DOI: https://doi.org/10.15276/hait.06.2023.7
UDC 004.93

The structural tuning of the convolutional neural network for speaker identification in mel frequency cepstrum coefficients space

Anastasiia D. Matychenko [1)]
ORCID: https://orcid.org/0009-0009-7894-4734; matychenko.8089532@stud.op.edu.ua
Marina V. Polyakova[1)]
ORCID: https://orcid.org/0000-0001-7229-7657; marinapolyakova943@gmail.comu. Scopus Author ID: 57017879200
[1)] Odessa Polytechnic National University, 1, Shevchenko Ave. Odessa, 65044, Ukraine

ABSTRACT

As a result of the literature analysis, the main methods for speaker identification from speech signals were defined. These are statistical methods based on Gaussian mixture model and a universal background model, as well as neural network methods, in particular, using convolutional or Siamese neural networks. The main characteristics of these methods are the recognition performance, a number of parameters, and the training time. High recognition performance is achieved by using convolutional neural networks, but a number of parameters of these networks are much higher than for statistical methods, although lower than for Siamese neural networks. A large number of parameters require a large training set, which is not always available for the researcher. In addition, despite the effectiveness of convolutional neural networks, model size and inference efficiency remain important for devices with a limited source of computing power, such as peripheral or mobile devices. Therefore, the aspects of tuning of the structure of existing convolutional neural networks are relevant for research. In this work, we have performed a structural tuning of an existing convolutional neural network based on the VGGNet architecture for speaker identification in the space of mel frequency cepstrum coefficients. The aim of the work was to reduce the number of neural network parameters and, as a result, to reduce the network training time, provided that the recognition performance is sufficient (the correct recognition is above 95 %). The neural network proposed as a result of structural tuning has fewer layers than the architecture of the basic neural network. Instead of the ReLU activation function, the related Leaky ReLU function with a parameter of 0.1 was used. The number of filters and the size of kernels in convolutional layers are changed. The size of kernels for the max pooling layer has been increased. It is proposed to use the averaging of the results of each convolution to input a two-dimensional convolution results to a fully connected layer with the Softmax activation function. The performed experiment showed that the number of parameters of the proposed neural network is 29 % less than the number of parameters of the basic neural network, provided that the speaker recognition performance is almost the same. In addition, the training time of the proposed and basic neural networks was evaluated on five datasets of audio recordings corresponding to different numbers of speakers. The training time of the proposed network was reduced by 10-39 % compared to the basic neural network. The results of the research show the advisability of the structural tuning of the convolutional neural network for devices with a limited source of computing, namely, peripheral or mobile devices.

Keywords: Speaker identification; VGGNet; convolutional neural network; mel frequency cepstrum coefficients; structural tuning; deep learning

***For citation*: Matychenko A. D., Polyakova M. V. "The structural tuning of the convolutional neural network for speaker identification in mel frequency cepstrum coefficients space". *Herald of Advanced Information Technology*. 2023; Vol. 6 No. 2: 115–127. DOI: https://doi.org/ten.15276/hait.06.2023.7**

INTRODUCTION

The access control of various services by voice is enabled with speaker recognition systems. These systems are applied in banking over a telephone network, telephone shopping, and database access services, also as a forensics tool. Besides, a biometric authentication, voice mail, security control for confidential information and remote access to computers also are very active areas where speaker recognition has practical uses [1].

Speaker recognition is the process of automatically recognizing who is speaking by using the speech signals to verify identities being claimed by people accessing systems [2]. Speaker recognition includes such subtasks as speaker identification and speaker verification. Speaker identification determines by speech signal who spoke among a given list of speakers. Speaker verification classified whether an audio signal belongs to a predetermined speaker or not based on his/her prerecorded utterances. In fact, the latter can be seen as a particular case of the former [3].

In this article the speaker identification problem is considered. The based characteristics of the speaker identification are the recognition

performance, number of parameters and computational complexity.

Last time speaker identification systems are popularized in the devices with limited computation source, for instance, in smart phones and smart speakers for access control [4]. The storage size, processing, memory and energy consumption of applications on such devices are limited by their computing power. However, these constraints are often inconsistent with the requirements of state-of-the-art speaker identification approaches. Then there is a need for elaboration of more efficient methods, taking into account these limitations [5].

1. ANALYSIS OF RECENT RESEARCH AND PUBLICATIONS

As a result of survey, it is noted, that the statistical approaches and deep learning models are used to solve the speaker identification problem.

Statistical approaches to speaker identification are the Gaussian mixture model (GMM) [6, 7] and the universal background model (UBM) [2]. In the GMM, the audio signal is represented by a set of Gaussian distributions. Each Gaussian distribution is characterized by a mean and a covariance matrix. The GMM can process audio signals of different duration and complexity, even if some voices overlap or change over time. This model does not require a large amount of data when estimating parameters. However, the GMM does not take into account the dependencies between different elements of the audio signal and the interference, which can negatively affect on the speaker identification performance.

To address the shortcomings, after estimating the parameters of the GMM, the speaker is identified by calculating the maximum a posteriori probability with the UBM. This allows better processing of alternative speech sounds such as whispering, slow or fast speech. They may be encountered when speaker is identified. UBM requires a large amount of data for training, containing audio signals from many different speakers, to represent speech features that are independent of the individual. This model takes into account the general characteristics of the audio signals rather than the specific characteristics of each individual speaker [2, 6].

The performance of the speaker identification based on UBM is largely influenced by the speaker and channel variations of speech. To avoid this problem in [8] is proposed to reduce the high-dimensional UBM feature vectors into low-dimensional vectors, named *i*-vectors by factor analysis. The *i*-vector framework eliminates the within-speaker and channel variabilities effectively. This leads to significant performance improvement of UBM based speaker identification [2, 8].

Besides it has been proposed to replace UBM and *i*-vector classifier by deep neural network (DNN) taking into account the experience of deep learning for speech recognition [9, 10]. The DNN-based *d*-vector framework assigns the ground-truth speaker identity of a training speech signal as the labels of the training frames of this signal. Then the model training is represented as a classification problem. The frames of a training signal are classified by DNN to the speaker identity of the signal, taking Softmax as the output layer and minimizing the cross-entropy loss between the ground-truth labels of the frames and the network output [2].

