The Neural Network for Emotions Recognition under Special Conditions

Mariia Tiahunova[a], Olesia Tronkina[a], Galina Kirichek[a] and Stepan Skrupsky[a]

[a] *National University "Zaporizhzhia Polytechnic", Zhukovsky str., 64, Zaporizhzhia, 69063, Ukraine*

Abstract

This paper describes a convolutional neural network for emotions recognition considering such special cases like low light conditions, objects on a human face (glasses) and various foreshortening. For correct recognition a dataset that suits the task was found and used, the data from the dataset was normalized, a model was built and trained, metrics with optimizers and a loss function were chosen. The model was tested on test data, its parameters were chosen in accordance with the reference values. All emotion (happy, surprise, angry, fear, neutral, sad and disgust) was recognized with a minimum accuracy of 76% and maximum accuracy of 100%.

Keywords 1

Artificial Intelligence, Deep Learning, Facial Emotion Recognition, Convolutional Neural Network

1. Introduction

Emotion is a process that reflects a person's subjective attitude to a stimulus. It is emotions that induce a person to take action, determine his health, and sometimes block thinking. Thanks to emotions, we understand what mental state a person is, on the basis of these conclusions, we choose a further model of behavior [1]. Emotions are also a signal of a person's desire and need for something. Emotions can save lives during an emergency [2].

In 1992, Paul Ekman, a psychologist at the University of California, San Francisco, described 6 basic emotions that people of all cultures experience and can recognize. This is happy, anger, fear, sadness, surprise, disgust [3]. Any emotion cannot be hidden because for every basic emotion there is an unmistakable facial expression. Even if someone deliberately tries to cheat, or if social norms prohibit the display of basic emotions, micro-expressions always betray them.

The current level of development allows you to create new algorithms and computer vision systems. With the increase in the power of graphics processors, the increase in Random Access Memory (RAM), it became possible to create emotion recognition systems on an ordinary computer [4, 5].

Thanks to the development of computer vision technologies, emotion recognition has become more accurate and accessible for a wide range of people. The software can be integrated into cameras and read people's emotions. Also, emotion recognition systems can be used not only in the field of security, but also in robotics, marketing, targeting, biometrics, bioengineering, as well as contactless payments. Using this system, you can check people for honesty in interviews or interrogations. These developments have long been used on street cameras in developed countries, which helps to reduce the crime rate. Also, many applications, thanks to access to the camera, read emotions and, draw conclusions about the interests of a person, and then, based on these interests, offer content.

CMIS-2021: The Fourth International Workshop on Computer Modeling and Intelligent Systems, April 27, 2021, Zaporizhzhia, Ukraine
EMAIL: mary.tyagunova@gmail.com (M. Tiahunova); olesyatronkina17@gmail.com (O. Tronkina); kirgal08@gmail.com (G. Kirichek); sskrupsky@gmail.com (S. Skrupsky)
ORCID: 0000-0002-9166-5897 (M. Tiahunova); 0000-0001-5239-2358 (O. Tronkina); 0000-0002-0405-7122 (G. Kirichek); 0000-0002-9437-9095 (S. Skrupsky)

© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-2864/paper11.pdf

2. Literature review

The idea of using machine learning is quite old and widespread. Many scientists tried to achieve better results by increasing the complexity of the algorithms, creating their own databases, using powerful hardware, training models for months. In general, the positions of researchers can be divided into two conditional groups. The first group includes studies in which the authors' attention is focused on the study of algorithms [6, 7]. The second group includes studies that analyze databases for better algorithm performance. Two of the most representative examples from each group are described below.

In the article by Shan Li and Weihong Deng “Deep Facial Expression Recognition: A Survey” [8] datasets for building deep learning algorithms. In this article, Shan Li and Weihong Deng discuss the available datasets, the work of modern algorithms on this data. The paper also describes the problems of using each algorithm, and assesses their quality. They made a diagram of the recognition accuracy on each of the known datasets. Each stage of the implementation of the system is described on the basis of basic knowledge and based on suggestions for frequent uses of the implementation for each stage. The uniqueness lies precisely in the overview of databases, their assessment and what data is better to choose for modern deep neural networks. Learning strategies are also described in the article based on static and dynamic images, the advantages and disadvantages are discussed. Based on this scientific work, a database was selected for development in this article, as well as the deep neural network algorithm that is used in this article.

For the second example, the article by Byoung Chul Ko "A Brief Review of Facial Emotion Recognition Based on Visual Information" [9]. It describes the classic and the simplest way to determine emotion from a face image is based on the classification of facial landmarks, the coordinates of which can be obtained using various algorithms. A method is described in which usually from 5 to 68 points are marked, tying them to the position of the eyebrows, eyes, lips, nose, jaw, which allows you to partially capture facial expressions. Normalized coordinates of points can be directly fed into a classifier (for example, SVM or Random Forest) and get a basic solution. Naturally, the position of the faces should be aligned. The simple use of coordinates without a visual component leads to a significant loss of useful information, therefore, to improve the system, various descriptors are calculated at these points: LBP, HOG, SIFT, LATCH, etc. After concatenating the descriptors and reducing the dimension using PCA, the resulting feature vector can be used for classification emotions.

Thus, the task of determining emotions is not fully understood and realized. This is due to insufficient data, high quality images are difficult to find and require a lot of storage space, difficult lighting conditions, and every emotion is subjective. It is especially difficult to pinpoint an emotion if the face is tilted. It is because of the angle of inclination that the key point recognition algorithm does not correctly identify the emotion. It also takes time and resources to find features, and neural networks differ from other machine learning models in that they skip the feature engineering stage. The subject of research is information models artificial neural networks, as well as the implementation and description of the algorithm convolutional neural networks. The complexity of the implementation lies in the training of the neural network.

3. Facial landmarks method

The classic way to identify emotions is a method based on facial landmarks, key points. In total, the algorithm has 2 stages. At the first stage, you need to identify a feature vector, this is done by a machine learning model based on input data. At the second stage, the already trained model assigns new images of the object to a certain class.

The vector of facial features can be anything, nose, eyes, lips, facial contour, eyebrows, etc. (Fig.1). It is these characteristics that serve as input to the model. Next, two statistical models are built: a shape model and a texture model.

