

# Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approaches

Oleksii Smirnov<sup>1</sup>, Eugene Fedorov<sup>2</sup>, Anastasiia Neskorođieva<sup>3</sup>, Tetiana Neskorođieva<sup>4</sup>

<sup>1</sup> Central Ukrainian National Technical University, avenue University, 8, Kropivnitskiy, 25006, Ukraine

<sup>2</sup> Cherkasy State Technological University, Cherkasy, Shevchenko blvd., 460, 18006, Ukraine

<sup>3</sup> Vasyl' Stus Donetsk National University, 600-richcha str., 21, Vinnytsia, 21021, Ukraine

<sup>4</sup> Uman National University of Horticulture, 1 Institutka st., Uman, Cherkassy region, 20305, Ukraine

## Abstract

The paper proposes a method for the intellectual classification of gymnastic elements using a combination of descriptive and generative approaches. The created method has the following advantages: the input image is not square, which expands the scope of application; the number of pairs "convolutional layer – downsampling layer" is determined empirically, which increases the classification accuracy of the model; the layer quantity is determined automatically, which speeds up the determination of the model structure; the use of a neural network allows us to label frames of gymnastic elements, and the use of a generative approach allows the resulting sequence of labeled frames of gymnastic elements analyze effectively. The proposed method for the intellectual classification of gymnastic elements can be used in various intelligent visual image recognition systems.

## Keywords

Intelligent classification, gymnastic elements, descriptive approach, generative approach, MLP neural network, 2D neural network LeNet, Adam algorithm, Viterbi algorithm

## 1. Introduction

Assessing the performance of elements in rhythmic gymnastics is a complex task. Every element, from turns and throwing movements to flexibility and balance, is subjected to rigorous analysis. The difficulty lies in the fact that the assessment of such elements is subject to subjective interpretation and requires a high level of professionalism from experts. In the previous work [1] is study of classification problems of gymnastic balance elements performed by rhythmic gymnastics athletes and based on frames. This article discusses classification gymnastics element turn in dynamics based analysis sequences frames. In this context, the development of intelligent methods for classifying gymnastics elements by video can significantly improve the objectivity and efficiency of the evaluation process in rhythmic gymnastics.

## 2. Related Works

The first approach to intelligent image classification was a generative approach, which was based on hidden Markov models [2, 3].

Hidden Markov models have one or more of the following disadvantages:

- insufficiently high classification accuracy;
- insufficiently high speed of parameter identification;

---

COLINS-2024: 8th International Conference on Computational Linguistics and Intelligent Systems, April 12–13, 2024, Lviv, Ukraine

✉ Dr.smirnovoa@gmail.com (O. Smirnov); fedorovee75@ukr.net (E. Fedorov); neskorodieva.a@gmail.com (A. Neskorođieva); tvnesk1@gmail.com (T. Neskorođieva)

🆔 0000-0001-9543-874X (O. Smirnov); 0000-0003-3841-7373 (E. Fedorov); 0000-0002-8591-085X (A. Neskorođieva); 0000-0003-2474-7697 (T. Neskorođieva)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

- complexity of identifying the structure of the hidden Markov model (number of states, size of the mixture for each state).

The second approach to intelligent image classification was the descriptive approach [4, 5, 6], and deep neural networks began to be used to increase recognition accuracy [7, 8].

LeNet-5 neural network [9, 10] has the simplest architecture and uses two pairs of convolutional and downsampling layers, as well as two fully connected layers. The convolutional layer reduces the shift sensitivity of image elements. A downsampling layer reduces the dimensionality of an image. Currently, a combination of LeNet-5 (for feature extraction) and Long Short-Term Memory (LSTM) (for classification) is popular [11, 12].

Neural networks of the Dark Net family [13], neural networks of the AlexNet family [14] and neural networks of the VGG family (Visual Geometry Group) [15, 16] and are a modification of LeNet. These neural networks can have several consecutive convolutional layers.

ResNet family [15, 16, 17] use a Residual block, which contains two consecutive convolutional layers. The output signals of the planes of the layer preceding this block are added to the output signals of the planes of the second convolutional layer of this block. The ResNet combination is currently popular (for feature extraction) and support vector machines (SVM) (for classification) [18].

Neural network DenseNet (Dense Convolutional Network) [16, 19] uses a fully connected (dense) block, which contains a set of Residual blocks. Output signals of the planes of the second convolutional layers of the current Residual block of this dense block are concatenated with the output signals of the planes of the second convolutional layer of all previous Residual blocks of this dense block and with the output signals of the planes of the layer preceding this dense block. In addition, the reduction of the planes of convolutional layers (usually by a factor of two) located between dense blocks is used.

Neural network GoogLeNet (Inception V1) [20] uses an Inception block that contains parallel convolutional layers with connection regions of different sizes and one downsampling layer. The output signals of the planes of these parallel layers are concatenated. To reduce the number of operations, convolutional layers with a unit connection region are sequentially connected to these parallel layers (in the case of convolutional layers, such a convolutional layer is placed before them, and in the case of a downsampling layer, such a convolutional layer is placed after it). The ResNet combination is currently popular (for feature extraction) and support vector machines (SVM) (for classification) [18], used for diagnosis using CXR images, which provided a diagnostic probability close to 100%.

Inception neural network V 3 [16, 17, 21] is a modification of GoogLeNet, and its Inception and Reduction blocks are a modification of the Inception block of the GoogLeNet neural network.

Inception neural network - ResNet - v 2 [16, 17, 22] is a modification of GoogLeNet and ResNet, its Inception block is a modification of the Residual and Inception blocks, the Reduction block is a modification of the Inception block.

Xception neural network [16, 23] uses Depthwise separable convolution block, which performs first a pointwise convolution and then a depthwise convolution. For both convolutions, a ReLU activation function is typically used.

MobileNet neural network [24, 25] uses Depthwise separable convolution block, which performs first depthwise convolution and then pointwise convolution. For both convolutions, a linear activation function is typically used.