Under the *d*-vector framework, in the test stage the output activation of each frame is taken from the last hidden layer of the DNN as the features of the frame. Next, the features of all frames of a signal are averaged, and a new compact representation of the speech signal, named *d*-vector, is obtained. It is shown that d-vector may generalize well to unseen speakers in the test stage [2].

According to another DNN-based framework, named *x*-vector, at first the frame-level features of speech frames are extracted by time-delay layers. Then the mean and standard deviation of the frame-level features of a signal are concatenated as a segment-level feature by a statistical pooling layer. Finally the segment-level features are classified to its speaker by a feedforward network [9, 11].

The advantage of the DNN acoustic model is its ability in modeling content-related phonetic states explicitly. This state not only generates highly compact representation of data, but also provides precise frame alignment. The DNN can model a complicated data distribution and mine the pronunciation characteristics of speakers [11]. On the contrary, GMM and UBM have no inherent meaning. They are trained by the expectation-maximization algorithm in an unsupervised manner. However, the computational complexity of a DNN over the UBM and *i*-vector framework is greatly increased [11, 12]. The training of the DNN requires a large number of labeled training data, since a DNN usually has more parameters than GMM or UBM. To overcome the computational complexity, a supervised UBM based on the DNN acoustic model is proposed. It reduces the training computational complexity of the DNN and achieves more than 20% error rate reduction over the UBM and *i*-vector framework. But training the DNN acoustic model still needs a large amount of labeled training data [12]. The using of the *d*-vector framework alone

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

yields higher error rate than the *i*-vector. The fusion of the *d*-vector and *i*-vector frameworks reduces error rate on 14 % and 25 % on clean and noisy test data respectively over the *i*-vector. With enlarged training data and data augmentation, the *x*-vector achieves significant performance improvement over the UBM/*i*-vector [2].

Based on *d*-vector and *x*-vector framework, different DNNs were proposed for the speaker identification. For example, some authors applied a residual network [13] as a backbone and modified it for specific purposes or applications [14, 15]. The residual network is a 2-dimensional convolutional neural network (CNN) with convolutions in both the time and frequency domains [13]. In this way, in [14] the number of parameters of the ResNet-34 network is reduced by cutting down the number of channels in each residual block and a network named thin ResNet is proposed.

Raw wave neural networks take raw waves in the time domain as the inputs to extract learnable acoustic features [10, 16]. It has been observed that with the use of CNN the filters of the first convolution layer capture the speaker information in low frequency regions. It is critical for the waveform-based CNNs since the first convolutional layer is more sensitive to the gradient vanishing problem, than the other layers [17, 18].

In addition, many other neural network architectures have also been applied to speech recognition, including VGGNet [19, 20]. The CNN can also be improved by inserting long short-term memory or other layers into the backbone networks [15, 16].

The Siamese neural network [21] is also used for the speaker recognition, which is a type of neural network architecture that contains two identical subnets. Both subnets have identical blocks of layers with the same parameters and weights. Parameter updates should be affected in both subnets. The similarity of the input data is detected by comparing the feature vectors. The advantages of such a neural network are better classification and small amount of data for training. The main disadvantage is increasing of the training time of such a neural network as compared to the other CNN, as it uses pairs of examples for training. Also, the training of the Siamese neural network is based on the distance between classes, not on the probability of belonging to a particular class.

2. FORMULATION OF THE PROBLEM

The analysis of the speaker recognition methods shows that their main characteristics are recognition performance, number of method parameters and training time. The high recognition performance can be achieved by using CNN. However, the number of parameters of these networks is much higher than for statistical methods, although lower than for Siamese neural networks. The large number of parameters requires a significant training set for network training, which is not always available to the researcher [4]. Besides, although the CNN have achieved a great success, the model size and inference efficiency are important for the devices with limited computation source, namely, edge or mobile devices [3, 5]. To reduce the impact of these problems the structural tuning of the available CNN is used for speaker identification.

Let CNN with the architecture S and parameter values P be preliminarily synthesized for speaker recognition: $CNN=\{S, P\}$. The set S includes layers of the synthesized network with layer parameters such as the size of the convolution or pooling kernel, and the number of convolutions. The set P contains a subset W of convolutional layer weights, and a subset B of bias values [22, 23].

The problem of structural tuning of the CNN is as follows. It is necessary to make a structural changes to the existing architecture S of the CNN. Structural tuning of the neural network involves layers adding and removing, variation of the CNN hyperparameters such as the number of layers, the number and size of layer kernels, and activation functions. These changes should reduce the number of parameters of the resulting network compared with the initial CNN. At the same time the speaker recognition performance of the resulting network should not decrease compared with the initial CNN. Result of the structural tuning of the neural network corresponds to the set $CNN_{tun}=\{S_{tun}, P_{tun}\}$. The architecture S_{tun} includes layers from the existing architecture S [24].

As a backbone in this article it is selected the CNN which designed in [25]. Based on VGGNet network architecture and *d*-vector framework, this network achieves the high recognition performance on test datasets of speech signals.

The aim of the research is the structural tuning of CNN for speaker identification to reduce the number of parameters of the network and, as a result, to reduce the network training time, provided that the speaker recognition performance is sufficient (the correct recognition is above 95 %).

3. MATERIALS AND METHODS

The design of a neural network for speaker identification includes three keys components, such as network input, network structure, and training loss function.

The network input can be categorized into raw wave signals in time domain and acoustic features in

time-frequency domain. The latter features include spectrogram, mel frequency cepstrum coefficients (MFCC), and mel-filterbank coefficients. The acoustic features also can be extracted with the techniques of spectral centroids, group delay function and integrated noise suppression [2]. In this research the MFCC are applied as input to CNN to improve the speaker recognition performance based on the *d*-vector framework.

Deep neural network, recurrent neural network, in particular, with long short-term memory, and CNN are used as base for speaker identification. The hidden layer of the neural network, for example, can be a convolutional layer [19], a layer with a long short-term memory [16], a fully connected layer and even a combination of different layers [16, 28]. Each layer of the network, topology and hyperparamers of the network can affect the speaker recognition performance. In this paper, a neural network based on the VGGNet architecture is structurally tuned.