Shape model – a parametric linear model describing the possible options for the position of key points.

Shape – a set of points on the image that make up the contours of the object under study. For

example, the shape of a human hand can be defined through points on the contours of the palm and fingers, and the shape of the face through points on the eyebrows, nose, lips, and outer contour [11].

A texture model is a similar model, but it describes the possible variations in pixel intensity. The texture is the intensity of pixels in the image inside the outer contour of the shape. The number of pixels inside the outer contour of the shape may differ for different images [12].

Therefore, all face images are reduced to a single averaged shape using piecewise affine transformation (Fig. 2).

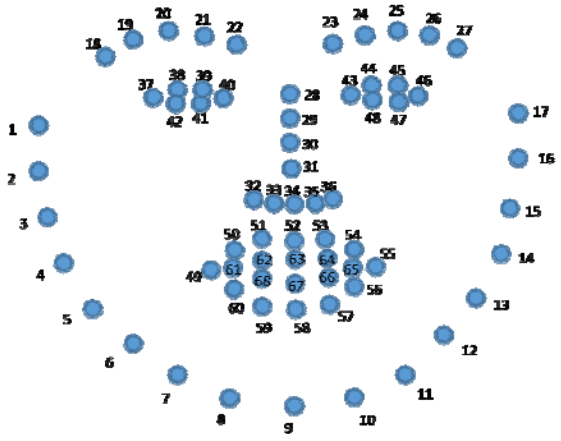

Figure 1: Facial landmarks

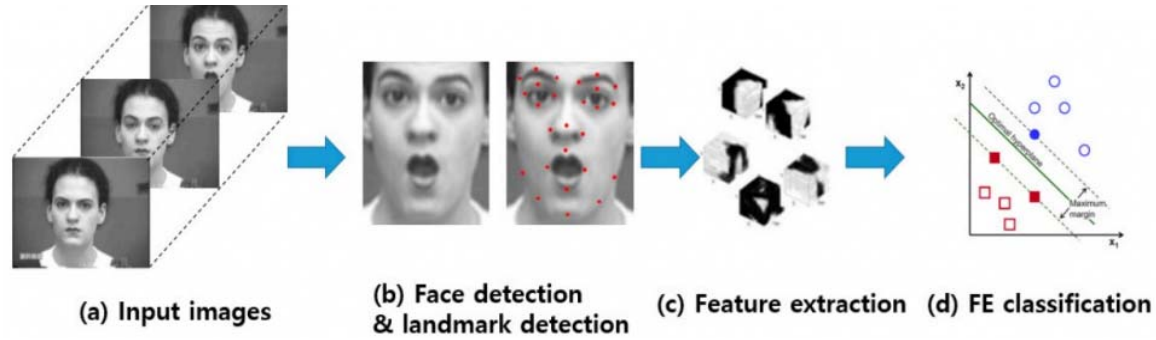

Figure 2: Facial landmarks method

Based on the review of this algorithm, a model is required to find the coordinates of key points. This complicates the algorithm by searching for a large number of features. The position of the images and faces should be aligned with ideal lighting and no noise. For these points need to calculate descriptors, otherwise this will lead to the loss of useful information. To date, the effectiveness of this method is an average of 72.65%. There are a large number of algorithms that make it possible to obtain key points with an accuracy sufficient for further classification of human emotions by these points [13]. But, to use this approach, it is necessary that the position of the face in the image be aligned, and in real conditions, this is almost unrealistic. This method is deprecated. Based on this, convolutional neural networks are the best algorithm for analyzing visual data.

4. Convolutional Neural Networks

Since the method of key points has a number of disadvantages, a replacement was proposed. This is using convolutional neural networks. This is a deep learning algorithm that is a good approach for image classification, based on the input image, importance is assigned to each feature using weights and biases. Image preprocessing requires fewer weights, which means it requires less processing power compared to other machine learning algorithms. For example, in fully connected neural

networks, filters are created manually for sufficient training, and a convolutional neural network is able to identify and learn these filters and features on its own.

This architecture was invented as a result of studying the work of the brain of cats. It turned out that neurons respond to stimuli / lines only in a limited field of view and from these lines they collect a complete picture. These lines are the signs, each sign has a different importance. That is, the convolutional network generalizes all features. Each filter has different features. The more filters there are, the deeper the signs.

The main advantage of such networks over fully connected neural networks is the decrease in the number of trained neurons, and as a result, the acceleration of network learning and a decrease in the required amount of training data [17]. Due to a large number of abstract layers, they provide partial resistance to changes in scale, displacement, rotation, change of perspective, and other distortions [18]. Convenient parallelization of computations, and, consequently, the possibility of implementing algorithms for working and training the network on GPUs.

4.1 Facial Expressions Databases

To start developing a neural network, we will need a sufficient amount of labeled data for training, including 6 classes of basic emotions. Large companies have the resources and the ability to independently collect and tag data. In this paper, only publicly available databases are considered, with already marked up images and representative data.

The Extended CohnKanade (CK+) [19] is the most widely used database for evaluating emotions. Contains about 600 videos and 5876 tagged images of 123 people. The sequences have durations from 10 to 60 frames, these frames show the transition from neutral emotions to the most pronounced ones. The videos are tagged with Ekman's seven main emotions: anger, contempt, disgust, fear, happiness, surprise, sadness. Images are mostly single-channel with the same background and size 640 * 490 pixels. But CK + does not provide a dataset for training, validating, and testing algorithms. Because of this, it is not possible to use this dataset in work.

EMOTIC or EMOTIon [20] this unique dataset contains images of people in their natural environment and is tagged with emotions that are intuitively identified. Most of the images are collected by hand from the internet and search queries. The database contains about 30,000 images of different people, emotions are divided into 2 types, 26 discrete categories. In the three-dimensional coordinate system of emotions, valence, arousal, and dominance. This project was created in order for machines to understand what a person is experiencing based on a coordinate system. Machines with these capabilities interact better with humans.

The FER-2013 [21] This is a classic database that was created for Kaggle competitions, and after the competition, the data was shared. There are 35,000 images in the dataset. The data is divided into training, private test and public test data. Images are labeled with a tag of 7 seven basic emotions: happiness, fear, anger, surprise, disgust, neutral, sadness. The images have already been converted to an array of numbers and placed in a .csv file. Based on this, the obvious advantage is the use of a small amount of memory. Also, this dataset can be freely downloaded to google drive. This will save space. Unfortunately, the size of the images is only 48 * 48 pixels, but all images were automatically registered, the face is in the center and occupies the same space in each image.