MobileNet 2 neural network [16, 26] uses Inverse Residual block, which first performs pointwise convolution, then depthwise convolution, and then pointwise again. For both convolutions, the SiLU activation function is typically used.

MobileNet 3 neural network [27, 28, 29] uses Squeeze and Excitation block in some Inverse Residual blocks.

Deep neural networks have one or more of the following disadvantages:

- insufficiently high classification accuracy;
- insufficiently high speed of parameter identification;

- complexity of identifying the structure of a neural network (number and size of layers of each type).

To increase the speed of identification of parameters of deep neural network models, parallel algorithms are used [27, 30].

In connection with this, the problem of creating an effective intellectual classification of gymnastic elements is urgent.

The goal of the work is to increase the efficiency of intellectual classification of gymnastic elements using a combination of descriptive and generative approaches.

To achieve this goal, it is necessary to solve the following tasks:

1. Create the structure of a method for the intellectual classification of gymnastic elements, which combines descriptive and generative approaches.
2. Develop a one-dimensional neural network model for classifying frames of gymnastic elements.
3. Create a model of a two-dimensional neural network for classifying frames of gymnastic elements.
4. Develop a method for identifying the parameters of a neural network model.
5. Create a method for classifying the sequence of frames of gymnastic elements.
6. Select quality criteria for the method of intellectual classification of gymnastic elements.
7. Conduct a numerical study of the proposed method for intelligent classification of gymnastic elements.

### 3. Methods and Materials

#### 3.1. Structure of the intellectual method classification of gymnastic elements based on a combination of descriptive and generative approaches

In the proposed method, the outputs of the neural network are considered as the probabilities of the appearance of the observation symbol (the  $t$ -th frame of the gymnastic element) in the  $j$ -th state (gymnastic pose) (at the  $j$ -th output of the neural network). The Viterbi dynamic programming method is applied to a labeled sequence of gymnastic element frames. On the other hand, the parameters of a neural network can be identified based on a sequence of frames labeled by the Viterbi method. This combination provides classification probabilities comparable to those of DTW, discrete and semi-continuous Hidden Markov Models (HMMs), and does not require a separate neural network for each gymnastic element, as in these methods.

Main stages of the proposed method:

1. To initially identify the parameters of the neural network, manually labeled frames of gymnastic elements from the database [31] are used. Based on the labeled frames of the database for future use in the Viterbi method, the following are calculated:

- a priori probability  $P(s_j)$  in the form

$$P(s_j) = \frac{m_j}{m},$$

where  $m_j$  is the number of frames marked with state  $s_j$  in the entire set of training data of the standard database,

$m$  is the number of all frames in the entire set of training data of the standard database.

- the probability of the initial state  $s_j$  for the Bakis HMM model or the HMM model with a limited transition is determined by the formula,  $\tilde{\pi}_j = \begin{cases} 1, & j = 1 \\ 0, & j > 1 \end{cases}$ ;
- probability of transitions between states  $a_{ij}$  in the form  $a_{ij} = \frac{n_{ij}}{n_i}$ ,

where  $n_{ij}$  is the number of any transitions from state  $s_i$  to state  $s_j$  across the entire set of training data of the standard database,

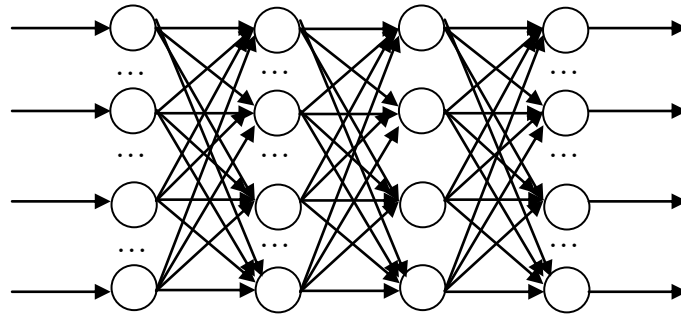
$n_i$  is the number of any transitions from the state  $s_i$  across the entire set of training data of the standard database.

2. Frames of gymnastic elements are recognized using a neural network model, i.e. segmentation is performed.
  3. A modified Viterbi algorithm is used, which optimizes segmentation (sequence of states). For this algorithm, the probability distribution of the occurrence of an observation symbol  $\mathbf{o}_t$  ( $t$ -th frame) in the  $j$ -th state is pre-calculated  $b_j(\mathbf{o}_t)$  according to Bayes' rule as an emission probability  $p(\mathbf{o}_t | s_j) = \frac{p(s_j | \mathbf{o}_t)P(\mathbf{o}_t)}{P(s_j)}$ , where the posterior probability  $p(s_j | \mathbf{o}_t)$  is the output of the  $j$ -th neuron of the neural network, the prior probability  $P(\mathbf{o}_t)$  is fixed and can be omitted,
  4. The parameters of the neural network model are identified using frame markers of gymnastic elements (segmentation result) obtained using a modified Viterbi algorithm.
  5. For a given subject area, frames of gymnastic elements are recognized using a neural network model.
  6. If the recognition error of the neural network exceeds the threshold, then go to step 3.
- Next, we consider models of neural networks that mark frames of gymnastic elements.

### 3.2. One-dimensional neural network for classifying frames of gymnastic elements based on a multilayer perceptron

Figure 1 shows a one-dimensional classification neural network based on a multilayer perceptron (MLP), which is a non-recurrent static multilayer neural network containing two hidden layers and an output layer. The classes are separated by hyperplanes.

For MLP, error-correction-based learning (supervised learning) is used in batch mode, and the Adam algorithm was used in the work.



