International Journal of Applied Science and Engineering

Software for preprocessing voice signals

Narzillo Mamatov*, Nilufar Niyozmatova, Abdurashid Samijonov

Tashkent University of Information Technologies named after Al-Kharezmi, Tashkent, Uzbekistan

ABSTRACT

One of the most important tasks of modern science is the development of software tools for human communication with devices (for example, a computer) in natural language, where speech input and output of information is carried out in the most user-friendly way. To create such tools, it is required to solve speech recognition problems. On the basis of many experimental studies, it can be concluded that the quality of speech recognition depends on the results of preliminary signal processing. Improving the quality of speech recognition requires new efficient and high-speed signal preprocessing methods and algorithms.

This article proposes a new approach and algorithm for the formation of signs of speech signals. Based on these features obtained by the proposed algorithm, the identification problem is solved. The article also provides a description of the software module for each stage of preprocessing of speech signals. The developed software is a voice-based identification tool.

Keywords: Algorithm, Signal, Speech signal, Filter, MFCC, PLP, LPCC.

1. INTRODUCTION

Sound is a superposition of sound vibrations (waves) of different frequencies. And speech consists of a sequence of sounds. We know from physics that a wave is characterized by amplitude and frequency. When processing speech signals, filtering and noise suppression, amplification, separation of information streams, information extraction, coding, compression and restoration of speech signals are carried out. Speech processing is widely used in all areas of speech technology.

2. SOLUTION OF THE PROBLEM

The developed software for the preliminary processing of speech signals consists of several stages, and each stage includes several modules (Fig. 1).

Stage 1. The initial stage of software development is called “Speech Signals”, where speech signals are received from a file or through a microphone.

Stage 2. The second stage is called speech preprocessing. At this stage, the signal is processed by the following modules:

Filling module. This is where the gaps in speech signals are filled.

Voice activation detection (VAD) module. This module is used to search for speech in audio. Typically, VAD is used to efficiently and reliably detect speech in sound and even in background noise. In completely clear audio recordings, even rudimentary energy detection may be sufficient for speech recognition; but, unfortunately, there are not always completely clean signals in the wild, so VAD must be noise-immune. VAD consists of steps:

- Signal division into frames.
- Formation of features for each frame.
- Classifier training in active and quiet frames.