4.2 Preparing data

First, we need to look at the data structure using the Pandas library. There are 3 columns in fer2013.csv (Fig. 3). In the first, the number of the emotion, in the second, a set of an array of image pixels, and the third column indicates which data type the image belongs to. Images are NumPy 28x28 arrays with pixel values ranging from 0 to 255.

	emotion	pixels	Usage
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...	Training
1	0	151 150 147 155 148 133 111 140 170 174 182 15...	Training
2	2	231 212 156 164 174 138 161 173 182 200 106 38...	Training
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...	Training
4	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...	Training

Figure 3: Structure of FER2013

Labels are an array of integers ranging from 0 to 6. They correspond to the class of emotion that the image represents. There are three types of sampling in total: training, test, and validation. The training subsampling contains the standard, the input data along with the correct / expect result. The model does not see the test dataset during training, and therefore, on this dataset, we can evaluate the quality of the model. Validation subsampling is needed to select the hyperparameters and select the best model.

Divide the data into 3 subsamples. There are 3 classes of the "Usage" column: "Training" contains 28709 images, "PublicTest" contains 3589 images, "PrivateTest" images. Data from the "Training" column can be used to train the model, data from "PrivateTest" for model validation, and "PublicTest" for test images. The data needs to be transformed so that it has the following structure: a feature vector $X_i = [x_1, \ldots, x_n]$ and a label (target variable) Y_i [Fig. 4]. The graph shows the number of pictures belonging to each class, on the train and validation subsample.

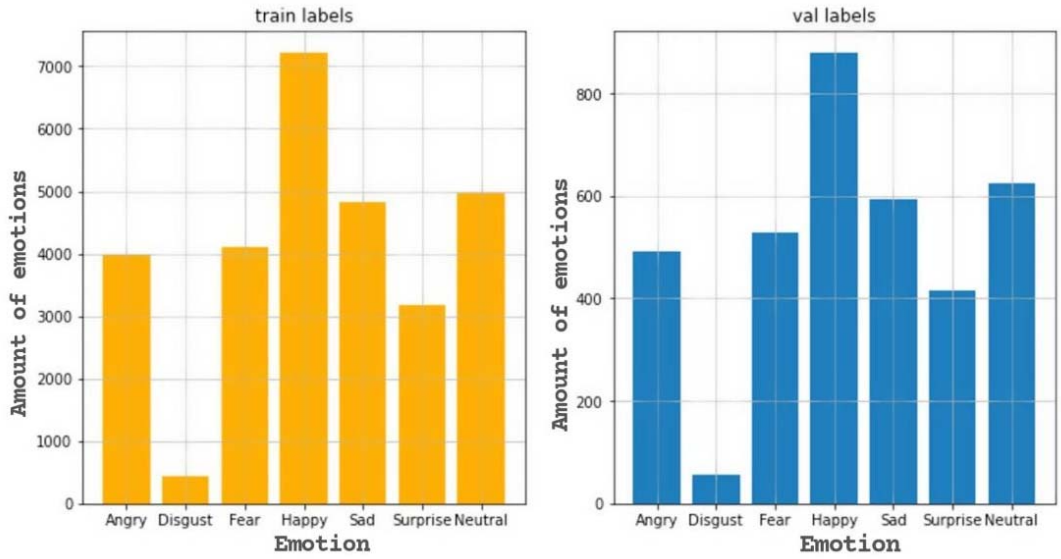

Figure 4: Distribution of data between classes

Since the pixel values fall in the range from 0 to 255, these values need to be scaled. To do this, divide the image array by 255. It is important that the training set and the test set are pre-processed in the same way. Also, the data needs to be converted to the float32 data type. Next, the data must be transformed into a size of 48 * 48 and the number of channels must be 1. In color images the number of channels is 3, and in black and white 1. The array of labels is converted into a one-dimensional vector. The method to_categorical(), converts an array or vector with integers with different categories to a new array with binary values and columns equal to the number of categories in the data. Below you can see the data converted from array to image and also normalized.

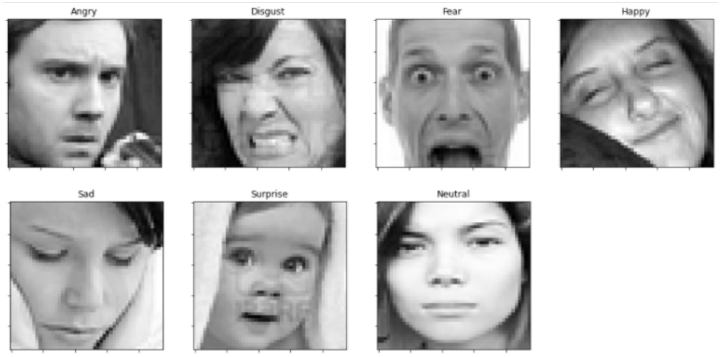

Figure 5: Normalized data

4.3 Neural Network Architecture

In total, the neural network will have 3 convolutional layers, 3 max-pooling layers, a Flatten layer, and 2 fully connected layers (Fig. 7) [22]. The first layer will be convolutional. It is he who is responsible for the convolution operation and the selection of convolution kernels. There are one-dimensional and two-dimensional analogs. On the first convolutional layer, the number of different convolution kernels to be applied will be 32. At the same time, this is the number of feature maps that we get on the next layer after convolution. Next comes the size of the convolution kernel. The most commonly used sizes are 3 * 3, 5 * 5, and 7 * 7, but there may be others, including non-square ones. This architecture uses 3 * 3. The size of the input data is determined from the image size, that is, 48 * 48 * 1. Used only in the first convolutional layer. Further, the input dimension is determined automatically from the output dimension of the previous layer. In the case of a convolutional layer, the dimension necessarily contains three parameters: the number of rows of pixels in the image matrix, the number of columns, the number of color channels (in the case of a black-and-white image, this is 1, and not the absence of channels). Next comes padding. Padding can have two values: "valid" and "same". The default is "valid", which minimizes the image. If you want the image not to shrink after the convolution operation, then set the "same" parameter, which, by adding zero rows and columns around the image, allows you to keep the image size the same. The indent can be filled with units and mirrored [23].