**Figure 1:** MLP based 1D neural network model

**One-dimensional neural network model is presented as follows:**

$$y_i^{(0)} = x_i,$$

$$y_j^{(k)} = f^{(k)}(s_j^{(k)}), s_j^{(k)} = b_j^{(k)} + \sum_{i=1}^{N^{(k-1)}} w_{ij}^{(k)} y_i^{(k-1)}, j \in \overline{1, N^{(k)}}, k \in \overline{1, L},$$

where  $N^{(k)}$  is the number of neurons in the  $k$ -th layer,

$k$  is the layer number,

$L$  is the number of layers,

$b_j^{(k)}$  is the threshold of the  $j$ -th neuron a in the  $k$ -th layer,

$w_{ij}^{(k)}$  is the connection weight from the  $i$ -th neuron to  $j$ -th neuron on  $k$ -th layer,

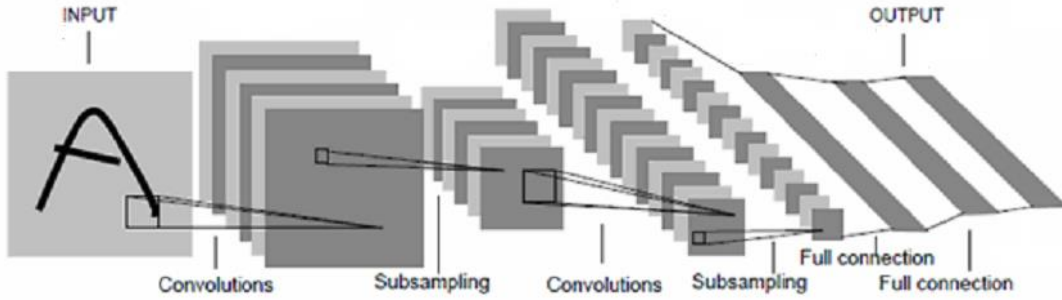
$y_j^{(k)}$  is the output of the  $j$ -th neuron on the  $k$ -th layer,

$f^{(k)}$  is the activation function of neurons of the  $k$ -th layer.

ReLU was used as quality,  $f^{(k)}$  softmax was used as quality  $f^{(L)}$ .

### 3.3 Two-dimensional neural network for classifying frames of gymnastic elements based on 2D LeNet

Figure 2 shows a two-dimensional neural network for classification based on 2D LeNet, which is a non-recurrent dynamic neural network and has a hierarchical structure.



**Figure 2:** Model LeNet based 2D neural network

2D\_LeNet is a special class of multilayer perceptron. It is formed by an input layer, which consists of a single receptor plane, alternating convolutional layers (corresponding to neocognitron  $S$ -layers) and downsampling (pooling) layers (corresponding to neocognitron  $C$ -layers), a sequence of fully connected layers (hidden MLP layers) and an output layer. The convolutional layer consists of convolutional planes. The downsampling layer consists of downsampling planes. Each convolutional plane consists of convolutional cells, each downsampling plane consists of downsampling cells. The convolutional layer reduces the shift sensitivity of image elements. A downsampling layer reduces the dimensionality of an image. The connection area of the cell plane of the previous layer is associated with a cell of the cell plane of the current layer. Geometrically, the communication area is usually a square. For all planes of one layer it has the same size. All cells of the same plane of cells of the current layer associated with the connection areas of the plane of cells of the previous layer have the same weights. The cell plane communication regions of the downsampling layer overlap. Because of this, one cell in the downsampling layer's cell plane entering different overlapping communication regions can activate multiple cells in the convolutional layer's cell plane. Communication area for 2D LeNet does not go beyond the boundaries of the plane, so the size of the convolutional layers gradually decreases.

For this neural network model, training is used based on error correction (supervised learning) in batch mode, and the Adam algorithm was used in the work.

#### 3.2.1. Neural network model

Let  $\nu$  be the position in the connection region,  $\nu = (\nu_x, \nu_y)$ ,  $K_I$  be the number of cell planes in the input layer  $I$  (for RGB images 3),  $K_{S_l}$  be the number of cell planes in the downsampling layer  $S_l$ ,  $K_{C_l}$  be the number of cell planes in the convolutional layer  $C_l$ ,  $A_l$  be the connection region of the layer plane  $S_l$ ,  $\hat{L}$  and be the number of convolutional (or downsampling) layers,  $\tilde{L}$  - the number of fully connected layers.

1.  $l = 1$ .
2. Calculate the output signal for the convolutional layer

$$u_{c_l}(m, i) = f_{c_l}(h_{c_l}(m, i)) \quad m \in \{1, \dots, N_{C_l}\}^2, i \in \overline{1, K_{C_l}},$$

$$h_{c_l}(m, i) = \begin{cases} b_{c_l}(i) + \sum_{k=1}^{K_I} \sum_{\nu \in A_l} w_{c_l}(\nu, k, i) x(m + \nu, k), & l = 1 \\ b_{c_l}(i) + \sum_{k=1}^{K_{S_{l-1}}} \sum_{\nu \in A_{l-1}} w_{c_l}(\nu, k, i) u_{s_{l-1}}(m + \nu, k), & l > 1 \end{cases},$$

where  $w_{c_1}(\nu, k, i)$  is the weight of the connection from the  $\nu$ -th position in the connection area of the  $k$ -th plane of the cells of the input layer  $I$  to the  $i$ -th plane of cells of the convolutional layer  $C_1$ ,

$w_{c_l}(\nu, k, i)$  is the weight of the connection from the  $\nu$ -th position in the connection area of the  $k$ -th plane of cells of the downsampling layer  $S_{l-1}$  to the  $i$ -th plane of cells of the convolutional layer  $C_l$ ,

$u_{c_l}(m, i)$  is the output of the cell in the  $m$ -th position in the  $i$ -th plane of the cells of the convolutional layer  $C_l$ ,

$f_{c_l}$  is the activation function of the neurons of the convolutional layer  $C_l$ .

3. Calculate the output signal for the downsampling layer (halving the scale)

$$u_{s_l}(m, k) = \frac{1}{4} \sum_{\nu \in \{0,1\}^2} u_{c_l}(2m + \nu, k), m \in \{1, \dots, N_{s_l}\}^2, k \in \overline{1, K_{s_l}},$$

where  $w_{s_l}(k, k)$  is the connection weight from the  $k$ -th plane of cells of the convolutional layer  $C_l$  to the  $k$ -th plane of cells of the downsampling layer  $S_l$ ,

$u_{s_l}(m, k)$  is the output of the cell in the  $m$ -th position in the  $k$ -th plane of cells of the downsampling layer  $S_l$ .