In addition, the speaker recognition performance is significantly determined by the objective function which minimizing losses in the training process of the neural network. The authors applied Softmax as the activation function of the output layer and cross-entropy as the loss function.

Extraction of mel frequency cepstrum coefficients as speech signal features

Mel frequency cepstrum coefficients features are extracted from speech signals as follow [25, 26].

Step 1. Framing and windowing of signal.
A framing is blocked the speech signal into frames of W samples in the time domain. After framing, each pth frame is windowed with window function $w(n)$ as $y_p(n)=x(n)w(pW-n)$ where W is the window function length. The window function is used to minimize signal discontinuities at the beginning and at the end of each frame. This function integrates all closet frequency lines of each frame, making the end of each frame connect smoothly with the beginning of the next frame.

Step 2. Short-time Fourier transform (STFT) of a signal is used to convert each frame from the time domain into the frequency domain by formula $Y_p(f)=\mathrm{STFT}(y_p(n))$.

Step 3. Mapping the power spectrum above $|Y_p(f)|^2$ onto the mel scale. This transform mimics of behavior of the human ear, which acts as a filter tuning on certain frequency only. The power spectrum estimate still contains a lot of information not required for speaker recognition. In particular the human ear cannot discern the difference between two closely spaced frequencies. This effect becomes more pronounced as the frequencies increase. For this reason, the mel filterbank with transfer function $U(f)$ is multiplied on power spectrum by formula $Y_{pU}(f)=|Y_p(f)|^2U(f)$. Thus, the energies of each filter outputs are summarized to detect how much energy occurs in various frequency bands. The first filter of mel filterbank is very narrow and gives an indication of how much energy exists near 0 Hertz. As the frequencies get higher mel filters get wider as the human ear become less concerned about variations of high frequency components of speech signals.

Step 4. The logarithm of mel filter output energies is taken at each of the mel frequencies.

Step 5. Discrete cosine transform (DCT) converts the log mel spectrum into the time domain: $T_p(u)=\mathrm{DCT}(\log_{10}(Y_{pU}(f)))$. The amplitudes of the resulting spectrum obtained at this step are called MFCC [25, 26]. The MFCC represent the local spectral properties. The image of MFCC matrix is shown on Fig. 1. Each row consists of the coefficients of one frame and each column corresponds to one extracted coefficient.

The proposed network architecture

The structural tuning was performed for the convolutional neural network from [25], the architecture of which is presented in Table 1. This neural network was used to solve the problem of speaker identification from an audio file with his speech.

As a structural tuning of the network from [25], the following is proposed: (1) reducing the number of convolutional layers; (2) altering the number of convolutions in the convolutional layers and the size of the convolution kernel; (3) placing Leaky ReLU activation functions instead of ReLU after the batch normalization layers; (4) increasing the size of the kernels for the max pooling layer; (5) averaging the results of each convolution to input a two-dimensional convolution results to a fully connected layer.

In architecture of the basic and proposed networks the multi-scale feature extraction which combines features of different scales is applied. This represents the local and global information of the speech signals. The lower layers of these networks process smaller frames of speech signals and represent the specific characteristics of each individual speaker. The higher layers of these networks process larger frames of speech signals and have ability to characterize general features of the audio signals. In the proposed network architecture scale values are changed to larger one compared with the basic network. This allows to reduce the number of parameters of the network and the network training time, provided the high speaker recognition performance.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

Table 1. **The based network architecture**

Layer number	Type	Comment	Activations	Learnables
1	Image input	H×W×1 with zero center normalization	H×W×1	–
2	Convolution	8 3×3 convolutions with stride 1 and same padding	H×W×8	Weights: 3×3×1×8 Bias: 1×1×8
3	Batch normalization	Batch normalization with 8 channels	H×W×8	Offset: 1×1×8 Scale: 1×1×8
4	ReLU	Activation function	H×W×8	–
5	Max pooling	2×2 max pooling with stride 2 and zero padding	H/2×W/2×8	–
6	Convolution	16 3×3×8 convolutions with stride 1 and same padding	H/2×W/2×16	Weights: 3×3×8×16 Bias: 1×1×16
7	Batch normalization	Batch normalization with 16 channels	H/2×W/2×16	Offset: 1×1×16 Scale: 1×1×16
8	ReLU	Activation function	H/2×W/2×16	–
9	Max pooling	2×2 max pooling with stride 2 and zero padding	H/4×W/4×16	–
10	Convolution	32 3×3×16 convolutions with stride 1 and same padding	H/4×W/4×32	Weights: 3×3×16×32 Bias: 1×1×32
11	Batch normalization	Batch normalization with 32 channels	H/4×W/4×32	Offset: 1×1×32 Scale: 1×1×32
12	ReLU	Activation function	H/4×W/4×32	–
13	Fully connected layer	Reshapes the output of previous layer to 1d-array of elements	1×1× × (H/4×W/4×32)	
14	Softmax	Activation function	1×1×*k*	–
15	Classification output	Class weighted cross-entropy loss with number of classes corresponding to the number of speakers	–	–

Source: **compiled by the [25]**

The neural network proposed as a result of the structural tuning has fewer layers than the neural network architecture from [25]. The number of filters and the size of kernels in the convolutional layers were altered. In the basic neural network [25], convolutional layers with the number of convolutions from 8 to 32, but with a constant convolutional kernel size of 3 pixels, were used. In this paper, it is proposed to use convolutional layers with a number of convolutions 20 and 16 and with different convolution kernel sizes – first 8 and then 3 pixels.

In [25], the activation function ReLU was used, which is described by the formula:

$$\mathrm{ReLU}(x)=\max(0, x),$$

and takes the value x for positive values of the argument, and the value 0 for other values of the argument.

For the proposed neural network, instead of the ReLU activation function, the related Leaky ReLU function with the parameter a=0.1 is used:

$$\mathrm{LeakyReLU}(x)=\max(0, x) – a \min(0, x),$$

which takes the value of x for positive values of the argument and the value $(–ax)$ for other values of the argument.

The ReLU and Leaky ReLU activation functions are computationally efficient and do not suffer from the vanishing gradient problem when the gradient of the activation function become very small for large or small input values. Then it is difficult to train the neural network effectively when using the sigmoid or hyperbolic tangent.