Next comes the layer that normalizes the input data – batchnormalization. When the weights of a layer change, the distribution of its outputs changes, so the next layer will have to start over. This interferes with learning. This problem is called intrinsic variable shifting. The solution is that we normalize each input separately, but not for all images in the dataset, but for mini batches. Next, you need to enter new parameters: shift and stretch. These parameters are also trained by gradient descent. [24]. Relu [25] is used as the activation function. An activation function is a function that accepts the result of an adder as input and performs some kind of transformation to turn the sum of the weighted inputs into an adequate output that can be interpreted in terms of solving the problem. Since in the case of a small value of the summation function, the activation function can return 0, that is, "nothing", then it is a kind of analog of the mechanism in a real neuron that is responsible for the excitation of the neuron (its "activation"). Relu returns x if x is positive, 0 otherwise. It is a good approximator - any function can be approximated by a combination of ReLU.

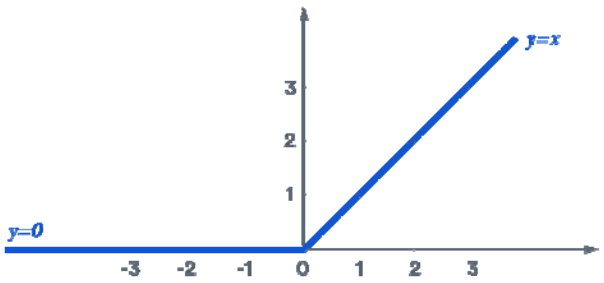

Figure 6: Relu

It also has the advantage that this function creates sparsity when neurons are activated. That is, unlike the sigmoid and the hyperbolic tangent, not all neurons will be activated, which will reduce the computational complexity. This property stems from the fact that all negative values vanish. Using ReLU significantly improves the convergence rate of gradient descent compared to sigmoid and hyperbolic tangent. The next is the pooling layer. The maxspooling layer is 2 * 2, other values are possible, most often this value is used. Typically, a convolutional layer followed by a pooling layer is called a convolutional block. The neural network architecture is visualized using the Tensorboard tool. Figure 7 shows the location of each layer based on the above architecture.

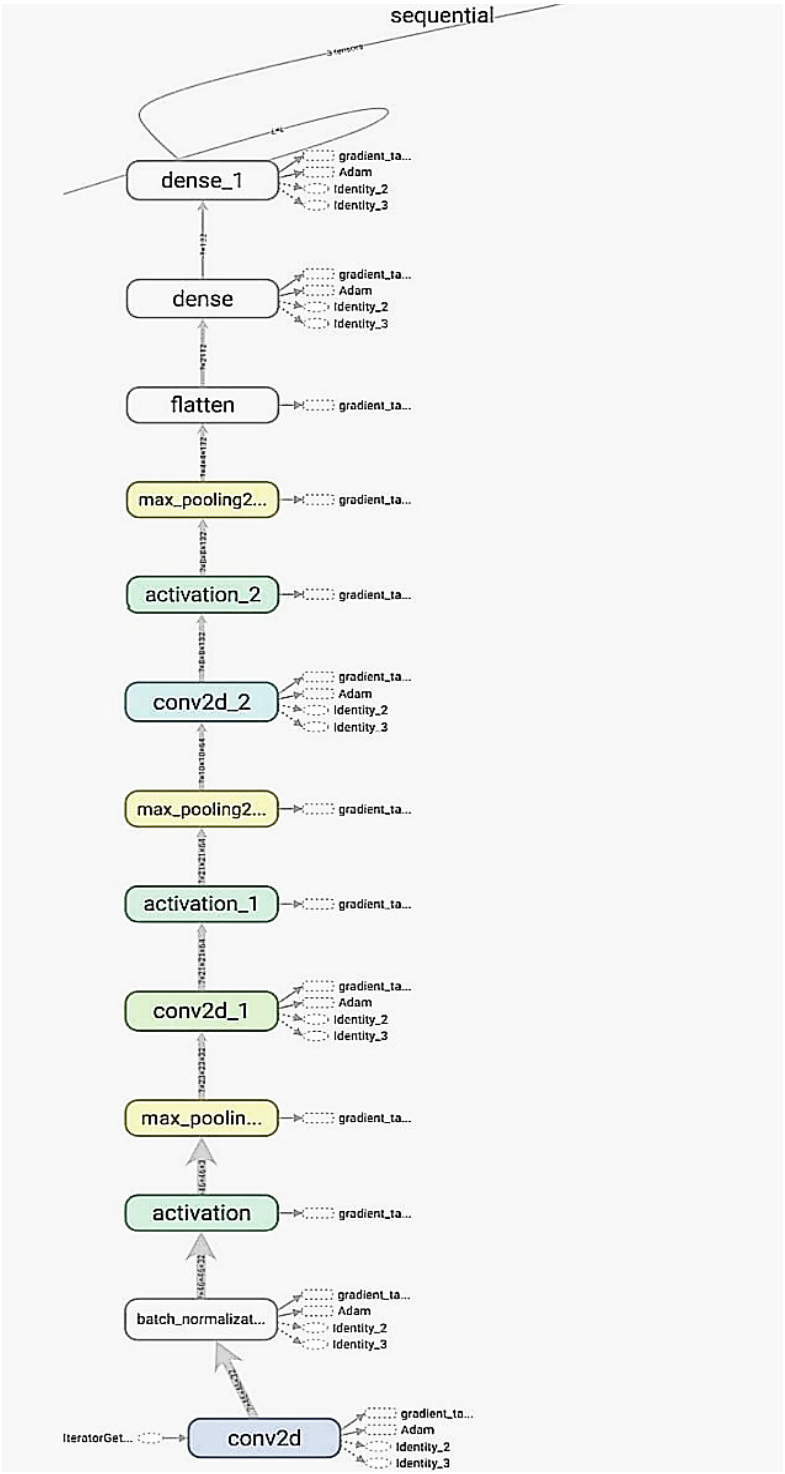

Figure 7: Neural network architecture

After the convolutional blocks come to the Flatten layer. It does not have any parameters, but simply converts multidimensional data into a flat vector, which can already be given as input to a fully connected layer. Convolution blocks will not give us the probability of assigning an image to any class. The fully connected layer is responsible for this. In total, this architecture has 2 fully connected layers. In the first fully connected layer 132 inputs to the neuron. In the second, there are 7 inputs to the neuron, according to the number of classes that need to be predicted. On the last layer there is a softmax function [25]. Since there are 7 inputs to the neuron on the last layer, this is the number of classes. Each neuron should give the probability of how much the object belongs to the desired class, a value from 0 to 1. All neurons should add up to one.
Softmax function (1):