OPEN ACCESS

Received: July 28, 2020

Accepted: November 3, 2020

Corresponding Author:
Narzillo Mamatov
m_narzullo@mail.ru

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

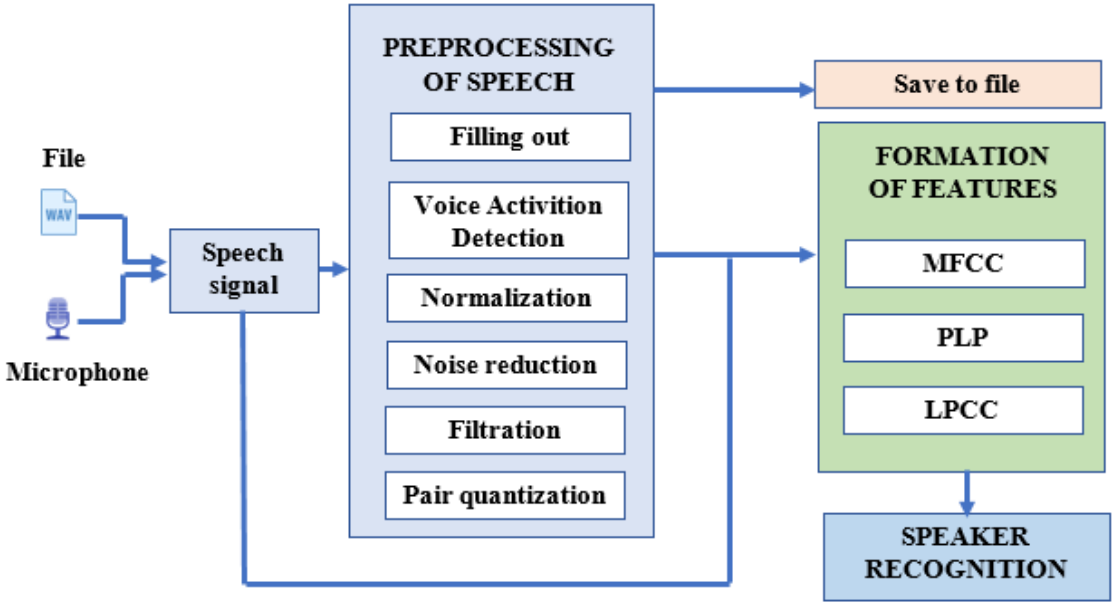

Fig. 1. The structure of the software

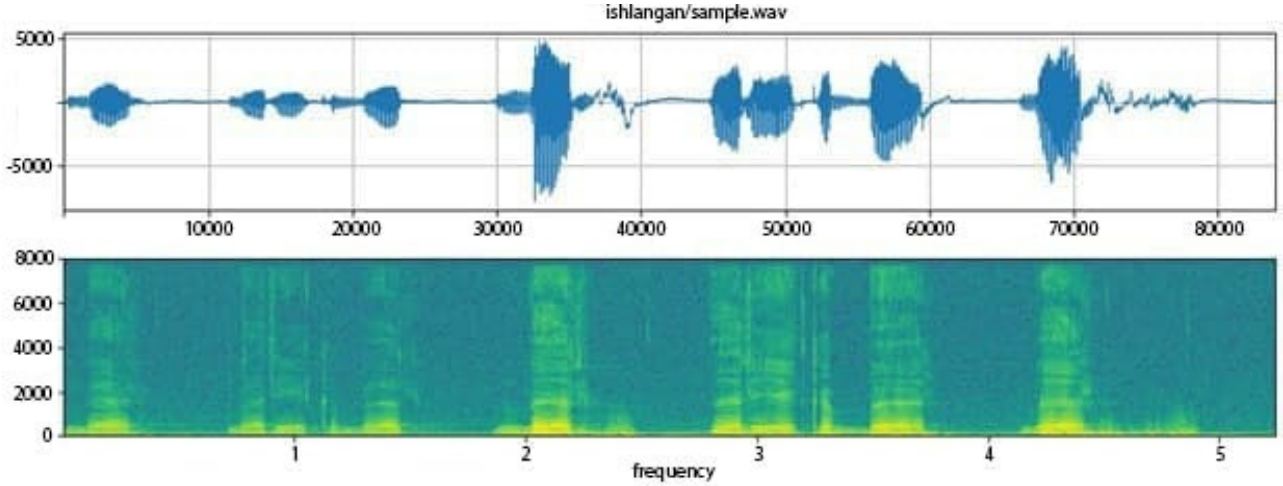

Fig. 2. The speech signal before applying VAD

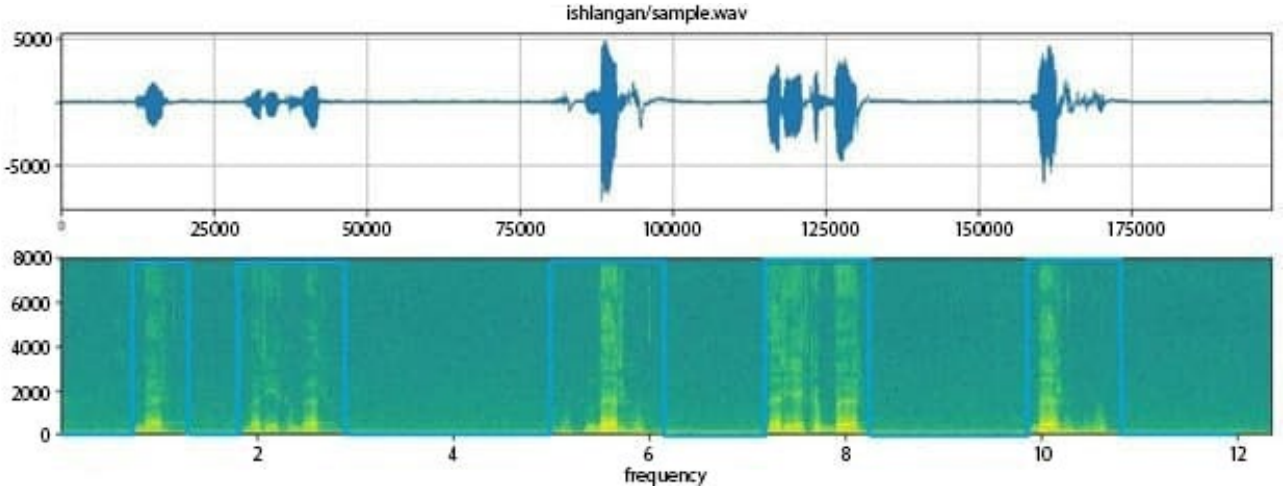

Fig. 3. Speech signal after applying VAD

- Classification of invisible frames as speech or silence.

An example on the created software using the function of detecting voice activity for a speech signal (https://github.com/wiseman/py-webrtcvad/) is shown in Fig. 2 and Fig. 3.

Normalization module. Volume normalization EBU R128 consists of two modes: dynamic and linear. Support is provided as one-pass (live streams, files) and two-pass (files) modes. This algorithm can target integrated loudness (IL), loudness range (LRA), and maximum true peak. If the normalization mode is not linear, then the audio stream will be sampled to 192 kHz to accurately determine the true peaks. To explicitly set the sample rate of the output, we need to use the -ar option or a sample filter (FFmpeg. https://ffmpeg.org/ffmpeg-filters. html#loudnorm).

The filter has the following variable parameters (Fig. 4):

- I, i - setting the built-in volume range. Here, the default values are -24.0, and the interval [-70.0; -5.0].
