

SPEAKER MODELING BY PREPROCESSING SPEECH SIGNALS

¹Tashev K.A., ²Fayzieva D.S., ³Yuldasheva N.S.

¹Candidate of technical sciences, associated professor, Tashkent university of information technologies

²PhD, Associated Professor, Tashkent university of information technologies

³Doctoral student, Tashkent university of information technologies

<https://doi.org/10.5281/zenodo.10031398>

Abstract. *This article examines the state of the art of speech activity boundaries, speech signal formation and processing technologies, and considers the attributes necessary for the speaker model.*

Keywords: *VAD algorithm, identification, verification, speaker model.*

Currently, it is necessary to provide consumers with sufficient information security, to effectively provide modern information communication technologies and services. One such service is recognition of a person based on speech signals. The effectiveness of recognition is highly dependent on the effectiveness of the initial digital processing of speech signals. Taking into account the ever-increasing requirements for speech signal processing devices, i.e. improving the quality of speech and increasing its transmission speed, compacting and reducing costs, standardization of equipment, work on improving these types of programs and devices is continued. needs to be done.

Preprocessing is done to prepare the signal symbols for the extraction stage. The signal can be affected by noise caused by internal organs, body or hand movements, and impulse noise. The initial processing is carried out in the following stages:

1. Low-pass filter. In this, unnecessary high-frequency noise is removed, that is, an elliptical low-pass filter with a frequency of 300 Hz is used.
2. Noise removal. In this case, values above a certain threshold are equalized to minimize the effect of burst noise.
3. Amplitude normalization. In this case, all signals are brought between -1 and +1.

Determining the boundaries of speech activity. In voice communication, the space where the speech information is located in the signal is called the speech active area. Intervals that do not contain speech information, regardless of the presence or absence of background noise, are called intermediate stops.

Algorithm for determining speech activity time limits in the literature VAD (Voice Activity Detection) is called. The VAD algorithm allows dividing the signal into two segments: A (activity) and S (silence).

One of the examples of the application of the VAD algorithm is the GSM (global system of mobile communication) standard cellular radio communication system [2]. Speech processing according to the principle of DTX continuous transmission allows you to turn on the transmitter during the active speech of the user and turn it off during a pause or at the end of the conversation. The DTX is driven by a VAD detector, which is used to detect and separate speech and non-speech intervals even when the SHM is below 0dB. In GSM systems, VAD plays a crucial role in reducing consumption, since the average speech activity of the speaker in a monologue is below 50%, and

in a dialogue, the participant's activity can decrease up to 30% of the conversation time [2]. The VAD algorithm consumes a lot of resources in terms of machine time, while the Rabiner-Sambur algorithm [4] is more economical.

The advantages of the algorithm are its simplicity and sufficient accuracy in determining the limits of speech activity even in low-energy sounds. There is a standardized VAD algorithm as part of the speech codec known as ITU Standard – G.729 [5]. The G.729 VAD algorithm uses four features of the phonogram to determine the limits of speech activity: full-range energy, low-frequency energy, zero-crossing frequency, and signal spectrum coefficients. The differences between each of these features and their average values are calculated. These averages are updated at break points. The decision about the presence or absence of speech is made by a multidimensional vector of parameters. The G.729 algorithm demonstrates better performance than VAD used in Half Rate GSM (coding rate not higher than 4 kbit/s). In addition, it is proposed to improve the G.729 algorithm, that is, to use the TE-LPC algorithm instead of the standard linear prediction algorithm to extract the spectrum envelope. The algorithm is based on the range-limited interpolation of the calculated signal spectrum [3].

The VAD algorithm with separate criteria for selecting active and inactive speech was proposed in [6]. VAD algorithms become more complicated when there is significant background noise, so more sophisticated approaches are used in such situations. In [6], a VAD algorithm with an adaptive threshold is proposed, which shows good results even in time-varying signal-to-noise ratio. Most of the algorithms for extracting the active part of speech are commercial and closed. The main problems of existing VAD-algorithms are unreliable performance in background interference and high computational resources. Another drawback is that speech is treated as louder than interference, and this is not always true. Another drawback is the need for careful experimental selection of scaling factors. In order to eliminate the mentioned shortcomings, in this work, an approach based on the analysis of the distribution of local extremums is proposed

In addition to the above methods, suitable segments can also be used to distinguish certain types of sounds by their characteristics. As such signs, you can get signal oscillation speed, average power and duration. One of the classification parameters is the signal's zero crossing frequency (ZCR). Unvoweled parts of the speech signal vibrate faster. Therefore, the value of ZCR is much higher, which indicates that this parameter can be used in "tone / not tone" segmentation [6,7]. From the energetic point of view, a simple but effective parameter for distinguishing vowels is average power, relatively large values are obtained for vowels [1]. To make the segmentation algorithm work more effectively, it is recommended to consider the average power of the segments in a certain frequency range. And the pronunciation is separated by a short duration compared to the explosive sound, sliding and vowel sounds due to the sharp opening of the barrier in the speech apparatus.