By introducing a small slope for negative values of x, Leaky ReLU ensures that all neurons in the network can contribute to the output, even if their inputs are negative. Leaky ReLU takes into consideration the negative inputs, but diminishes the impact they have on the output.

When the processed speech signals are noised the Leaky ReLU, in contrast to the ReLU, can help to avoid discarding potentially important information. Introducing some noise into the neuron outputs, Leaky ReLU reduces the probability of overfitting and improves generalization performance. Therefore, it is preferred when generalization performance is a priority.

Also, as a result of structural tuning, the size of the kernels for the max pooling layer was increased. It is proposed the averaging of the results of each convolution to input a two-dimensional convolution results to a fully connected layer with the Softmax activation function.

The architecture of the proposed neural network for the speaker identification is shown in Table 2. This network is designed to process a one-channel image with a size of H×W pixels. The first convolutional layer uses 20 convolution kernels of size 8×8×1 pixels with a stride of 1 pixel. The feature maps at the output of this layer are batch normalized and then the Leaky ReLU activation function with a=0.1 is applied.

Next, feature maps from the output of the Leaky ReLU activation function layer, were inputted to the 3×3 max pooling with stride of 3 pixels. Then, feature maps from the output of the pooling layer were inputted to the second convolutional layer with 16 convolution kernels of size 3×3×20 pixels and a stride of 1 pixel. The feature maps at the output of this layer are batch normalized and then the Leaky ReLU activation function with a=0.1 is applied.

Finally, the obtained feature maps were inputted to the H/3×W/3 average pooling without stride. The resulting feature maps from the output of the pooling layer were inputted to the Softmax activation function layer and to the classification layer, where k is the number of speakers.

The Adam method with an initial learning rate of 0.005 was applied to train the proposed CNN. When training the proposed CNN, a cross-entropy loss function was used, for which the relative frequencies of speaker audio files is w_i, i=1,…, k.

The value of cross-entropy L was calculated by the formula

$$L = -\sum_{i=1}^{k} w_i t_i \log_2 y_i,$$

where t_i = 1 if the spectrogram is assigned to audio file of i-th speaker, otherwise t_i = 0. The y_i value is the result of calculating the value of the Softmax function for the spectrogram of audio file of i-th speaker. It is interpreted as the probability that the spectrogram of audio file corresponds to i-th speaker, i=1, ..., k.

Table 2. **The proposed network architecture**

Layer number	Type	Comment	Activations	Learnables
1	Image input	H×W×1 with zero center normalization	H×W×1	–
2	Convolution	20 8×8 convolutions with stride 1 and same padding	H×W×20	Weights: 8×8×1×20 Bias: 1×1×20
3	Batch normalization	Batch normalization with 32 channels	H×W×20	Offset: 1×1×20 Scale: 1×1×20
4	Leaky ReLU	Activation function with a=0.1	H×W×20	–
5	Max pooling	3×3 max pooling with stride 3 and zero padding	H/3×W/3×20	–
6	Convolution	16 3×3×20 convolutions with stride 1and same padding	H/3×W/3×16	Weights: 3×3×20×16 Bias: 1×1×16
7	Batch normalization	Batch normalization with 32 channels	H/3×W/3×16	Offset: 1×1×16 Scale: 1×1×16
8	Leaky ReLU	Activation function with a=0.1	H/3×W/3×16	–
9	Average pooling	H/3×W/3 average pooling with stride 0 and zero padding	1×1×16	–
10	Softmax	Activation function	1×1×k	–
11	Classification output	Class weighted cross-entropy loss with number of classes corresponding to the number of speakers	–	–

Source: **compiled by the authors**

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

4. EXPERIMENTAL RESEARCH OF THE PROPOSED NETWORK FOR SPEAKER IDENTIFICATION

The results of the classification of audio recordings by the proposed CNN and CNN from [25] in comparison with the ground-truth labels were estimated by confusion matrices. The element of the confusion matrix n_{ij} located in the *i*-th row and the *j*-th column, shows how many audio recordings from the class with the *i*-th label are assigned to the class with the *j*-th label. The sum of the elements of each row of the confusion matrix is the same as the number of audio recordings from the class with the *i*-th label.

Based on the elements of the confusion matrix, we estimated the precision *Pr*, recall *Rc* and F-score *F* for each class of audio recordings C_i, i.e. for each speaker, using the following formulas:

$$Pr(C_i)=n_{ii}/(n_{1i}+n_{2i}+...+n_{ki}),$$
$$Rc(C_i)= n_{ii}/(n_{i1}+n_{i2}+...+n_{ik}),$$
$$F(C_i)=2\times Pr(C_i)\times Rc(C_i)/(Pr(C_i)+Rc(C_i)),$$

where k is the number of classes of audio recordings, i.e. the number of speakers.

During the experiment, five datasets were used, four of which were taken from the Kaggle website [27, 28]. The bit depth of the files used in the experiment ranged from 16 bits to 32 bits. The difference in audio quality between 16 and 32 bits is extremely large, because 32-bit audio is used for professional sound recording, while 16-bit audio corresponds to CD playback, streaming of maximum quality. Sampling frequency ranged from 16 kHz to 48 kHz. The range is not very large, but the frequency in range from 44.1 kHz to 48 kHz is common sampling frequency, and the frequency from 16 kHz to 22.05 kHz is typical for low-bandwidth audio streaming.

Most files (above 91 %) are stereo with two separated channels [29]. This is not a problem, since channel mixing is a very common technique in digital audio processing. When implementing software in Python the stereo channels are mixed into a mono mix and software worked with single channel audio.

The first dataset was obtained from two speakers with a total number of 296 files, 294 of which are one second long. All files have two channels, a bit depth of 16 bits and a sampling frequency of 48 kHz. The first speaker has 140 files, and the second speaker has 156 files. Out of this dataset, 236 files were used for training of the proposed neural network and 60 files were used for testing. The second dataset consists of 3301 files from two speakers with duration of almost one second. All files have one channel, 16-bit depth and 16 kHz sampling frequency. The first speaker corresponds to 1500 files, and the second speaker to 1801 files. Out of this dataset, 2640 files were used for training of the proposed neural network, and 661 files were used for testing.