- LRA, lra - setting the target volume range. Here the default values are 7.0 and the interval is [1.0; 20.0].
- TP, tp - setting the maximum true peak. Here the default values are -2.0 and the interval is [-9.0; 0.0].

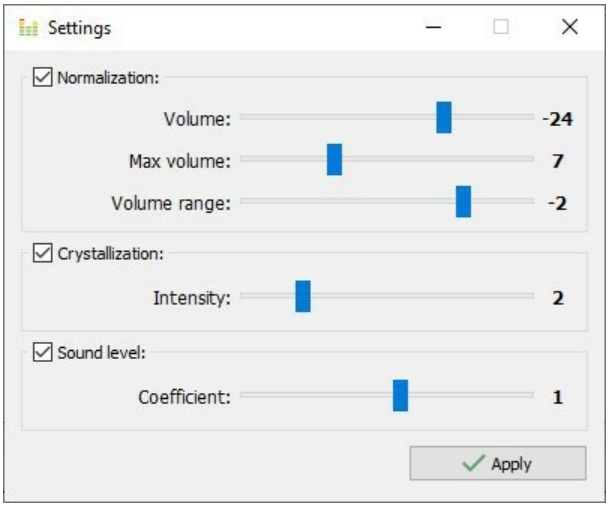

Fig. 4. Normalization window: ffmpeg.filter(stream, filter_name = "loudnorm", i = i, lra = lra, tp = tp)

Noise reduction module. The authors of Wiedecke et al. (2019) described in detail the noise reduction of audio samples using the FFT. In experimental studies, the parameters were set with the corresponding values (Fig. 5):

- nr - the noise reduction (dB), the default value is 12 (dB), and the allowable range is nr from 0.01 to 97.
- nf - the minimum noise level in dB, where the allowable range is from -80 to -20. The default value is -50 dB.
- nt - the type of noise that has the parameters: wn is white noise, vn is vinyl noise, sn is shellac noise and cn is the user noise defined in option bn. nt parameter defaults to white noise.
- bn is the user noise range which is each of the 15 bands, where the bands are separated by " or '|'.
- rf - the residual level (dB), where the permissible range of rf is from -80 to -20. The default value is -38 (dB).
- tn - the noise tracking, which takes the values 1 (enable) or 0 (disable). The default is 0. With this power on, the noise level is automatically adjusted.
- tr - the track balances, enable or disable. The default is 0.