Characterization of speech signals The effectiveness of recognition systems depends on how the characters are selected. The better the initial character space is chosen, the higher the recognition quality. The problem of recognizing a person based on a point also begins with the formation of a character vector from the point signal.

A person expresses his feelings, thoughts, views, and ideas through oral speech. The process of speech production includes articulation, speech and fluency [8,9]. It is a set of natural human movement abilities, and in adults, the task of producing about 14 different sounds per second is classified through the coordinated movement of about 100 muscles connected to the

spinal cord and cranial nerves. The ease with which people speak differs from the complexity of the task, and this complexity may help explain why speech may be highly susceptible to neurodegenerative disorders [10].

Person recognition is the ability of software or hardware to receive a speech signal, identify the speaker present in the speech signal, and then recognize him [11]. Speaker recognition performs a task similar to that of the human brain. It starts with speech, which is an introduction to speaker recognition. Typically, the process of speaker recognition occurs in three main steps: acoustic processing, feature extraction, and classification or recognition [12].

Speech signal processing is necessary to remove important speech attributes [13] and noise before identification. The goal of character extraction is to describe the speech signal with a predetermined number of signal components [14,15].

Character extraction is performed by transforming the speech waveform into a parametric representation with a relatively lower data rate for further processing and analysis. This is commonly referred to as external interface signal processing [16,17]. It transforms the processed speech signal into a compact but logical image, which is more original and reliable than the real signal. Since the initial element is the starting element of the sequence, the quality of subsequent functions depends significantly on the quality of the external interface [17]. Therefore, the optimal classification comes from perfect and quality features. In modern automatic systems of person recognition (ASR), the process of character extraction usually consists of finding a relatively reliable pattern for several conditions of the same speech signal, even if the environmental or dynamic conditions change, while preserving the descriptive part of the information in the speech signal. consists of [14,15].

Character extraction approaches typically provide a multidimensional feature vector for each speech signal [18]. There are a wide range of parameters to parametrically represent the speech signal for the recognition process, such as perceptual linear prediction (PLP), linear predictive coding (LPC), and mel frequency cepstral coefficients (MFCC). MFCC is the most popular and widely used [16,19]. Character extraction is the most important part of speaker recognition. Speech features play an important role in distinguishing the speaker from others [20]. Character extraction reduces the volume of the speech signal without harming the power of the speech signal [21].

Frontal processing usually consists of three sub-processes. First, some forms of speech activity detection are performed to remove non-speech components from the signal. Then, elements that provide information about the speaker are extracted from the speech. Despite the fact that the speech signal does not contain specific characteristics that indicate the identity of the speaker, it is known from the theory of the source filter in the speech source that the shape of the speech spectrum of the speaker's speech contains information about resonances through the vocal tract and information about fundamental tone harmonics through the glottal source. encodes through Thus, most speaker monitoring systems use some form of spectral response. Short-term analysis is typically used to calculate a sequence of amplitude spectra using LPC or FFT analysis, with 20 ms intervals every 10 ms. Often, amplitude spectra are converted to cepstral responses after passing through a bank of low-pass filters, and time-differential cepstrals are added. Typical function vectors have 24-40 elements. The final process in external interface processing is a form of channel compensation. Often, some form of linear channel compensation is applied to the elements. In addition to channel compensation in the sign domain, there are powerful

compensation methods, as well as effective matching methods, that can be applied to compatibility estimation models and domains [14].

Speaker Modeling During the recording process, the speaker's speech goes through the pre-processing steps described above, and a character vector is formed to create its model. Required attributes for the speaker model:

- theoretical basis for mathematical approach to understanding, extension and improvement of model behavior;
- generalization to new data that may not correspond to the model registration data and to new data;
- economic appearance in size and calculation.

There are many modeling techniques that have some or all of these attributes and are used in speaker verification systems. The choice of modeling depends primarily on the type of speech used, processing, ease of training and updating, and storage and computational issues. A brief description of the most common modeling methods is given below.