For each file of the first and second datasets, a spectrogram of MFCC of 20 × 44 pixels was obtained (Fig. 1a).

The third dataset consists of 7501 one-second files obtained from five speakers. Each speaker had almost the same number of files; each file had one channel, a bit depth of 16 bits and a sampling frequency of 16 kHz. Out of this dataset, 6000 files were used for training of the proposed neural network and 1501 files were used for testing.

The fourth dataset was obtained from 20 speakers. The total number of files is 9237, most of which are two seconds long. Most files had two channels, a bit depth of 32 bits and a sampling frequency of 16 kHz. Out of this dataset, 7389 files were used for training of the proposed neural network and 1848 files were used for testing.

For each file of the third and fourth datasets, a spectrogram of MFCC of 40 × 82 pixels was obtained (Fig. 1b).

The fifth dataset is a composite dataset of the four previous ones, with 29 speakers and a total of 20335 files. Due to the large amount of data, the spectrogram of the MFCC was enlarged to 40 × 164 pixels. Out of this dataset, 16327 files were used for training the proposed neural network, and 4008 files were used for testing.

Thus, most of the files are between one and three seconds long, which means that the speaker may have time to say only a couple of words. Since the audio files are featured using spectrograms of MFCC, files with longer duration will increase the number of features and the time for training and generalization of the neural network.

At the first stage of the experiment, the precision, recall, and F-score of speaker recognition by the proposed and basic neural networks were evaluated for the testing sets from five datasets (Table 3). For each speaker, the value of the index obtained with the proposed neural network is given, and the value of the same index obtained with the basic neural network is given in brackets. The last column of this table shows the number of files corresponding to each speaker.

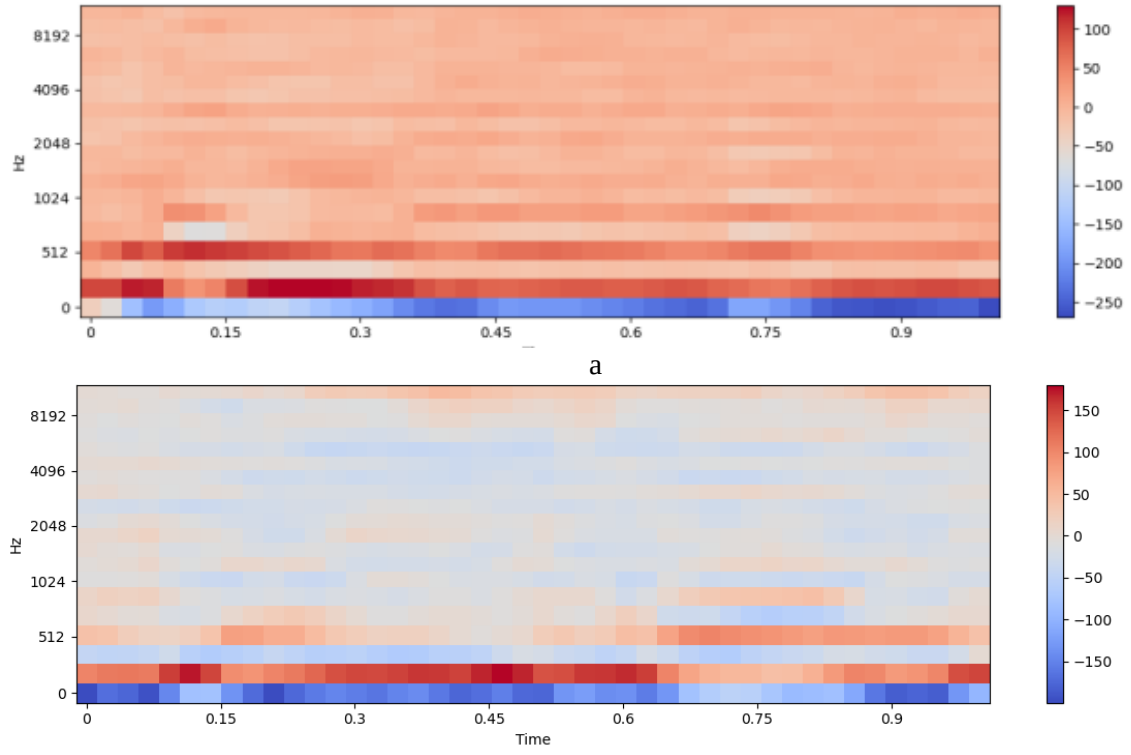

Fig. 1. **Spectrograms of the MFCC for one of the files: a – the second dataset; b – the third dataset**

Source: **compiled by the authors**

The analysis of values in Table 3 shows that the speaker recognition performance with the basic and the proposed neural networks is quite high and differs within the statistical error of the calculation of the indexes.

At the second stage of the experiment, the confusion matrices were calculated for each of the five researched datasets, provided that the speakers were recognized by the proposed and basic CNNs. The analysis of the obtained confusion matrices revealed the following. For the first dataset, one audio file corresponding to the Bimal was misclassified by both the proposed and basic networks. For the second dataset, the two audio files corresponding to the Navid were misclassified by the proposed network. The basic CNN made a similar mistake for only one audio file. For the third dataset, five audio files corresponding to the speaker Jens Stoltenberg were classified by the proposed network incorrectly: 2 audio files were classified as corresponding to Benjamin Natanjau and 3 audio files were classified as corresponding to Julia Gillard. The basic CNN misclassified 10 audio files.

For the fourth dataset, only one audio file was misclassified by the proposed and basic neural networks. Finally, for the fifth dataset, 27 and 23 audio files were misclassified by the proposed and basic network respectively.

At the third stage of the experiment, the number of parameters of the basic and proposed neural networks was calculated. The basic CNN with the architecture in Table 1 contains

$$3\times3\times8+8+8+8+3\times3\times8\times16+16+16+16+3\times3\times\times16\times32+32+32+32=6000$$ parameters.

The proposed CNN contains

$$8\times8\times20+20+20+20+3\times3\times20\times16+16+16+16=$$
$$=4268$$ parameters (Table 2).

Thus, the number of parameters of the proposed CNN is 29 % less than the number of parameters of the basic CNN, provided that the speaker recognition performance is similar.