Filtration module. This module is one of the main modules for processing time or spatial series of measurements. Currently, there are many filtering methods such as Savitsky- Golay filter, median filtering, polynomial

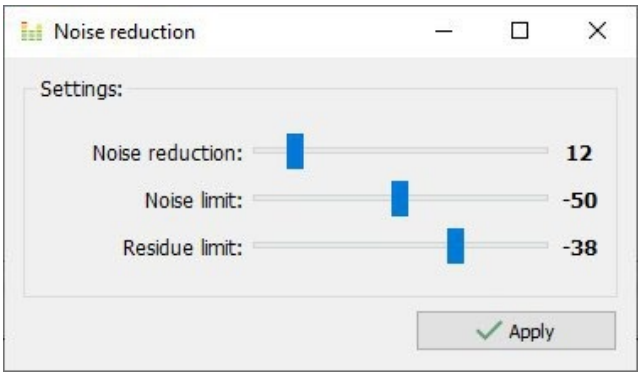

Fig. 5. Noise reduction settings: ffmpeg.filter(stream, filter_name = "afftdn", nr = nr, nf = nf, rf = rf)

approximation, cosine filtering, Fourier transform, wavelet transform, etc.

The Savitsky-Golay filter is a digital filter. The Savitsky-Golay filter is applied to a set of digital data points for smoothing data, where to improve data accuracy without distorting the signal. Accuracy is maximized in the process, known as convolution, by selecting successive subsets of adjacent data points using a low-degree polynomial using the least squares method. If the data points are located at the same distance, then the analytical solution of the least squares equations can be found in a single set of "convolution coefficients," which can be applied to all subsets of data to obtain smoothed estimates. A signal (or derivatives of a smoothed signal) at the center point of each subset. This method, based on established mathematical procedures, was popularized by Abraham Savitsky and Marcel J. E.

The created software uses the savgol_filter (https://docs.scipy.org/doc/scipy0.15.1/reference/generated /scipy.signal. savgol_filter.html.) function in the Python programming language. Filtering options are given in Fig. 6. The results of applying the Savitsky-Golaya filter for windows of different lengths are shown in Fig. 7 and Fig. 8.

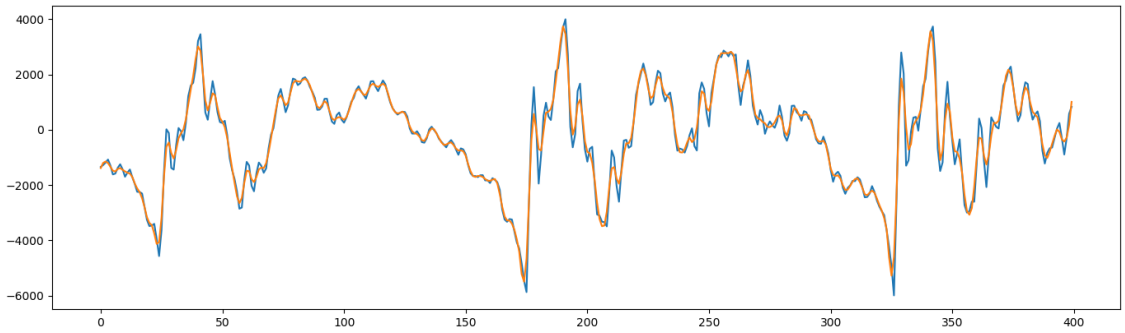

Fig. 7. An example of applying the Savitsky-Golay filter. Window length = 5, polynomial order = 3