As mentioned above, speaker recognition includes two stages: a training stage and a testing stage. During the training phase, models of registered speakers are created and saved. To recognize a user, his voice sample is compared to the speaker model declared for the verification system, and all stored models if this is an identification task.

Speaker models are constructed from character vectors extracted from speech samples and can be template models or stochastic models. For template models, a fit score is calculated by estimating the distance between the observed speech pattern and the model. In stochastic models, a fit is obtained by measuring the probability that the observed speech pattern and the model are from the same speaker. This provides flexibility and flexibility over the template model.

Vector quantization, dynamic time warping, and nearest neighbors are examples of template models, while Gaussian mixture models and hidden Markov models are stochastic model classifications. Vector quantization and the Gaussian mixture model are the most studied methods and are discussed as examples of each model classification.

Compare with sample. In this method, the model consists of a template, which is a sequence of character vectors derived from fixed expressions. In testing, alignment and measurement of similarity between a test phrase and a speaker sample are evaluated using dynamic time warping (DTW). This approach is used almost exclusively in text-based applications.

Close neighbors. This method does not use an exact model; instead, all character vectors from the input speech are stored to represent the speaker. In the test, each test character vector is evaluated as the distance to its k nearest neighbors in the speaker vectors. Character vector reduction methods are widely used to limit storage and computation.

Neural networks. The specific model used in this can take many forms, such as multilevel perception or radial basis functions. The main difference from the other described approaches is that these models are specifically trained to distinguish between the modeled speaker and some alternative speakers. Training can be complex and resource-intensive, and models are sometimes not generalizable.

Hidden Markov Models (HMM). This method uses HMMs that encode the temporal evolution of elements and effectively model statistical changes in features to provide a statistical representation of how a speaker produces sounds. During registration, HMM parameters are estimated from speech using specified automatic algorithms. In the test, the probability of a

sequence of test functions is calculated from the speaker's HMM. For text-dependent applications, whole phrases or phonemes can be modeled using multi-state left-handed HMMs. For text-independent applications, single-state HMMs called Gaussian mixture models (GMMs) are used. Based on the results, HMM-based systems generally provide the best performance.

Modeling Speaker. Voice recognition differs from many biometric systems in that the object of recognition is a process rather than a static image, as in the case of fingerprint, face or pupil recognition. Therefore, often a sound sample is represented not as a single symbol vector, but as a sequence of symbol vectors, each of which describes the characteristics of a small part (one window) of the speech signal. After the signal processing stage, the sequence of vectors obtained is used to create a model of the speaker or to compare it with the built models.

One of the problems with using a speaker model matching estimate with a test phrase in a verification system is that non-speaker variability (e.g., text, microphone, noise) can lead to large test-to-test variance in the estimate, making decision-making difficult. Determining the desired threshold to achieve is difficult. By using a dummy model to generate a likelihood ratio estimate, the system can use this relative estimate to provide more consistent threshold settings. The idea of using a dummy model dates back to the early 1990s, but its use in speech verification systems is widespread and can be critical to good performance.

For closed set identification, this normalization is less important, since the classification is done according to the similarity score of the speaker model, ranking the models relative to each other. However, if the program is open source, some form of normalization is required to ensure stable threshold settings. Two primary approaches are used to represent the impostor model in a likelihood ratio test. In general, these approaches can be applied to any speaker modeling technique.

The first approach, known as probability sets, sets, or background sets, uses a set of other speaker models to compute pseudo fit estimates. The fit score of an impostor is typically calculated as a function, such as the maximum or average fit score from a set of non-impostor speaker models. Non-registered speaker models may come from other registered speakers or as fixed models from another corpus. Different ways to choose and use a set of background models are covered.

A second approach, known as general, universal, or common background modeling, uses a single speaker-independent model trained on the speech of a large number of speakers to represent the speaker-independent speech [6]. The idea here is to represent impostors using a generic speech pattern that is compared to a particular speaker's speech pattern. An advantage over the cohort approach is that only one mock model is trained and evaluated. This approach also allows using Maximum A-Posteriori training to match the candidate model to the background model, which can improve performance and reduce the model's computational and storage requirements.

Vector quantization method. Vector quantization (VQ) is the process of comparing vectors from a large vector space to a finite number of fields in that space. Each area is called a cluster and can be represented by its center called a code word. A collection of all code words constitutes a code book. A conceptual diagram of this reflection process is shown in Figure 1, with only two speakers and two dimensions of acoustic space. The distance from the vector to the nearest codeword of the codebook is called VQ distortion. In the testing phase, the speech input from the unknown speaker is "vector quantized" using each training codebook and the total distortion VQ is calculated. A speaker matching the VQ codebook with the least total distortion is determined.