4. If  $l \leq \bar{L}$ , then  $l = l + 1$ , go to 2.

5. Output calculation for a fully connected layer:  $u_{d_l}(j) = f_{d_l}(h_{d_l}(j))$ ,  $j \in \overline{1, N_{d_l}}$ ,  $l \in \overline{1, \bar{L}}$ ,

$$h_{d_l}(j) = \begin{cases} b_{d_1}(j) + \sum_{k=1}^{K_{s_{\bar{L}}}} \sum_{\nu \in \{1, \dots, N_{s_{\bar{L}}}\}^2} w_{d_1}(\nu, k, j) u_{s_{\bar{L}}}(\nu, k), & l = 1 \\ b_{d_l}(j) + \sum_{z=1}^{N_{d_{l-1}}} w_{d_l}(z, j) u_{d_{l-1}}(z), & l > 1 \end{cases},$$

where  $w_{d_1}(\nu, i, j)$  is the weight of the connection from the  $\nu$ -th position in the connection area of the  $k$ -th plane of cells of the downsampling layer  $S_{\bar{L}}$  to the  $k$ -th neuron on the first fully connected one layer  $D_1$ ,

$w_{d_l}(z, j)$  is the weight of connection from the  $i$ -th fully connected neuron layer  $D_{l-1}$  to  $j$ -th neuron on the  $l$ -th fully connected layer  $D_l$ ,

$u_{d_l}(j)$  is the output of the  $j$ -th fully connected neuron layer  $D_l$ ,

$f_{d_l}$  is the activation function of fully connected neurons layer  $D_l$ .

6. Output calculation for output layer

$$u_o(j) = f_o(h_o(j)), j \in \overline{1, N_o}, h_o(j) = b_o(j) + \sum_{z=1}^{N_{d_{\bar{L}}}} w_o(z, j) u_{d_{\bar{L}}}(z),$$

where  $w_o(z, j)$  is the weight of connection from the  $i$ -th fully connected neuron layer  $D_{\bar{L}}$  to the  $j$ -th neuron on the output layer  $O$ ,

$u_o(j)$  is the output of the  $j$ -th neuron of the output layer  $O$ ,

$f_o$  is the activation function of the neurons of the output layer  $O$ .

ReLU was used as quality  $f_{c_l}$ ,  $f_{d_l}$  softmax was used as quality  $f_o$ .

### 3.2.2. Method for identifying parameters of a neural network model based on the Adam algorithm

step 1. Initialization.

step 1.1. The initial vector of weights is specified  $\mathbf{w}(0)$ .

step 1.2. The initial vector of the first moments is specified  $\mathbf{m}(-1) = \mathbf{0}$ .

step 1.3. The initial vector of the second moments is specified  $\mathbf{v}(-1) = \mathbf{0}$ .

step 1.4. The parameter is set  $\eta$  to determine the learning rate (usually  $\eta = 0.001$ ), the decay rates of the first and second moments  $\beta_1$  and  $\beta_2$ , respectively,  $\beta_1, \beta_2 \in [0, 1)$  (usually  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ ), as well as the stability parameter  $\varepsilon$  to prevent division by zero (usually  $\varepsilon = 10^{-8}$ ).

step 1.5. The initial gradient is calculated  $\mathbf{g}(0)$ .

step 1.6.  $n = 0$ .

step 2. The vector of first moments is calculated based on the exponential moving average  $\mathbf{m}(n) = \beta_1 \mathbf{m}(n-1) + (1 - \beta_1) \mathbf{g}(n)$ .

step 3. The vector of second moments is calculated based on the exponential moving average  $\mathbf{v}(n) = \beta_2 \mathbf{v}(n-1) + (1 - \beta_2) \mathbf{g}^2(n)$ .

step 4. The vector of weights is calculated (the moments are corrected due to their initialization to zero and the learning step is scaled)

$$\hat{\mathbf{m}}(n) = \mathbf{m}(n)/(1 - \beta_1^{n+1}), \hat{\mathbf{v}}(n) = \mathbf{v}(n)/(1 - \beta_2^{n+1}), \mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\eta \hat{\mathbf{m}}(n)}{\sqrt{\hat{\mathbf{v}}(n) + \epsilon}}.$$

### 3.2.3. Method for classifying a sequence of frames of gymnastic elements based on the Viterbi algorithm

To avoid numerous multiplications during the operation of the Viterbi algorithm, you can logarithmize all the parameters of the model and move from multiplications to addition, since addition is much simpler to implement and faster to calculate. The modified Viterbi algorithm is described as follows:

1. Preprocessing:

$$\hat{\pi}_j = \ln \pi_j, 1 \leq j \leq N, \hat{b}_j(\mathbf{o}_t) = \ln b_j(\mathbf{o}_t), 1 \leq j \leq N, 1 \leq t \leq T, \hat{a}_{ij} = \ln a_{ij}, 1 \leq i, j \leq N.$$

2. Initialization:

$$\hat{\delta}_1(j) = \hat{\pi}_j + \hat{b}_j(\mathbf{o}_1), 1 \leq j \leq N, \psi_1(j) = 0, 1 \leq j \leq N.$$

3. Recursion:

$$\hat{\delta}_{t+1}(j) = \max_{1 \leq i \leq N} [\hat{\delta}_t(i) + \hat{a}_{ij}] + \hat{b}_j(\mathbf{o}_{t+1}),$$

$$\psi_{t+1}(j) = \arg \max_{1 \leq i \leq N} [\hat{\delta}_t(i) + \hat{a}_{ij}], 1 \leq t \leq T-1, 1 \leq j \leq N.$$

4. End:

$$\ln P = \max_{1 \leq i \leq N} [\hat{\delta}_T(i)], q_T^* = \arg \max_{1 \leq i \leq N} [\hat{\delta}_T(i)].$$