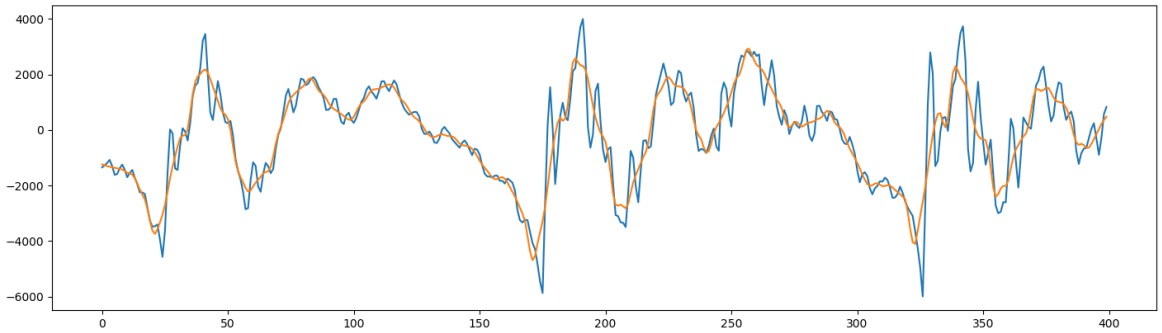

Fig. 8. An example of applying the Savitsky-Golay filter. Window length = 15, polynomial order = 3

Pair quantization.

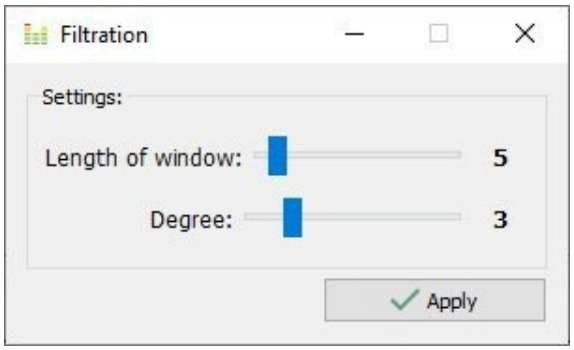

Fig. 6. Filtration window: scipy.signal.savgol_filter(x, window_length, polyorder)

Stage 3. At this stage, the pre-processed speech signal can be saved to a file in audio format.

Stage 4. This stage is called by us "Formation of features". At this stage, the features of speech signals are formed as: Mel-frequency cepstral coefficient (MFCC), Linear Predictive Coding (LPC), Perceptual Linear Prediction **(**PLP) and medium quantum (Fig. 10).

The cepstral linear prediction coefficient method, the perceptual linear prediction coefficient method and robust PLP (PLP-RASTA), the method of MFCC on the chalk scale are the most powerful methods based on cepstral signal analysis.

Linear Prediction Cepstrum Coefficients (LPCC) is a cepstral linear prediction coefficient method. Based on the calculation of the coefficients of the autoregressive model for each frame of the audio signal. After obtaining all the model parameters, cepstral LPCC coefficients are calculated based on the recursive function.

PLP is a perceptual linear prediction coefficient method. The method differs from the LPCC method in that it takes into account the characteristics of the perception of various frequencies by a person - before calculating the parameters of the autoregressive model, the signal undergoes a certain pre-processing. The calculated instantaneous Fourier spectrum is converted into a spectrum on the barque scale, after which the operation of convolution of the masking curves of the critical bands with the obtained spectrum is performed to obtain the frequency masking effect. Next, the volume curve and cepstral processing are approximated.

The advantage of the PLP method compared to LPCC is that it allows you to suppress information related to the individual characteristics of the speaker by choosing the appropriate model order. However, this method is more sensitive to the pitch frequency.

Mel-frequency cepstral coefficient (MFCC). MFCC - based on human auditory perception, and obtained on a scale of twisted frequencies. To calculate the MFCC, a speech window is first created to divide the speech signal into frames. To obtain the same amplitude for all formats, high-frequency formants are reduced in amplitude, compared with low-frequency formants, high frequencies are emphasized. After creating a window, a fast Fourier transform is applied to find the power of the spectrum for

each frame. After application (FFT) based on the filter base using the melting scale, it is processed by the power spectrum. To calculate the MFCC coefficients after converting the power spectrum to the logarithmic region, a DCT is applied to the speech signal. The following formula is used to calculate Mel for an arbitrary frequency (Chakroborty et al., 2006; Hasan et al., 2004):

$$mel(f) = 2595 * \log_{10}\left(1 + \frac{f}{700}\right) \quad (1)$$

where *mel* (*f*) -Mel frequency, a *f* - frequency.