Clustering training vectors. Acoustic vectors (MFCC) obtained from speakers' speech samples are used to create codebooks for them using the VQ method. The Linde Buzo Gray (LBG) algorithm [18] is a popular algorithm for performing the VQ process.

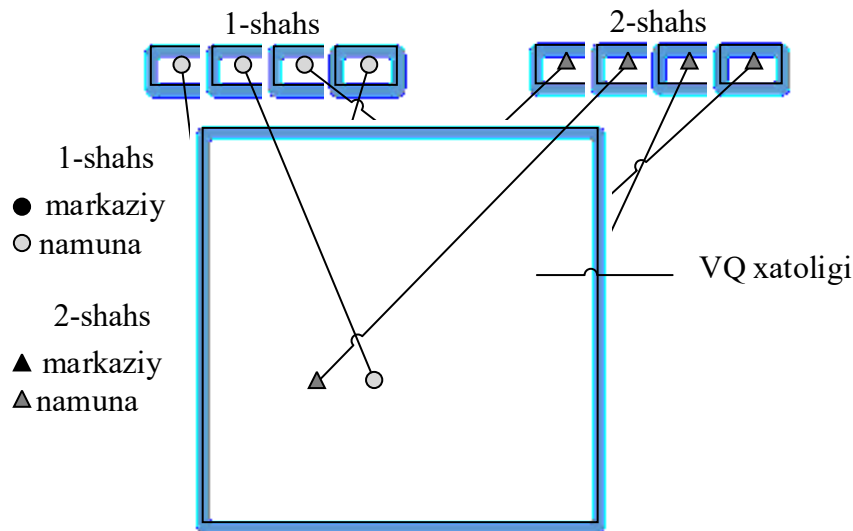


Figure 1 – Conceptual diagram of vector quantization codebook formation

The LBG algorithm implements the VQ process through the following iterative procedure:

1. A codebook is created using one code word. This code word is the center of all vectors.
2. Double the size of the codebook by splitting the current codebook in half according to the following rule:

$$y_n^+ = y_n(1 + \varepsilon), y_n^- = y_n(1 - \varepsilon)$$

here y_n - previous code book, y_n^+ and y_n^- - new code books, from n 1 code will be resized to the current size of the book, ε - the splitting parameter is usually taken as 0.01.

1. Based on the similarity measure, vectors are clustered around code words. Each vector is associated with the nearest code word.
2. By determining the center of clusters, new code words are obtained for each cluster.
3. Steps 3 and 4 are repeated until the average distance falls below the specified threshold.
4. Repeat steps 2, 3 and 4 until a codebook of the required size is generated.

Gaussian mixture model (GMM). An important step in the implementation of the above similarity ratio detector is the selection of the true similarity function. The choice of this function depends largely on the functions used, as well as the specific features of the program. GMM is the most successful similarity function for text-independent speaker recognition when what the speaker will say is unknown in advance. In text-based applications with strong prior knowledge of spoken text, hidden Markov input for similarity functions can be used. However, to date, the use of more complex similarity functions such as those based on NMM has not been shown to be superior to GMM for text-independent speaker recognition problems such as NIST (SRE). Typically, systems using Gaussian mixture models use diagonal covariance matrices.

EM-algorithm. GMM is defined by the following set of parameters:
 $\lambda = \{\omega_i, \vec{\mu}_i, \Sigma_i\}, i = 1 \dots L.$

The GMM parameters can be easily determined using the EM-algorithm.

EM is an abbreviation of the English words "expectation" (waiting) and "maximization" (maximization), and the two stages of the algorithm are called this. To the introduction of the EM-algorithm $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_W\}$ a sequence vector of training symbols is transmitted. There is a problem with initializing the model parameters before starting the training process. The YEM-algorithm does not guarantee to find the global maximum in the space of the training sequence of vectors, and therefore the result of training the system depends significantly on the initial values of the parameters. The k-means (k-means) clustering algorithm is often used to run the models [22]. In this case, the initial values correspond to the centers of the clusters, the initialization of the covariance matrices is based on the training vectors in this cluster, the weight coefficients of the components are defined as the ratio of the training vectors. After the model parameters are initialized with initial values, they are re-estimated using two steps of the YEM algorithm:

The expected value of the posterior probability is calculated for each component - "Expectation" stage

$$p(i | \vec{x}_w, \lambda) = \frac{v_i p_i(\vec{x}_w)}{\sum_{i=1}^L \omega_i p_i(\vec{x}_w)}$$

New parameters of the model are calculated - "Maximization" stage

$$\omega_i = \frac{1}{W} \sum_{w=1}^W p(i | \vec{x}_w, \lambda)$$

$$\vec{\mu}_i = \frac{\sum_{w=1}^W n(i | \vec{x}_w, \lambda) \vec{x}_w}{\sum_{w=1}^W p(i | \vec{x}_w, \lambda)},$$

$$\Sigma_i = \frac{\sum_{w=1}^W p(i | \vec{x}_w, \lambda) (\vec{x}_w - \vec{\mu}_i)(\vec{x}_w - \vec{\mu}_i)}{\sum_{w=1}^W p(i | \vec{x}_w, \lambda)}.$$

These steps are repeated until the parameters converge.

Universal background model. In addition to the EEM algorithm, there is another way to estimate the suhandon model based on Gaussian mixtures. It is based on the use of the so-called universal background model (UFM, UBM, universal background model), which is a Gaussian mixture model trained using a large number of samples. This method makes it possible to speed up the calculation of the suhandon model and improve the quality of the suhandon recognition system in comparison with the YEM algorithm. There are several ways to calculate UMF. The simplest is to combine many candidate character vectors into one training sequence and obtain the UFM using the EEM algorithm. When using this method, it is necessary to take into account the balance of different subclasses in the combined set of character vectors. For example, when implementing a voice recognition system that should work equally well with voice actors of both sexes, a balance between male and female voice recordings should be maintained when receiving UFM. The same is true for recordings from different microphones [1].

Another way to calculate the UFM is to obtain a model for each subclass and combine their components into a single model. The weight coefficients of the resulting UFM are then recalculated so that their sum is equal to one. This method allows using unbalanced data for training and controlling the composition of the final UFM. Other methods can be used to calculate UFM.

The method of maximum a posteriori estimation (MAP-estimation) is used to obtain a model for a specific candidate. This step is also called "adaptation" in some sources, which means that the UFM is adjusted to obtain a speaker-independent dynamic model. The first step is to perform one iteration of the EEM algorithm using formulas (1.23)–(1.26), using the symbol vectors of the suhandon training signal as input and the UFM parameters as initial model parameters.

It is experimentally shown that the results improve when the fit is performed only for the mathematical expectation vectors, and it is shown that r^μ values 8 to 20 uses are suggested. In the article was used $r^\mu = 16$. Classification of speech signals. $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_W\}$ character vectors that best match the presented sequence S^x the degree of similarity of this sequence in the base and of each suhandon to determine the speaker.

It is necessary to evaluate the model $S = \{1, 2, \dots, N\}$, and then find the speaker corresponding to its maximum value. When Gaussian mixtures are used to model the noise, it requires finding the model with the highest posterior probability. The probability of all words is the same, i.e. $P(\lambda^n) = 1/N$ and $p(X)$ taking into account that the magnitude is the same for all candidates, the classification rule is simplified to the following form: $S^X = \underset{ISMSN}{\operatorname{argmax}} p(X | \lambda^n)$.

$p(X | \lambda^n)$ - n It is not difficult to see that is a similarity function depending on. If $p(X | \lambda^n)$ similarity function $L(X | \lambda^n)$ is replaced by the logarithmic similarity function, assuming that the individual vectors in X are not connected, it is calculated as follows:

$$L(X | \lambda^n) = \sum_{w=1}^W \sum_{i=1}^L \ln(\omega_i^n N(\vec{x}_w | \vec{\mu}_i^n, \Sigma_i^n)),$$

here $\lambda^n = \{\omega_i^n, \vec{\mu}_i^n, \Sigma_i^n\}$ – n -speaker model. The resulting classification rule is defined as follows: $S^X = \underset{1 \leq \pi \leq N}{\operatorname{argmax}} L(X | \lambda^n)$.

In conclusion, it can be said that in addition to the above-described classifier of speech signals based on GAM, methods such as distance calculation, nearest neighbor method, k-means algorithm, and vector quantization can be used in speaker recognition. Binary classifiers based on the method of support vectors (SVM, support vector machine) are also often used in speaker verification.