5. Restoring the path (sequence of states):

$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T-1, T-2, \dots, 1.$$

### 3.2.4. Quality criteria selection for the method of intellectual classification of gymnastic elements

In the work, to assess the identification of neural networks parameters, the following were selected:

- accuracy criterion

$$Accuracy = \frac{1}{I} \sum_{i=1}^I [\mathbf{d}_i = \hat{\mathbf{y}}_i] \rightarrow \max_W,$$

$$\hat{y}_{ij} = \begin{cases} 1, & j = \arg \max_z y_{iz} \\ 0, & j \neq \arg \max_z y_{iz} \end{cases}$$

- categorical cross-entropy criterion

$$CCE = -\frac{1}{I} \sum_{i=1}^I \sum_{j=1}^K d_{ij} \ln y_{ij} \rightarrow \min_W,$$

where  $\mathbf{y}_i$  is the  $i$ -th vector according to the model,  $y_{ij} \in [0,1]$ ,

$\mathbf{d}_i$  is the  $i$ -th test vector,  $d_{ij} \in \{0,1\}$ ,

$I$  is the power of the training set,

$K$  is the number of classes (neurons in the output layer),

$W$  is the vector of weights;

- performance criterion

$$T \rightarrow \min.$$

## 4. Experiment

A numerical study was carried out based on the dataset [31]. RG Rotate Dataset consists of 49 examples of performing a turn in the back split position without using the hands, with the torso horizontal (Split back without help, trunk horizontal). The data were collected from the video broadcast of the final stage of the 2021 Olympic Games in Tokyo. The examples consist of elements performed by 8 different gymnasts with 4 types of apparatus. Each example consists of an ordered set of images, the number of images in the example depends on the duration of the athlete's performance of the element. This structure allows you to store changes in body position when performing a rotation element. One second of execution is described by 30 frames. The data set is divided into a training set of 39 examples and a test set of 10 examples of element execution. The total dataset size for the 49 examples was 7,355 record images. No preprocessing of the data set was performed. From the datasets, 80% of the images were randomly selected for the training set and 20% of the images for the validation and test sets. Due to the fact that deep neural networks do not contain recurrent connections, training was carried out using GPU. To implement the proposed neural networks, the tensorflow package was used, Google was chosen as the software environment Collaboratory.

The frames of one example of execution show the body positions when performing a rotation element (Fig. 3).



**Figure 3:** Example figure caption

Table 1 presents the structure of a neural network model based on MLP, where  $K$  is the number of classes.

**Table 1**  
**MLP- based neural network model**

Layer type	Input size
Input	1280x720
Resizing	32x32
Full connect or Dense (1 layer)	1024
Full connect or Dense (2 layer)	1024
Output (Full connect or Dense)	$K$

Table 2 presents the structure of a neural network model based on 2 D LeNet, where  $K$  is the number of classes.

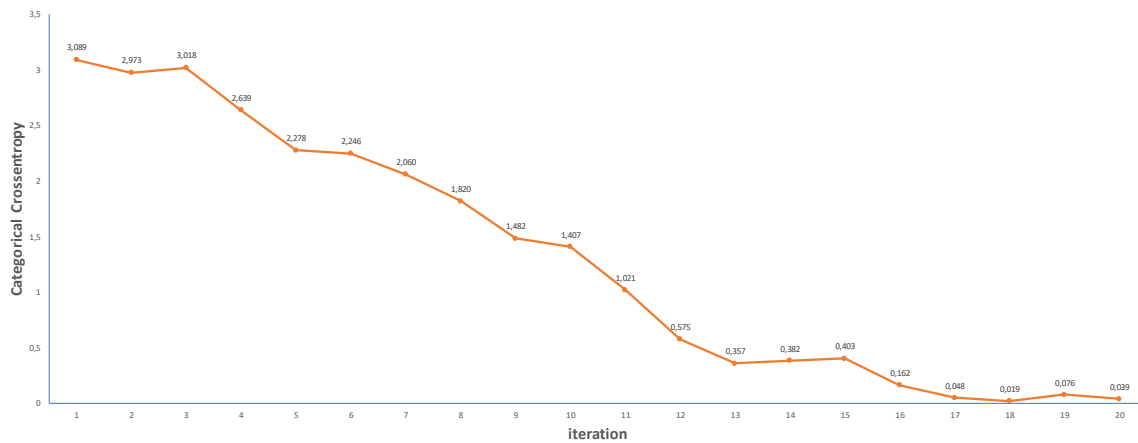


**Table 2**  
**2D neural network model LeNet**

Layer type	Input size
Input	1280x720
Resizing	32x32
Conv2D	32x32x4
MaxPooling2D	16x16x4
Conv2D	16x16x16
MaxPooling2D	8x8x16
Flatten	1024
Full connect or Dense (1 layer)	1024
Full connect or Dense (2 layer)	1024
Output (Full connect or Dense)	K

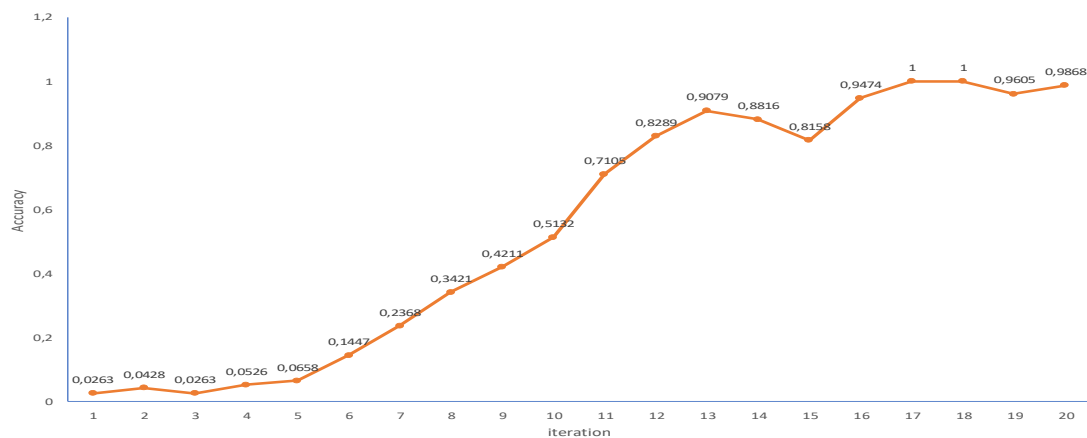
## 5. Results

Fig. 4 shows the dependence of losses (based on categorical entropy) on the number of iterations for the three-layer MLP model.