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{Z_k}} \tag{1}$$

5. Model configuration

After creating the model, it needs to be configured with losses and metrics. The categorical cross-entropy [26] was chosen as the loss function. Categorical crossentropy (2) this loss function is commonly used to classify 3 or more classes. The function simply calculates the difference between the two probability distributions and quantifies it. The categorical crossentropy loss function calculates the loss of an example:

$$Loss = \sum_{i=1}^{output\ size} y_i \times \log \hat{y}_i \tag{2}$$

where $\hat{y}_i$ is the i-th scalar value in the model output, y_i is the corresponding target value, and output size is the number of scalar values in the model output. This function is good at showing how two vectors differ from each other.

Adam [25] with learning rate lr = 1e-3 is used as an optimizer. This is a good default algorithm, good convergence. Adam combines the benefits of RMSProp, AdaGrad and SGD + Momentum. We first estimate the first pulse and the weighted sum of the gradients, and then we estimate the second pulse and the square of the gradients. The first impulse plays the role of speed, and the second serves to optimize the parameters. Initially, both parameters are equal to zero, so a correction offset is added to them to avoid too large a step at the very beginning.

Accuracy (3) [25] is used as a quality metric. Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$accuracy = \frac{number\ of\ correct\ predictions}{total\ number\ of\ \ predictions} \tag{3}$$

6. Training and testing

A training dataset was used to train the neural network, and a validation dataset was used for validation. Batch size = 64, epoch = 20. The model works ineffectively if it is given as input and trained from images at once. Therefore, the data was divided into 64. In order for the network to learn, many iterations of 64 must go through. The larger the batch sample size, the more memory is required from the computer. Below are the training schedules. They show that 20 epochs, even 12 epochs can be taken, is quite enough. A little more and there would be overfitting (Fig. 8-9). Figure 8 shows the accuracy at each epoch of the training and validation data. Figure 9 shows the change in the growth of the loss function on the data.

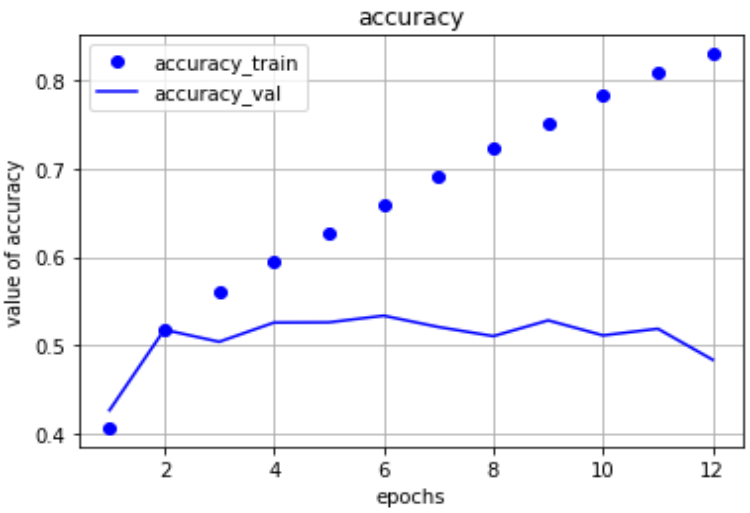

Figure 8: Accuracy

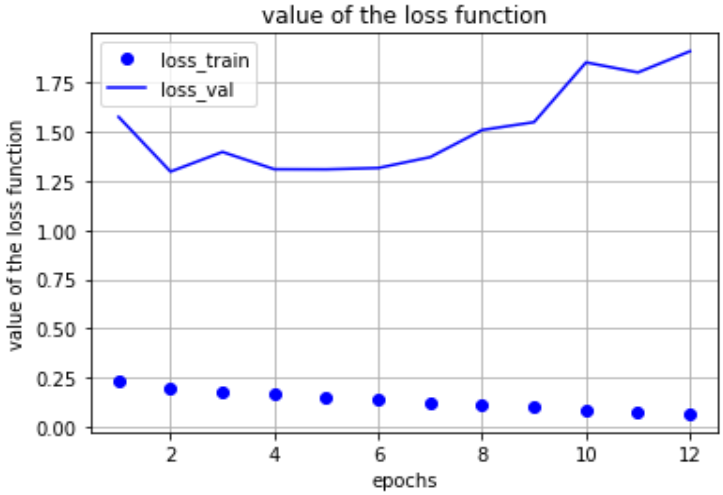

Figure 9: Loss

Figures 10 - 16 show the testing stage. On the 10th figure, the neural network recognized the emotion "happiness" with a probability of 1.0 on the test sample. Figure 11 shows that the neural network recognized the "surprise" emotion with a 1.0 probability. On the 12th figure, the neural network recognized the "anger" emotion with a probability of 0.78 and the "fear" emotion with a probability of 0.22. This is because emotions are similar and subjective and can be similar to each other. Figure 13 shows a similar situation, the neural network recognized 2 emotions, "fear" with a probability of 0.78 and "sad" with a probability of 0.21. In figure 14, the "neutral" emotion was recognized with a probability of 0.76, the "sad" emotion with a probability of 0.22, and the "angry" emotion with a low probability of 0.05. In Figure 15, the convolutional neural network recognized the "sad" emotion with a probability of 0.78 and the "angry" emotion with a probability of 0.18, and a low probability of other emotions. In Figure 16, the neural network recognized the "angry" emotion with a probability of 0.82. Only in figure 16 did the predictions turn out to be wrong. In the remaining figures, the neural network correctly predicted all emotions. Above is a label for each emotion. The model does not always make accurate predictions, it is especially difficult to recognize the disgust emotion, and there were few examples of this emotion in the dataset.