REFERENCES

1. H. Beigi. Fundamentals of speaker recognition. Springer US, 2011.
2. Amrouche, A. Effect of GSM speech coding on the performance of Speaker Recognition System / A. Amrouche, A. Krobba, M. Debyeche // 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA): Book of abstracts. - Kuala Eurnpur, 2010. - pp. 137-Nickel, R., "Automatic Speech Character Identification", IEEE Circuits and Systems Magazine, vol. 6, no. 4, 2006, pp 8-29.
3. .Зилинберг А.Ю. Разработка и исследование временных и спектральных алгоритмов VAD (Voice Activity Detection) //А.Ю.Зилинберг, Ю.А.Корнеев //Российская школа-

- конференция «Мобильные системы передачи данных» / Зеленоград: МИЭТ, 2006. – С. 58–70.
4. Рабинер Л., Шафер Р. Цифровая обработка речевых сигналов. – М.: Радио и связь, 1981. – 496 с.
 5. <https://www.itu.int/rec/T-REC-G.729>
 6. Маматов Н.С., Нуримов П.Б., Самижонов А.Н. Нутқ сигналларида овоз фаоллигини аниқлаш алгоритмлари. «Ахборот коммуникация технологиялари ва дастурий таъминот яратишда инновацион ғоялар» Республика илмий-техник конференцияси 17-18 май 2021 йил.
 7. X.-L. Zhang and J. Wu. Denoising deep neural networks based voice activity detection. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 853–857. IEEE, 2013.
 8. Hariharan M, Vijejan V, Fook CY, Yaacob S. Speech stuttering assessment using sample entropy and Least Square Support vector machine. In: 8th International Colloquium on Signal Processing and its Applications (CSPA). 2012. pp. 240-245
 9. Manjula GN, Kumar MS. Stuttered speech recognition for robotic control. Ahmad AM, Ismail S, Samaon DF. Recurrent neural network with backpropagation through time for speech recognition. In: IEEE International Symposium on Communications and Information Technology (ISCIT 2004). Vol. 1. Sapporo, Japan: IEEE; 2004. pp. 98- 102
 10. Shaneh M, Taheri A. Voice command recognition system based on MFCC and VQ algorithms. World academy of science. Engineering and Technology. 2009;57:534-538
 11. Mosa GS, Ali AA. Arabic phoneme recognition using hierarchical neural fuzzy petri net and LPC feature extraction. Signal Processing: An International Journal (SPIJ). 2009;3(5): 161
 12. Yousefian N, Analoui M. Using radial basis probabilistic neural network for speech recognition. In: Proceeding of 3rd International Conference on Information and Knowledge (IKT07), Mashhad, Iran. 2007
 13. Cornaz C, Hunkeler U, Velisavljevic V. An Automatic Speaker Recognition System. Switzerland: Lausanne; 2003. Retrieved from: http://read.pudn.com/downloads60/sourcecode/multimedia/audio/209082/asr_project.pdf
 14. Shah SAA, ul Asar A, Shaukat SF. Neural network solution for secure interactive voice response. World Applied Sciences Journal. 2009;6(9):1264-1269 Some Commonly Used Speech Feature Extraction Algorithms <http://dx.doi.org/10.5772/intechopen.80419> 17
 15. Ravikumar KM, Rajagopal R, Nagaraj HC. An approach for objective assessment of stuttered speech using MFCC features. ICGST International Journal on Digital Signal Processing, DSP. 2009;9(1):19-24
 16. Kumar PP, Vardhan KSN, Krishna KSR. Performance evaluation of MLP for speech recognition in noisy environments using MFCC & wavelets. International Journal of Computer Science & Communication (IJCSC). 2010;1(2):41-45
 17. Kumar R, Ranjan R, Singh SK, Kala R, Shukla A, Tiwari R. Multilingual speaker recognition using neural network. In: Proceedings of the Frontiers of Research on Speech and Music, FRSM. 2009. pp. 1-8
 18. Narang S, Gupta MD. Speech feature extraction techniques: A review. International Journal of Computer Science and Mobile Computing. 2015;4(3):107-114

19. Козлов А.В. Система идентификации дикторов по голосу для конкурса NIST SRE 2013 // А.В.Козлов, О.Ю.Кудашев, Ю.Н.Матвеев, Т.С.Пеховский, К.К.Симончик, А.К.Шулипа // Труды СПИИРАН. – 2013. – № 2. – С. 350–370.
20. S. Furui. An overview of speaker recognition technology. In Automatic speech and speaker recognition, pages 31–56. Springer, 1996.
21. Макхоул Д. Векторное квантование при кодировании речи / Д.Макхоул, С.Рукос, Г.Гиш // ТИИЭР. – 1985. – Т.73. – №11. – С. 19–61.