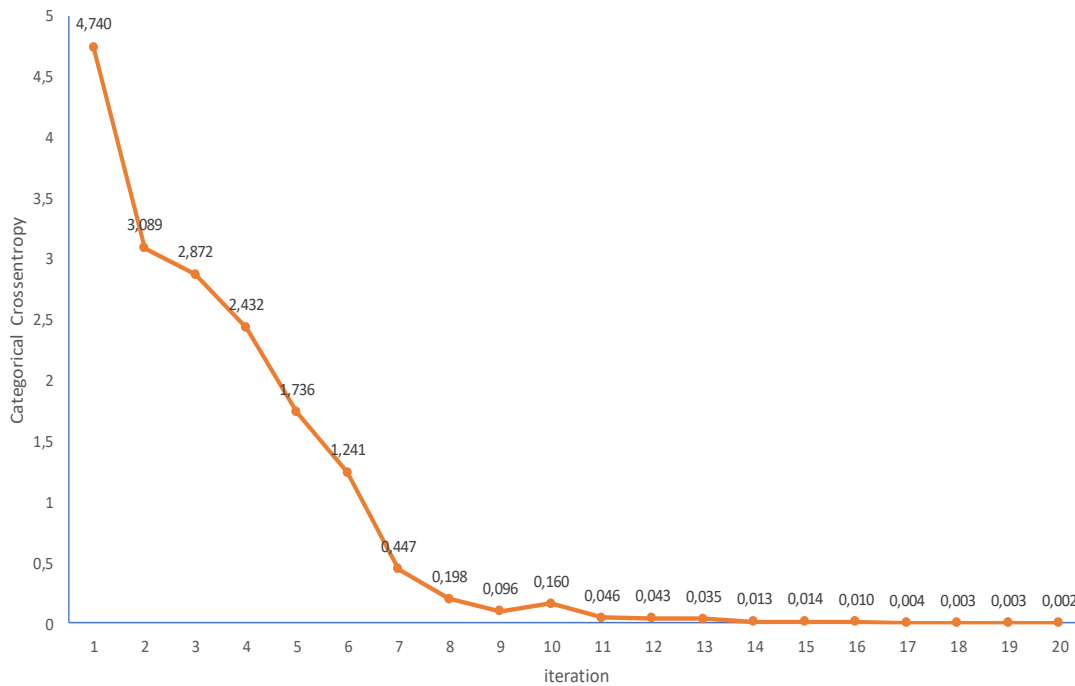
**Figure 4:** Losses dependence (based on categorical entropy) on the number of iterations for a model based on a three-layer MLP

Fig. 5 shows the accuracy dependence on the number of iterations for a model based on a three-layer MLP.



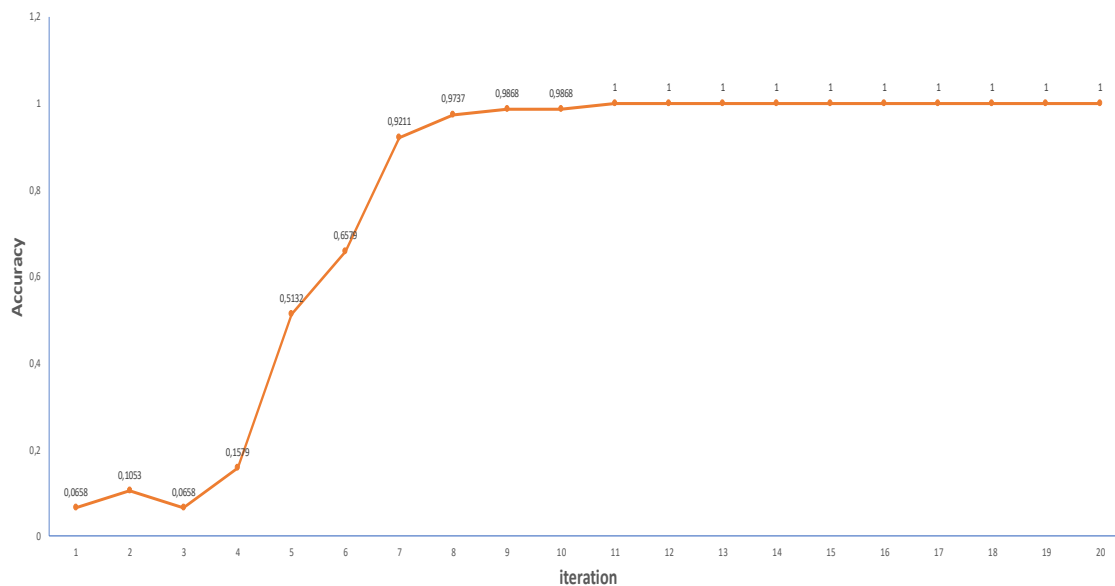
**Figure 5:** Accuracy dependence on the number of iterations for a model based on a three-layer MLP

Figure 6 shows the losses dependence (based on categorical entropy) on the number of iterations for the 2 D model LeNet.