At the last stage of the experiment, we compared the training time for the proposed and basic CNNs. The research was performed using an Intel(R) Core(TM) i7-10750H processor, 2.60GHz CPU, 16GB memory, Windows 11 operating system, 64 bit. It was observed that the training time depends on the datasets consisting of different numbers of files. The number of epochs for training on each dataset was chosen to be 250. Compared to the basic CNN, the training time of the proposed CNN was reduced for the first, second, third, fourth, and fifth datasets by 18 %, 39 %, 21 %, 10 %, and 20 %, respectively (Table 4).

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

Table 3. **The results of speaker recognition on five datasets**

	Precision	Recall	F1-score	Support
The first dataset				
Bimal	1.00 (1.00)	0.94 (0.97)	0.97 (0.98)	31
Mihir	0.94 (0.97)	1.00 (1.00)	0.97 (0.98)	29
The second dataset				
Navid	1.00 (1.00)	0.97 (1.00)	1.00 (0.98)	378
Zohreh	0.96 (1.00)	1.00 (1.00)	0.98 (1.00)	283
The third dataset				
Benjamin Netanyahu	0.97 (0.97)	1.00 (1.00)	0.98 (0.99)	310
Jens Stoltenberg	1.00 (1.00)	0.98 (0.99)	0.99 (0.99)	310
Julia Gillard	0.99 (1.00)	0.97 (0.98)	0.98 (0.99)	283
Magaret Tarcher	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	283
Nelson Mandela	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	315
The forth dataset				
Dang Anh Tu	1.00 (1.00)	0.99 (1.00)	0.99 (1.00)	87
Dinh Dong Thuc	0.99 (1.00)	1.00 (1.00)	0.99 (1.00)	97
Dinh Tien Anh	1.00 (1.00)	0.99 (1.00)	1.00 (1.00)	91
Hoang Ngoc Anh Trung	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	103
Hoang Van Phuong	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	77
Lac Minh Long	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	43
Le Minh Phat	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	90
Le Thanh Dat	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	92
Le Trong Phuc	1.00 (1.00)	1.00 (0.99)	1.00 (0.99)	83
Mai Nguyen Thai Hoc	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	98
Mai Van Hiep	1.00 (0.99)	1.00 (1.00)	1.00 (0.99)	97
Nguyen Hoai Bao	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	110
Nguyen Hung	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	112
Nguyen Hung Hoai Nam	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	106
Nguyen Huu Huy	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	107
Nguyen Minh Khanh	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	101
Nguyen Phu Khanh	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	95
Nguyen Quang Khai	0.99 (1.00)	1.00 (1.00)	0.99 (1.00)	85
Nguyen Quoc Son	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	80
Nguyen Son Dinh	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	94
The fifth dataset				
Benjamin Netanyahu	0.98 (0.99)	0.99 (0.99)	0.98 (1.00)	277
Dang Anh Tu	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	60
Dinh Dong Thuc	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	108
Dinh Tien Anh	1.00 (1.00)	0.99 (1.00)	0.99 (1.00)	99
Hoang Ngoc Anh Trung	0.99 (0.98)	1.00 (1.00)	1.00 (1.00)	101
Hoang Van Phuong	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	95
Jens Stoltenberg	0.99 (0.99)	0.99 (0.99)	0.99 (1.00)	308
Julia Gillard	0.98 (0.99)	0.98 (1.00)	0.98 (1.00)	315
Lac Minh Long	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	44
Le Minh Phat	1.00 (1.00)	0.99 (1.00)	0.99 (1.00)	86
Le Thanh Dat	1.00 (1.00)	1.00 (1.00)	1.00 (0.99)	109
Le Trong Phuc	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	91
Margaret Thatcher	0.99 (0.99)	0.99 (0.99)	0.99 (1.00)	308
Mai Nguyen Thai Hoc	1.00 (1.00)	0.99 (0.98)	0.99 (1.00)	99
Mai Van Hiep	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	99
Navid	1.00 (0.99)	0.98 (0.98)	0.99 (1.00)	346
Nelson Mandela	1.00 (0.99)	1.00 (1.00)	1.00 (1.00)	308

Table 3 (continued)

	Precision	Recall	F1-score	Support
Nguyen Hoai Bao	1.00 (0.99)	0.99 (1.00)	1.00 (1.00)	102
Nguyen Hung	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	111
Nguyen Hung Hoai Nam	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	118
Nguyen Huu Huy	0.99 (1.00)	1.00 (1.00)	0.99 (1.00)	96
Nguyen Minh Khanh	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	102
Nguyen Phu Khanh	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	94
Nguyen Quang Khai	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	88
Nguyen Quoc Son	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	82
Nguyen Son Dinh	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	92
Zohreh	0.98 (0.99)	1.00 (0.99)	0.99 (1.00)	270

Source: **compiled by the authors**

Table 4. **Training time of the proposed and basic neural networks**

Dataset	1	2	3	4	5
Training time of the basic CNN, s	20	167	319	1500	3060
Training time of the proposed CNN, s	17	120	264	1368	2540

Source: **compiled by the authors**

CONCLUSIONS

The problem of the speaker identification for the devices with limited computation source, edge or mobile devices is solved.

The scientific novelty of obtained results is the further developing of the speaker identification method using a CNN in the space of MFCC. For this method the number of network parameters and the network training time are reduced provided that the correct speaker recognition is above 95 %. The applying of the proposed CNN will reduce the resource consumption for speaker identification systems for devices with limited computing power.

The practical significance of obtained results is that the software realizing the proposed CNN is developed, as well as experiment to research speaker recognition performance is done. The experimental results allow to recommend the proposed CNN for use in practice, as well as to determine effective conditions for its application.

Prospects for further research are to automated design of hardware and software aimed at the implementation of CNN in announcer identification systems [30].

REFERENCES

1. Singh, N., Khan, R. A. & Shree, R. “Application of speaker recognition”. *Procedia Engineering*. 2012; 38: 3122–3126. DOI: https://doi.org/10.1016/J.PROENG.2012.06.363.

2. Bai, Z. & Zhang X.-L. “Speaker recognition based on deep learning: An overview”. *Neural Networks*. 2021, 140: 65–99. DOI: https://doi.org/10.1016/j.neunet.2021.03.004.