Based on other articles, such as [11], the accuracy on the JAFFE dataset ranged from 0.249 to 0.502. In this article, the accuracy is higher even with the minimal resources of the graphics processing unit (GPU Tesla k80). In the article [27] the accuracy on the training data reached 0.96, which is 0.03 less than what was obtained in this article.

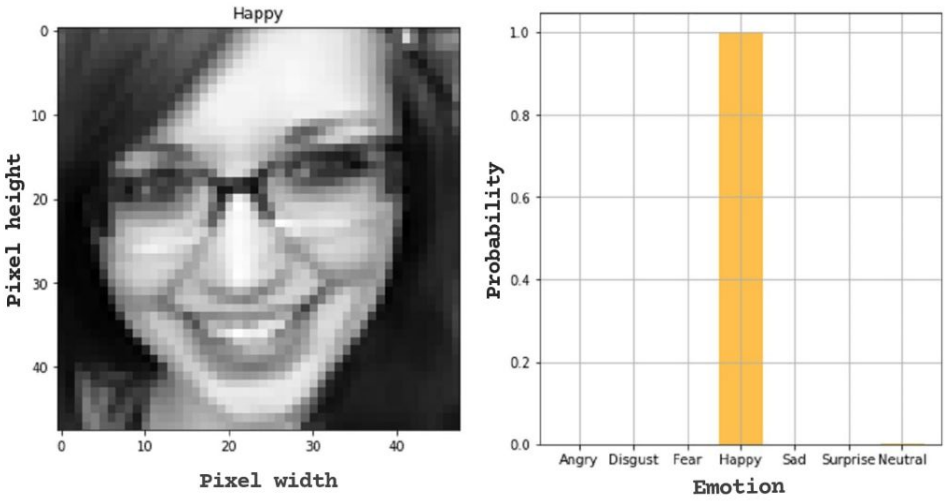

Figure 10: Testing emotion “happy”

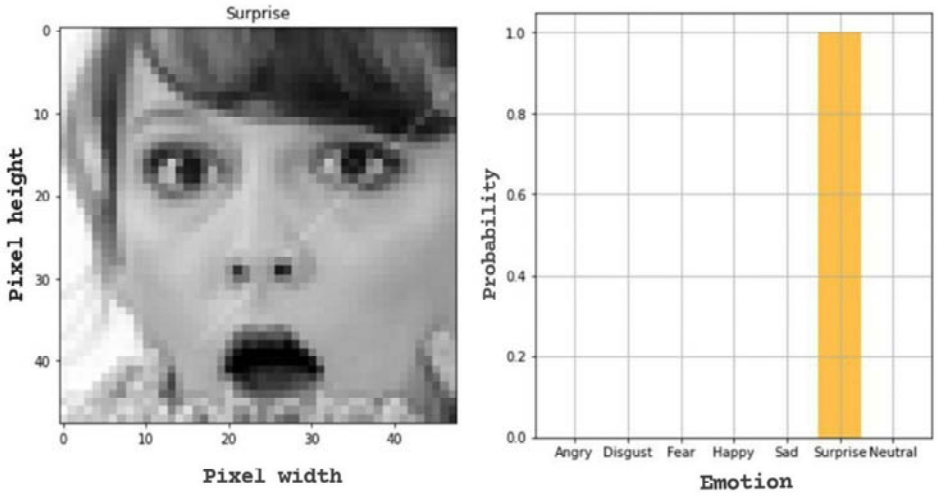

Figure 11: Testing emotion “surprise”

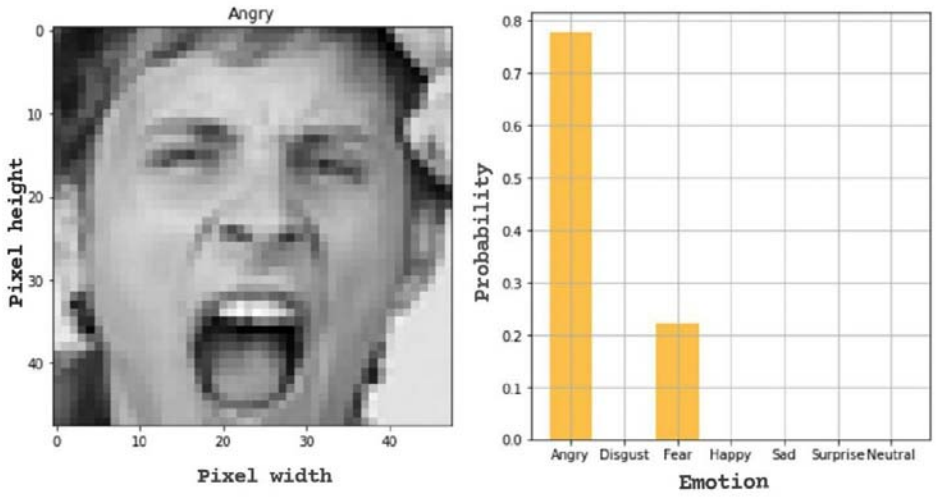

Figure 12: Testing emotion “angry”

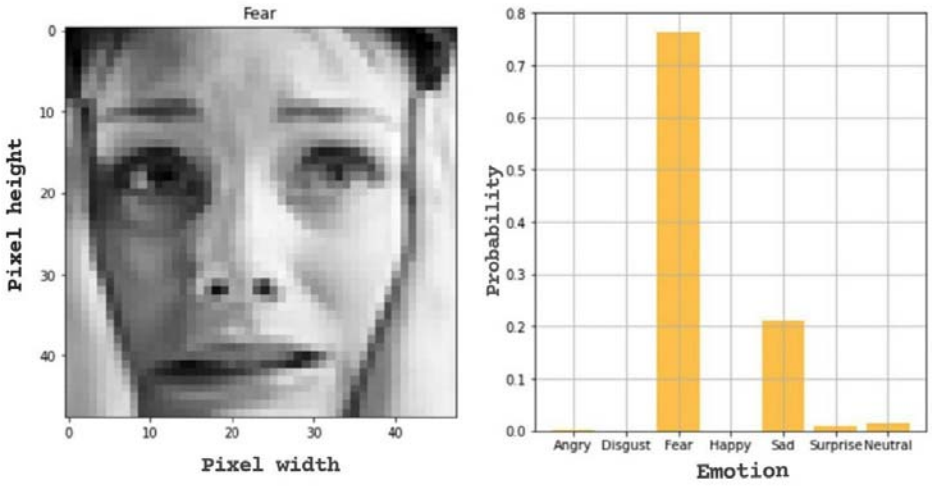

Figure 13: Testing emotion “fear”