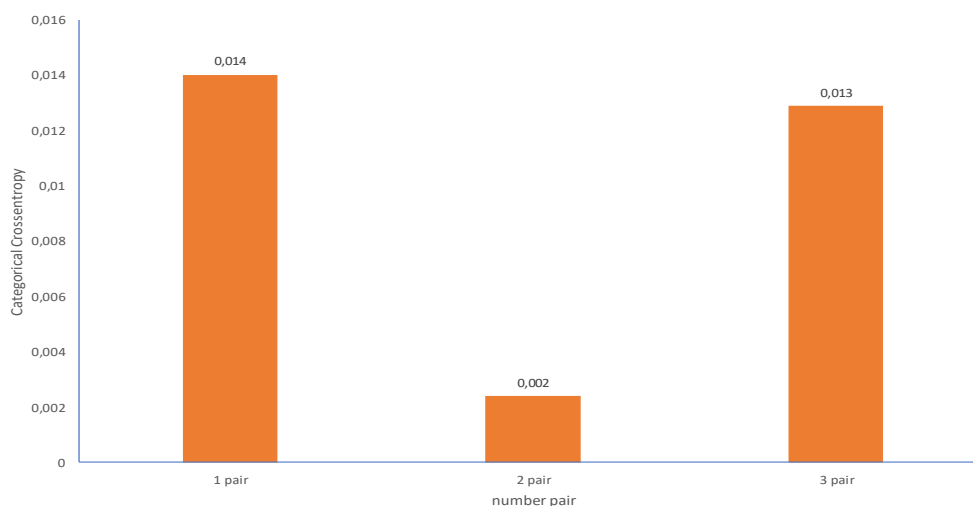
**Figure 6:** Losses dependence (based on categorical entropy) on the number of iterations for a 2D model LeNet

Figure 7 shows the dependence of accuracy on the number of iterations for a model based on 2 D LeNet.



**Figure 7.** Accuracy dependence on the number of iterations for a 2D model LeNet

Figure 8 shows the dependence of the loss (based on categorical entropy) on the number of pairs “convolutional layer – downsampling layer” for the 2D model LeNet.



**Figure 8:** Losses dependence (based on categorical entropy) on the number of convolutional layer–downsampling layer pairs for a 2D-based model LeNet

## 6. Discussions

As a result of the numerical study, the following was established:

- the minimum number of iterations for a neural network model based on a three-layer MLP in terms of losses (based on categorical entropy) (according to Fig. 3) and accuracy (according to Fig. 4) is 18;
- minimum number of iterations for a 2D neural network model LeNet in terms of loss (based on categorical entropy) (according to Fig. 5) and accuracy (according to Fig. 6) is 11;
- the best number of “convolutional layer – downsampling layer” pairs for a 2D neural network model LeNet in terms of loss (based on categorical entropy) is 2 (according to Fig. 7).

To prevent overfitting, the KFold cross-entropy method with a number of folds of 5 was used.

## 7. Conclusions

1. To solve the problem of increasing the efficiency of classification of gymnastic elements, corresponding artificial intelligence methods were investigated. These studies have shown that today the most effective is the use of hidden Markov models (generative approach) and neural networks (descriptive approach).
2. The created method has the following advantages: the input image is not square, which expands the scope of application; the number of pairs “convolutional layer – downsampling layer” is determined empirically, which increases the accuracy of identification by model; the number of planes is defined as the quotient of the number of cells in the input layer divided by two to the power of two (the power is equal to twice the number of the pair “convolutional layer - downsampling layer”) to preserve the total number of cells in the layer after downsampling, which halves the size of the layer planes by height and width, which automates the determination of the structure of the model layers; the use of a neural network allows us to label frames of gymnastic elements, and the use of a generative approach allows the resulting sequence of labeled frames of gymnastic elements analyze effectively.
3. Further prospects for research are the use of the proposed method of intelligent classification for various intelligent visual image recognition systems.

## References

- [1] A. Neskrodieva, M. Strutovskyi, A. Baiev, O. Vietrov. Real-time Classification, Localization and Tracking System (Based on Rhythmic Gymnastics), in: Proceedings of the IEEE 13th International Conference on Electronics and Information Technologies, 14.11(2023): 11–16. doi:10.1109/elit61488.2023.10310664
- [2] S. Goumiri, D. Benboudjema, and W. Pieczynski. "A new hybrid model of convolutional neural networks and hidden Markov chains for image classification". *Neural Computing and Applications*, volume 35, May (2023): 17987–18002. doi:10.1007/s00521-023-08644-4.
- [3] B. Mor, S. Garhwal, A. Kumar. A Systematic Review of Hidden Markov Models and Their Applications. *Arch Computat Methods Eng* Vol. 28, (2021): 1429–1448. doi.org/10.1007/s11831-020-09422-4.
- [4] J. Zhang, J. Sun, J. Wang, Z. Li, X. Chen. An object tracking framework with recapture based on correlation filters and siamese networks. *Comput. Electr. Eng.* 98, 107730 (2022). doi:10.1016/j.compeleceng.2022.107730.
- [5] T. Ding, K. Feng, Ya. Wei, Yu Han, T. Li. DeoT: an end-to-end encoder-only Transformer object detector. *Journal of Real-Time Image Processing*. Vol.20, Issue 1 (2023). doi:10.1007/s11554-023-01280-0.
- [6] L Liu, B. Lin, Y. Yang. Moving scene object tracking method based on deep convolutional neural network. *Alexandria Engineering Journal*. Vol.86 (2024): 592-602 doi:10.1016/j.aej.2023.11.077.
- [7] R. Solovyev, W. Wang, T. Gabruseva: Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*. Vol. 107 (2021): 1-6. doi:10.1016/j.imavis.2021.104117.
- [8] T. Neskrodieva, E. Fedorov. Method for automatic analysis of compliance of expenses data and the enterprise income by neural network model of forecast, in: Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science. CEUR Workshop, 2631, Lviv-Shatsk, 2020, pp. 145–158. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088880635&partnerID=40&md5=c0564b0cbe18017126f328fd3a4779c4>
- [9] L. Wan, Y. Chen, H. Li, and C. Li, "Rolling-Element Bearing Fault Diagnosis Using Improved LeNet-5 Network." *Sensors*, vol. 20, 2020, no. 6, p. 1693. doi:10.3390/s20061693.
- [10] X. Ouyang et al., "A 3D-CNN and LSTM Based Multi-Task Learning Architecture for Action Recognition." *IEEE Access*, vol. 7, (2019): 40757–40770. doi:10.1109/access.2019.2906654.
- [11] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks." *Energy*, vol. 182, pp. 72–81. Sep. 2019. doi:10.1016/j.energy.2019.05.230.
- [12] R. Yang et al. CNN-LSTM deep learning architecture for computer vision-based modal frequency detection. *Mechanical Systems and Signal Processing*, vol. 144, (2020): 106885. doi:10.1016/j.ymssp.2020.106885.
- [13] A. Kumar, Z. J. Zhang, and H. Lyu. Object detection in real time based on improved single shot multi-box detector algorithm. *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 10(2020). doi: 10.1186/s13638-020-01826-x.
- [14] W. Tang, J. Sun, Sh. Wang, Yu. Zhang. Review of AlexNet for Medical Image Classification. – arXiv preprint arXiv.2311.08655. 2023: 1-13.
- [15] G.S.Ch. Kumar, R.K. Kumar, K.P.V. Kumar, N.R. Sai, M. Brahmaiah. Deep residual convolutional neural Network: An efficient technique for intrusion detection system. *Expert Systems With Applications* Vol. 238 (2024): 1-16. doi:10.1016/j.eswa.2023.121912.
- [16] S. Wang, J. Tian, P. Liang, X. Xu, Zh. Yu, S. Liu, D. Zhang. Single and simultaneous fault diagnosis of gearbox via wavelet transform and improved deep residual network under imbalanced data. *Engineering Applications of Artificial Intelligence*. Vol. 133 (2024): 1-17. doi:10.1016/j.engappai.2024.108146