3. Nunes, J. A. C., Macêdo, D. & Zanchettin, C. “Am-mobilenet1d: A portable model for speaker recognition”. *International Joint Conference on Neural Networks (IJCNN)*. 2020. p. 1–8. DOI: http://dx.doi.org/10.1109/ IJCNN48605.2020.92075192020.

4. Georges, M., Huang, J. & Bocklet, T. “Compact speaker embedding: lrxvector”. *Interspeech Conference*. 2020. p. 3236–3240. DOI: http://dx.doi.org/10.21437/Interspeech.2020-2106.

5. Safari, P., India, M. & Hernando, J. “Self-attention encoding and pooling for speaker recognition”. *Interspeech Conference*. 2020. p. 941–945. DOI: http://dx.doi.org/10.21437/Interspeech.2020-1446.

6. Vyas, M. “A Gaussian mixture model based speech recognition system using MATLAB”. *Signal & Image Processing: An International Journal (SIPIJ)*. 2013; 4 (4): 109–118. DOI: http://doi.org/10.5121/sipij.2013.4409.

7. Vintsuk, T. K., Sazhok, M. M., Selukh, R. A., Fedorin, D. Ya., Yukhimenko, O. A. & Robeyko, V. V. “Automatic recognition, understanding and synthesis of speech signals in Ukraine”. *Control Systems and Computers.* 2018; 6 (278): 7–24. DOI: https://doi.org/10.15407/usim.2018.06.007.

8. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P. & Ouellet, P. “Front-end factor analysis for speaker verification”. *IEEE Transactions on Audio Speech and Language Processing*. 2011; 19 (4): 788–798. DOI: https://doi.org/10.1109/TASL.2010.2064307.

9. Mamyrbayev, O., Kydyrbekova, A., Alimhan, K., Oralbekova, D., Zhumazhanov, B. & Nuranbayeva, B. "Development of security systems using DNN and i & x-vector classifiers". *Eastern-European Journal of Enterprise Technologies*. 2021; 4 (9): 32–45. DOI: https://doi.org/10.15587/1729-4061.2021.239186.

10. Mamyrbayev, O., Alimhan, K., Oralbekova, D., Bekarystankyzy, A. & Zhumazhanov, B. "Identifying the influence of transfer learning method in developing an end-to-end automatic speech recognition system with a low data level". *Eastern-European Journal of Enterprise Technologies*. 2022; 1 (9): 84–92. DOI: https://doi.org/10.15587/1729-4061.2022.252801.

11. Snyder, D., Garcia-Romero, D., Povey, D. & Khudanpur, S. "Deep neural network embeddings for text-independent speaker verification". *Interspeech Conference*. 2017. p. 999–1003. DOI: https://doi.org/10.21437/Interspeech.2017-620.

12. Snyder, D., Garcia-Romero, D., & Povey, D. "Time delay deep neural network-based universal background models for speaker recognition". *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 2015. p. 92–97. DOI: https://doi.org/10.1109/ASRU.2015.7404779.

13. Wang, Z., Yao, K., Li, X. & Fang, S. "Multi-resolution multi-head attention in deep speaker embedding". *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020. p. 6464–6468. DOI: https://doi.org/10.1109/ICASSP40776.2020.9053217.

14. Xie, W., Nagrani, A., Chung, J. S. & Zisserman, A. "Utterance-level aggregation for speaker recognition in the wild". *IEEE International Conference on Acoustics, Speech and Signal* Processing (ICASSP). 2019. p. 5791–5795. DOI: https://doi.org/10.1109/ICASSP.2019.8683120.

15. Zhao, Y., Zhou, T., Chen, Z. & Wu, J. "Improving deep CNN networks with long temporal context for text-independent speaker verification". *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2020. p. 6834–6838. DOI: https://doi.org/10.1109/ICASSP40776.2020.9053767.

16. Jung, J.-W., Heo, H.-S., Yang, I.-H., Shim, H.-J. & Yu, H.-J. "Avoiding speaker overfitting in end-to-end DNNs using raw waveform for textindependent speaker verification". *Interspeech Conference*. 2018. p. 3583–3587. DOI: https://doi.org/10.21437/Interspeech.2018-1608.

17. Muckenhirn, H., Doss, M. M. & Marcell, S. "Towards directly modeling raw speech signal for speaker verification using CNNs". *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2018. p. 4884–4888. DOI: https://doi.org/10.1109/ICASSP.2018.8462165.

18. Ravanelli, M. & Bengio, Y. "Speaker recognition from raw waveform with SincNet". *IEEE Spoken Language Technology Workshop*. 2018. p. 1021–1028. DOI: https://doi.org/10.1109/SLT.2018.8639585.

19. Bhattacharya, G., Alam, M. J. & Kenny, P. "Deep speaker embeddings for short-duration speaker verification". *Interspeech Conference*. 2017. p. 1517–1521. DOI: https://doi.org/10.21437/Interspeech.2017-1575.

20. Yadav, S. & Rai, A. "Learning discriminative features for speaker identification and verification". *Interspeech Conference*. 2018. p. 2237–2241. DOI: https://doi.org/10.21437/Interspeech.2018-1015.

21. Liu, X., Sahidullah, M. & Kinnunen, T. "A comparative re-assessment of feature extractors for deep speaker. *Interspeech Conference*. 2020. p. 3221–3225. DOI: https://doi.org/10.21437/INTERSPEECH.2020-1765.

22. Leoshchenko, S. D., Oliynyk, A. O. , Subbotin, S. O., Hoffman, E. O. & Kornienko, O. V. "Method of structural adjustment of neural network models to ensure interpretability". *Radio electronics, Computer science, Control*. 2021; 3: 86–96. DOI: https://doi.org/10.15588/1607-3274-2021-3-8.

23. Polyakova, M. V. "Image segmentation with a convolutional neural network without pooling layers in dermatological disease diagnostics systems". *Radio Electronics, Computer Science, Control.* 2023; 1: 51–61. DOI: http://doi.org/ 10.15588/1607-3274-2023-1-5.

24. Lovkin, V. M., Subbotin, S. A., Oliinyk, A. O. & Myronenko, N. V. "Method and software component model for skin disease diagnosis ". *Radio Electronics, Computer Science, Control*. 2023; 1: 40–50. DOI: http://doi.org/ 10.15588/1607-3274-2023-1-4.