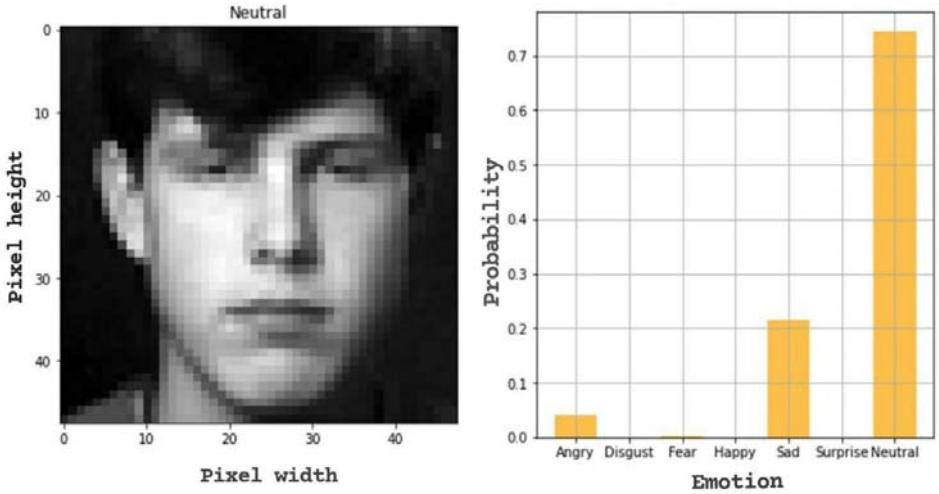

Figure 14: Testing emotion “neutral”

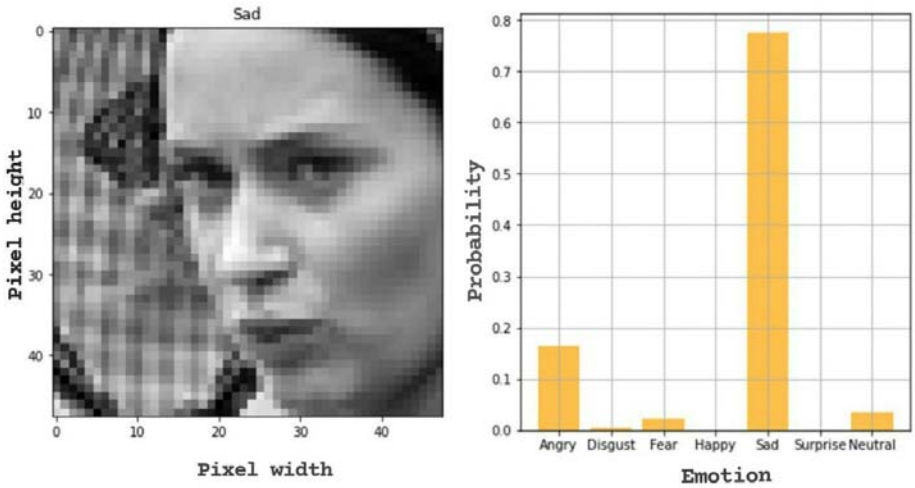

Figure 15: Testing emotion “sad”

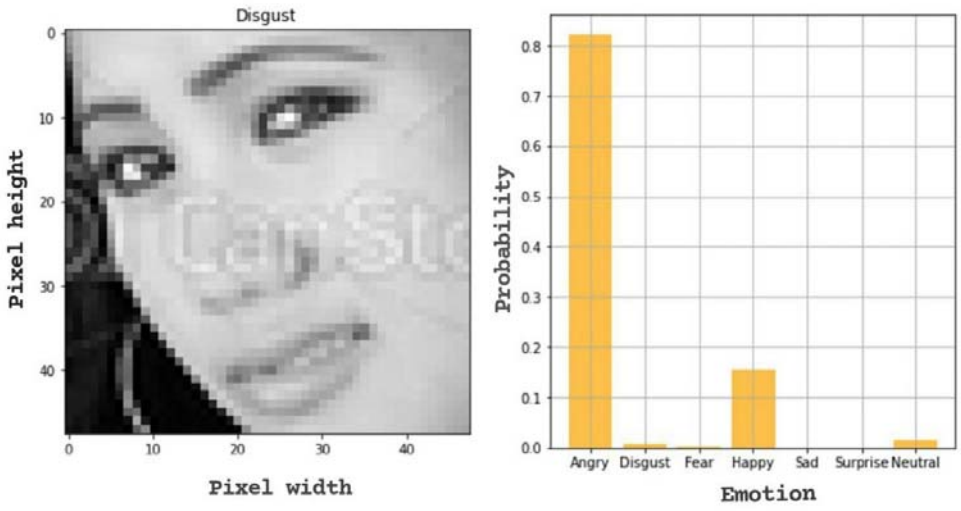

Figure 16: Testing emotion “disgust”

7. Conclusion

The article implements a convolutional neural network capable of recognizing human emotions. This is a solution to a technically and mathematically complex problem, requiring in-depth study of the subject area, own analysis, skills and abilities in the processing and selection of experimental data, deep knowledge in the field of discrete mathematics, geometry, programming, as well as psychology and many other areas of science. In this work, a convolutional neural network is implemented that is capable of detecting emotion in low light conditions, with objects on the face (glasses) and at an angle of inclination. This is what distinguishes this algorithm from the previous ones. For this, was found a dataset suitable for this task, all data from the dataset was normalized, a model was built and trained, metrics with optimizers and a loss function were selected. The model is also tested on test data, and the model parameters are selected on the validation data.

The scientific novelty consists in the fact that this algorithm has a high accuracy, learns quickly and does not require large computing resources. Also, the model can be saved and integrated into any system, such as a video camera. This can be used for commercial purposes. Training this network proved to be optimal for the available machine resources, and at the same time it trained 10 times faster than on a central processor unit. The convolutional neural network model is also capable of identifying features and reducing unnecessary features. Also, a convolutional neural network is able to adapt to image resizing, scaling, rotations, unlike other algorithms.

