739.
[13] M. Moshenchenko, et al., Optimization Algorithms of Smart City Wireless Sensor Network Control, in Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3188, 2021, pp. 32–42.
[14] V. Druzhynin, et al., Features of processing Signals from Stationary Radiation Sources in Multi-Position Radio Monitoring Systems, in CEUR Workshop Proceedings, vol. 2746, 2020, pp. 46–65.
[15] S. Nowlan, G. Hinton, Simplifying Neural Networks by Soft Weight-Sharing, Neural Computation, vol. 4, no. 4, 1992, pp. 473–493.
[16] P. Sermanet, Y. LeCun, Traffic Sign Recognition with Multiscale Convolutional Networks, in International Joint Conference on Neural Networks (IJCNN), 2011, pp. 2809–2813. doi: 10.1109/IJCNN.2011.6033589.
[17] O. E. Rumelhart, O. Zipser, Feature Discovery by Competitive Leaming, Cognitive Science, vol. 9, 1985, pp. 75–112.
[18] F. Rosenblatt, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, vol. 65, no. 6, 1958, pp. 386–408.
[19] B. Zhurakovskyi, et al., Calculation of Quality Indicators of the Future Multiservice Network, Future Intent-Based Networking, vol. 831, 2022, pp. 197–209. doi: 10.1007/978-3-030-92435-5_11.
[20] M. Naser-Moghaddas, P. D. Barjoe, A Heuristic Artificial Neural Network for Analyzing and Synthesizing Rectangular Microstrip Antenna, International Journal of Computer Science and Network Security, vol. 7, no. 12, 2007, pp. 278–281.
[21] S. R. Krishna, J. L. Narayana, L. P. Reddy, ANN Models for Microstrip Line Synthesis and Analysis, World Academy of Science, Engineering and Technolog, vol. 46, 2008, pp. 867–871.
[22] N. Peiravian, X. Zhu, Machine Learning for Android Malware Detection Using Permission and API Calls, in IEEE 25th International Conference on Tools with Artificial Intelligence, 2013, pp. 300–305.
[23] Y. LeCun, et al., Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, vol. 1, no. 4, 1989, pp. 541–551.
[24] J. Bergstra, Y. Bengio, Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research, vol. 13, 2012, pp. 281–305.
[25] N. Srivastava, et al., Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, vol. 15, no. 1, 2014, pp. 1929–1958.
[26] N. Fedorova, et al., Software System for Processing and Visualization of Big Data Arrays, Advances in Computer Science for Engineering and Education, vol. 134, 2022, pp. 324–336. doi: 10.1007/978-3-031-04812-8_28.
[27] A. G. Howard, Some Improvements on Deep Convolutional Neural Network based Image Classification, 2013. arXiv:1312.5402.