25. Bunrit, S., Inkian, T., Kerdprasop, N. & Kerdprasop, K. "Text-independent speaker identification using deep learning model of convolution neural network". *International Journal of Machine Learning and Computing*. 2019; 9 (2): 143–148. DOI: https://doi.org/10.18178/ijmlc.2019.9.2.778.

26. Liu, J-C., Leu, F-Y., Lin, G-L. & Susanto, H. “An MFCC-based text-independent speaker identification system for access control”. *Concurrency and Computation: Practice and* Experience. 2018; 30 (2): e4255. DOI: https://doi.org/10.1002/cpe.4255.

27. “Kaggle Speaker Identification”. – Available from: https://www.kaggle.com/code/auishikpyne/speaker-identification/data. – [Accessed: April 2022].

28. “Kaggle Speaker Recognition without noise”. – Available from: https://www.kaggle.com/datasets/rsm2213839/speaker-recognition. – [Accessed: April 2022].

29. “Kaggle Speaker Recognition”. – Available from: https://www.kaggle.com/datasets/phmanhth/speaker-recognitiona. – [Accessed: April 2022].

30. Tsmots, I. G., Berezsky, O. M. & Berezkyy, M. O. “Methods and hardware to accelerate the work of a convolutional neural network”. *Applied Aspects of Information Technology*. 2023; 6 (1): 13–27. DOI: https://doi.org/10.15276/aait.06.2023.1.

Conflicts of Interest:the authors declare no conflict of interest

Received 24.03.2023
Received after revision 06.06.2023
Accepted 19.06.2023

DOI: https://doi.org/10.15276/hait.06.2023.7
УДК 004.93

Структурне налаштування згорткової нейронної мережі для ідентифікації дикторів у просторі мелчастотних кепстральних коефіцієнтів

Матиченко Анастасія Денисівна[1)]
ORCID: https://orcid.org/0009-0009-7894-4734; matychenko.8089532@stud.op.edu.ua
Полякова Марина Вячеславівна[1)]
ORCID: https://orcid.org/0000-0001-7229-7657; marinapolyakova943@gmail.com. Scopus Author ID: 57017879200
[1)] Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Аналіз літератури дозволив виділити основні методи ідентифікації дикторів за мовними сигналами. Це статистичні методи на основі моделі суміші гаусівських розподілів та універсальної фонової моделі, також нейромережеві методи, зокрема із використанням згорткових або сіамських нейронних мереж. Основними характеристиками цих методів є якість розпізнавання, кількість параметрів і час навчання. Високої якості розпізнавання дозволяє досягти застосування згорткових нейронних мереж, однак кількість параметрів цих мереж значно вища, ніж для статистичних методів, хоча і нижча, ніж для сіамських нейронних мереж. Значна кількість параметрів вимагає великої навчальної вибірки для навчання мережі, яка не завжди є в розпорядженні дослідника. Крім того, незважаючи на ефективність згорткових нейронних мереж, розмір моделі та ефективність виведення остаються важливими для пристроїв з обмеженим джерелом обчислень, а саме периферійних або мобільних пристроїв. Тому аспекти налаштування структури існуючих згорткових нейронних мереж є актуальними для дослідження. У роботі проведено структурне налаштування існуючої згорткової нейронної мережі на основі архітектури VGGNet для ідентифікації дикторів у просторі мелчастотних кепстральних коефіцієнтів. Метою роботи було зменшення кількості параметрів нейронної мережі і, як наслідок, скорочення часу навчання мережі за умови достатньої якості розпізнавання (правильне розпізнавання вище за 95 %). Запропонована у результаті структурного налаштування нейронна мережа має менше шарів, ніж архітектура базової нейронної мережі. Замість функції активації ReLU застосовано спориднену до цієї функції функцію Leaky ReLU з параметром 0.1. Змінено кількість фільтрів та розмірність ядер в згорткових шарах. Збільшено розмірність ядер для пулінгового шару з обранням максимального елементу. Запропоновано використання усереднення результатів кожної згортки для переходу від двовимірної згортки до повнозвязного шару з функцією активації Softmax. Експеримент показав, що кількість параметрів запропонованої нейронної мережі менша на 29 % кількості параметрів базової нейронної мережі за умови майже однакової якості розпізнавання дикторів. Окрім того, на п'яти датасетах аудіозаписів, що відповідали різної кількості дикторів, оцінювався час навчання запропонованої та базової нейронної мережі. Було отримано скорочення часу навчання запропонованою мережею на 10-39 % у порівнянні з базовою

нейронною мережею. Результати дослідження показують доцільність застосування структурного налаштування згорткової нейронної мережі для пристроїв з обмеженим джерелом обчислень, а саме периферійних або мобільних пристроїв.

Ключові слова: Ідентифікація дикторів; VGGNet; згорткова нейронна мережа; мелчастотні кепстральні коефіцієнти; структурне налаштування; глибоке навчання

ABOUT THE AUTHORS

Anastasiia D. Matychenko - Bachelor, Department of Applied Mathematics and Information Technology. Odessa Polytecnic National University, 1, Shevchenko Ave. Odessa, 65044, Ukraine.
ORCID: https://orcid.org/0009-0009-7894-4734; matychenko.8089532@stud.op.edu.ua
***Research field*:** Intelligent data analysis; machine learning; digital image processing

Матиченко Анастасія Денисівна - бакалавр кафедри Прикладної математики та інформаційних технологій. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Marina V. Polyakova - Doctor of Engineering Sciences, Associated Professor, Professor of Department of Applied Mathematics and Information Technology. Odessa Polytecnic National University, 1, Shevchenko Ave. Odessa, 65044, Ukraine.
ORCID: https://orcid.org/0000-0001-7229-7657; marinapolyakova943@gmail.com. Scopus Author ID: 57017879200
***Research field*:** Intelligent data analysis; machine learning; digital image processing

Полякова Марина Вячеславівна - доктор технічних наук, доцент, професор кафедри Прикладної математики та інформаційних технологій. Національний університет «Одеська політехніка», проспект Шевченка,1. Одеса, 65044, Україна