The practical value lies in the fact that the algorithm is implemented in the form of software that allows you to detect emotions in people in low light conditions, with a small number of computing resources. This algorithm detects a person's emotion every second with a very low probability of error.

References

[1] C. Darwin, P. Prodger, The expression of the emotions in man and animals, 3rd ed, Oxford University Press, 1998.

[2] Y. I. Tian, T. Kanade, J. F. Cohn, Recognizing action units for facial expression analysis, IEEE Transactions on pattern analysis and machine intelligence 23.2 (2001) 97-115. doi:10.1109/34.908962.

[3] P. Ekman, W. V. Friesen, Constants across cultures in the face and emotion, Journal of personality and social psychology 17 (1971) 124–129. doi:10.1037/h0030377.

[4] A. Oliinyk, S. Skrupsky, S. A. Subbotin, Parallel computer system resource planning for synthesis of neuro-fuzzy networks, in: R. Szewczyk, M. Kaliczyńska (Eds.), Recent Advances in Systems, Control and Information Technology, volume 543 of Advances in Intelligent Systems

and Computing, Springer, Cham, 2016, pp. 88-96. doi:10.1007/978-3-319-48923-0_12.
[5] G. Kirichek, S. Skrupsky, M. Tiahunova, A. Timenko, Implementation of web system optimization method, in: Proceedings of the Third International Workshop on Computer Modeling and Intelligent Systems, CMIS '2020, CEUR Workshop Proceedings 2608, Zaporizhzhia, Ukraine, 2020, pp. 199–210.
[6] S. Subbotin, A. Oliinyk, V. Levashenko, E. Zaitseva, Diagnostic rule mining based on artificial immune system for a case of uneven distribution of classes in sample, Communications - Scientific Letters of the University of Zilina 18.3 (2016) 3–11. URL: http://komunikacie.uniza.sk/index.php/communications/article/view/301.
[7] A.O. Oliinyk, T.A. Zaiko, S.A. Subbotin, Factor analysis of transaction data bases, Automatic Control and Computer Sciences 48.2 (2014) 87-96. doi:10.3103/S0146411614020060.
[8] S. Li, W. Deng, Deep Facial Expression Recognition: A Survey, IEEE Transactions on Affective Computing (2020) 1-1. doi:10.1109/TAFFC.2020.2981446.
[9] B. C. Ko, A brief review of facial emotion recognition based on visual information, Sensors 18.2 (2018) 401. doi:10.3390/s18020401.
[10] W. J. Baddar, J. Son, D. H. Kim, S. T. Kim, Y. M. Ro, A deep facial landmarks detection with facial contour and facial components constraint, in: 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 1–5. doi:10.1109/ICIP.2016.7532952.
[11] I. Tautkute, T. Trzcinski, A. Bielski, I know how you feel: Emotion recognition with facial landmarks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2018, pp. 1878-1880. doi:10.1109/cvprw.2018.00246.
[12] Kostinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization, in: 2011 IEEE International Conference on Computer Vision Workshops (ICCVW), Barcelona, Spain, 2011, pp. 2144–2151, doi:10.1109/ICCVW.2011.6130513.
[13] Y. Sun, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), arXiv preprint arXiv:1406.4773, 2014, pp. 1891-1898.
[14] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097–1105. URL: https://kr.nvidia.com/content/tesla/pdf/machine-learning/imagenet-classification-with-deep-convolutional-nn.pdf.
[15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.
[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A.Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1–9. doi:10.1109/cvpr.2015.7298594.
[17] B. Hasani, M. H. Mahoor, Spatio-temporal facial expression recognition using convolutional neural networks and conditional random fields, in: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition, FG '2017, IEEE, Washington, DC, 2017, pp. 790–795. doi:10.1109/fg.2017.99.
[18] G. Kirichek, V. Harkusha, A. Timenko, N. Kulykovska, System for detecting network anomalies using a hybrid of an uncontrolled and controlled neural network, in: Proceedings of the Computer Science Software Engineering: Proceedings of the 2nd Student Workshop, CSSE@SW '2019, CEUR Workshop Proceedings 2546, Kryvyi Rih, Ukraine, 2019, pp. 138–148.
[19] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops, IEEE, San Francisco, CA, USA, 2010, pp. 94–101. doi:10.1109/cvprw.2010.5543262.
[20] C. F. Benitez-Quiroz, R. Srinivasan, A. M. Martinez, Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016.

doi:10.1109/cvpr.2016.600.
[21] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee et al., Challenges in representation learning: A report on three machine learning contests, in: International Conference on Neural Information Processing. Springer, Berlin, Heidelberg, 2013, pp. 117–124. doi:10.1007/978-3-642-42051-1_16.
[22] Z. Meng, P. Liu, J. Cai, S. Han, Y. Tong, Identity-aware convolutional neural network for facial expression recognition, in: 12th IEEE International Conference on Automatic Face & Gesture Recognition, FG '2017, IEEE, Washington, DC, USA, 2017, pp. 558–565. doi:10.1109/fg.2017.140.
[23] G. Levi, T. Hassner, Emotion recognition in the wild via convolutional neural networks and mapped binary patterns, in: Proceedings of the 2015 ACM on international conference on multimodal interaction, 2015, pp. 503–510. doi:10.1145/2818346.2830587.
[24] A. T. Lopes, E. de Aguiar, A. F. De Souza, T. Oliveira-Santos, Facial expression recognition with convolutional neural networks: coping with few data and the training sample order, Pattern Recognition 61 (2017) 610–628. doi:10.1016/j.patcog.2016.07.026.
[25] S.O. Subbotin Neironni merezhi teoriia ta praktyka (Neural networks: theory and practice), 2020.
[26] Peltarion.com, Categorical crossentropy, 2021. URL: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-crossentropy.
[27] M. A. Ozdemir, B. Elagoz, A. Alaybeyoglu, R. Sadighzadeh, A. Akan, Real time emotion recognition from facial expressions using CNN architecture, in: 2019 Medical Technologies Congress (TIPTEKNO), IEEE, Izmir, Turkey, 2019, pp. 529-532. doi:10.1109/TIPTEKNO.2019.8895215.