MFCC are calculated based on the following formula:

$$\hat{C}_n = \sum_{n=1}^{k}\left(\log S_k\right)\cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right] \quad (2)$$

where k – is the number of Cepstral melting factors, $\hat{S}_k$ is the output of filterbank and $\hat{C}_n$ is the final mfcc coefficients, C is the number of MFCCs ($n = \overline{0, C-1}$).

Based on the MFCC, the low-frequency region is effectively determined than the high-frequency region and it can calculate formants lying in the low-frequency range, and also describes the resonances of the speech tract.

In Fig. 9 shows a general view of the developed software pre-processing speech signals.

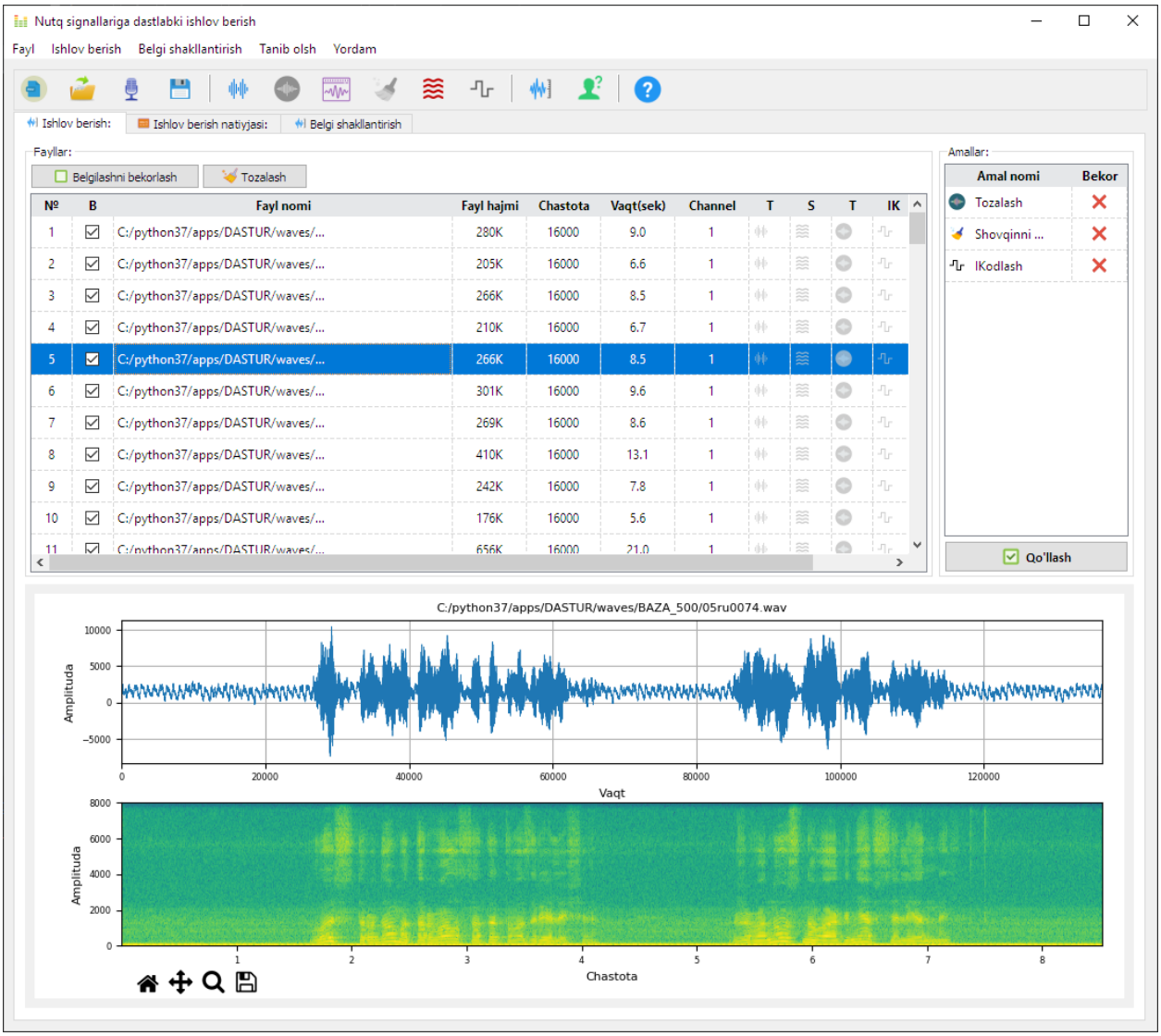

Fig. 9. Window software pre-processing speech signals

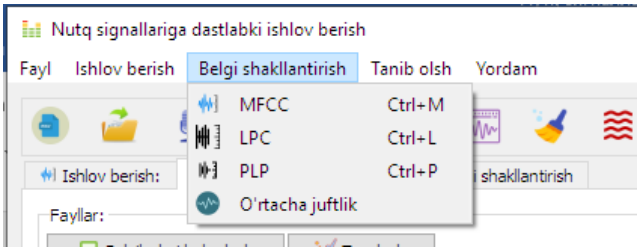

Fig. 10. Menu of formation features

MFCC is universal and recognized as an interface procedure for typical identification applications (Chakroborty et al., 2006). In addition, MFCC is an ideal representation when the source characteristics are stable and consistent for sounds (Chu et al., 2008). MFCC is also able to capture information from sampled signals in which the frequency is not more than 5 kHz. Such signals cover most of the energy of sounds generated by humans.

In speech recognition, MFCC is widely used (Ravikumar et al., 2008). Formants exist above 1 kHz and they are not taken into account due to the large distance in the high-frequency range between the filters (Chakroborty et al., 2006). In the presence of MFCC background noise, the signs are not accurate and cannot be generalized (Hasan et al., 2004).

Perceptual Linear Prediction (PLP). Based on the PLP method, critical bands are combined, the intensity is compressed into loudness and a preliminary emphasis on equal loudness when extracting relevant information from speech. PLP is based on a non-linear scale and was originally intended for use in speech recognition tasks by eliminating functions that depend on the speaker (Ravikumar et al., 2009). PLP gives a representation