- [17] F. E. L. da Cruz, G. Corso, G. Z. dos Santos Lima, S. R. Lopes, and T. de Lima Prado. Statistical inference for microstate distribution in recurrence plots. *Physica D: Nonlinear Phenomena*, vol. 459. (2024):134048. doi:10.1016/j.physd.2023.134048.
- [18] Detection of COVID-19 chest X-ray using support vector machine and convolutional neural network. *Communications in Mathematical Biology and Neuroscience*, 2020, doi:10.28919/cmbn/4765.
- [19] Jia, Jia, P. Lv, X. Wei, W. Qiu. SNO-DCA: A model for predicting S-nitrosylation sites based on densely connected convolutional networks and attention mechanism. *Heliyon* Vol.10 (2024): 1-11. doi:10.1016/j.heliyon.2023.e23187
- [20] F.B.N. Barber, A.E. Oueslati. Human exons and introns classification using pre-trained Resnet-50 and GoogleNet models and 13-layers CNN model. *Journal of Genetic Engineering and Biotechnology* Vol.22 (2024):1-8. doi:10.1016/j.jgeb.2024.100359
- [21] H. Wang, Sh. Xu, K.-b. Fang, Zh.-Sh. Dai, G.-Zh. Wei, L.-F. Chen. Contrast-enhanced magnetic resonance image segmentation based on improved U-Net and Inception-ResNet in the diagnosis of spinal metastases. *Journal of Bone Oncology*. Vol.42 (2023): 1-9. doi:10.1016/j.jbo.2023.100498.
- [22] M.N. Khan, S. Das, J. Liu. Predicting pedestrian-involved crash severity using inception-v3 deep learning model. *Accident Analysis and Prevention*. Vol.197 (2024): 1-17. doi:10.1016/j.aap.2024.107457.
- [23] X. Tang, F.R. Sheykhahmad. Boosted dipper throated optimization algorithm-based Xception neural network for skin cancer diagnosis: An optimal approach. *Heliyon*. Vol.10 (2024): 1-21. doi:10.1016/j.heliyon.2024.e26415.
- [24] D. Garg, G.K. Verma, A.K. Singh. EEG-based emotion recognition using MobileNet Recurrent Neural Network with time-frequency features. *Applied Soft Computing*. Vol.154 (2024): 1-14. doi:10.1016/j.asoc.2024.111338.
- [25] L. Geng, Y. Hu, Z. Xiao, and J. Xi. Fertility Detection of Hatching Eggs Based on a Convolutional Neural Network. *Applied Sciences*, vol. 9, no. 7, (2019): 1408. doi:10.3390/app9071408.
- [26] A.M. Rifai, S. Raharjo, E. Utami, D. Ariatmanto. Analysis for diagnosis of pneumonia symptoms using chest X-ray based on MobileNetV2 models with image enhancement using white balance and contrast limited adaptive histogram equalization (CLAHE). *Biomedical Signal Processing and Control*. Vol.90 (2024): 1-8. doi.org/10.1016/j.bspc.2023.105857.
- [27] T. Neskrodieva, E. Fedorov, M. Chychuzhko, and V. Chychuzhko, Metaheuristic method for searching quasi-optimal route based on the ant algorithm and annealing simulation. *Radioelectronic and computer systems*, no. 1, (2022): 92–102. doi:10.32620/reks.2022.1.07.
- [28] Yi. Liu, Zh. Wang, R. Wang, J. Chen, H. Gao. Flooding-based MobileNet to identify cucumber diseases from leaf images in natural scenes. *Computers and Electronics in Agriculture*. v. 213, (2023): 1-12. doi:10.1016/j.compag.2023.108166
- [29] P.A. Arjun, S. Suryanarayan, R.S. Viswamanav, S. Abhishek, T. Anjali. Unveiling Underwater Structures: MobileNet vs. EfficientNet in Sonar Image Detection. *Procedia Computer Science*. v. 233 (2024): 518-527. doi:10.1016/j.procs.2024.03.241
- [30] T. Neskrodieva, E. Fedorov. Method for Automatic Analysis of Compliance of Settlements with Suppliers and Settlements with Customers by Neural Network Model of Forecast." *Mathematical Modeling and Simulation of Systems (MODS'2020)*. (2020): 156–165. doi:10.1007/978-3-030-58124-4\_15
- [31] Dataset RG Rotate, 2024. URL:  
[https://drive.google.com/file/d/1HpLAu5esBvsi0VZ0YFywdzlc71B\\_KQcR/view?usp=sharing](https://drive.google.com/file/d/1HpLAu5esBvsi0VZ0YFywdzlc71B_KQcR/view?usp=sharing)