corresponding to a smoothed short-term spectrum that has been aligned and compressed, similar to human hearing, which makes it look like MFCC. The PLP approach repeats some important features of hearing, and the subsequent auditory spectrum of speech is approximated by an autoregressive pan-polar model (Hermansky, 1990). PLP gives minimal resolution at high frequencies, which means an approach based on a bank of auditory filters, but gives orthogonal results that are similar to cepstral analysis. It uses linear predictions for spectral smoothing, so the name is perceptual linear prediction (Kumar and Chandra, 2011). PLP is a combination of spectral analysis and linear prediction analysis.

To calculate the characteristics of PLP, speech is highlighted as a window (Hamming window), the Fast Fourier Transform (FFT) is calculated and squared. This gives energy spectral estimates. The trapezoidal filter is then applied at intervals of 1 cortex to integrate the overlapping critical-band filter responses in the power spectrum. This effectively compresses higher frequencies into a narrow band. Then, convolution of the symmetric frequency domain on the scale of distorted frequencies of the cortex allows low frequencies to mask high frequencies, smoothing the spectrum. Subsequently, the spectrum is preliminarily emphasized in order to approach the uneven sensitivity of the human hearing at different frequencies. The spectral amplitude is compressed, this reduces the change in the amplitude of the spectral resonances. The inverse discrete Fourier transform (IDFT) is performed to obtain autocorrelation coefficients. Spectral smoothing is performed by solving the autoregressive equations. Autoregressive coefficients are converted to cepstral variables (Kumar and Chandra, 2011).

PLP is based on a non-linear scale and was first used in speech recognition problems to eliminate speaker-dependent functions (Ravikumar et al., 2009). PLP represents a smoothed aligned and compressed, similar to human hearing, short-term spectrum, transforms it as MFCC.

The PLP approach has some important hearing features, as the auditory spectrum of speech is approximated by an autoregressive pan-polar model (Hermansky, 1990). PLP gives minimal resolution at high frequencies and is based on a bank of auditory filters, but it gives results similar to cepstral analysis.

For spectral smoothing, linear predictions are also used, therefore it is called as perceptual linear prediction (Kumar and Chandra, 2011). PLP consists of a combination of spectral and linear prediction analysis.

To calculate the PLP characteristics, the speech is divided, then the fast Fourier transform is performed on these windows and the obtained values are squared. This characterizes the energy spectral estimates. Then a trapezoidal filter is applied at intervals of 1 cortex.

This makes it possible to integrate overlapping critical-band filter responses in the power spectrum. As a result of integration, high frequencies are effectively compressed into a narrow band.

Then, on the scale of distorted frequencies of the cortex, convolution is performed in the symmetric frequency domain, which allows smoothing the spectrum, where low frequencies mask with high frequencies.

To get closer to the uneven sensitivity of human hearing at different frequencies, the spectrum is subsequently preliminarily emphasized. To reduce the change in the amplitude of the spectral resonances, the spectral amplitude is compressed.

To obtain autocorrelation coefficients, the inverse discrete Fourier transform (IDCT) is performed and spectral smoothing is performed by solving the autoregressive equations.

On the basis of autoregressive coefficients, cepstral variables are calculated (Kumar and Chandra, 2011). To calculate the frequency of the cortex, the following formula is used:

$$bark(f) = \frac{26.81 f}{1960 + f} - 0.53 \quad (3)$$

where *bark* (*f*) is the core frequency, and *f* is the frequency in Hz.

The identification results based on PLP are much better than LPC (Kumar and Chandra, 2011). It justifies that PLP effectively suppresses speaker-specific information. In addition, it has improved independent recognition characteristics and is resistant to noise, changes in channels and microphones.

Based on PLP, autoregressive noise components are accurately restored. In addition, PLP is more sensitive to any changes in the frequency of formants.

Linear Prediction Cepstrum Coefficients (LPCC). LPCC is the cepstral coefficient, which is obtained from the calculated envelope of the LPC spectrum. The LPCC coefficients are illustrations of the Fourier transform of the logarithmic spectrum of quantities LPC (El Choubassi et al., 2003; Wu et al., 1997).

Cepstral analysis ideally symbolizes speech signals and characteristics with a limited size of functions and therefore they are usually used in the field of speech processing (Wu et al., 1997).

Rosenberg and Sambur noted that adjacent predictor coefficients have a high correlation, and with less correlated characteristics the representations will be more effective, therefore LPCC is of this kind.

If the signals have a minimum phase, then the LPC is easily converted to LPCC (Holambe and Deshpande, 2012).

In speech processing based on LPCC, they are likewise computed as LPC (Wu et al., 1997).

LPCC are calculated based on the following formula (Holambe and Deshpande, 2012):

$$C_m = a_m + \sum_{k=1}^{m-1} \left[\frac{k}{m}\right] c_k \, a_{m-k}, \quad (4)$$

$$0 \le k \le N-1, 0 \le m \le M-1,$$

here a_m is the linear prediction coefficient, c_m is the cepstral coefficient, M is total number of triangular mel weighting filters, N is the number of points used to compute the DFT.

LPCCs have a lower error rate than LPC functions (Wu et al., 1997) and have a slight noise vulnerability (El Choubassi et al., 2003). High order cepstral coefficients are mathematically limited. This leads to an extremely vast array of variances when moving from lower order cepstral coefficients to higher order cepstral coefficients. Similarly, LPCC estimates are known for their high sensitivity to quantization noise.

In a high-frequency speech signal, cepstral analysis gives a slight separability between the source and the filter in the region of quantity. Cepstral coefficients having a lower order are sensitive to spectral tilt, and those with a higher order are sensitive to noise.

The result of this step, i.e. the processed speech signal is transmitted to the identification step. If the identification result is lower than expected, proceed to stage 3 and the process continues.

Stage 5. This stage is called by us "Speaker recognition (identification). At this stage, the processed speech signal is used to identify a person by voice. If the result is lower than the specified result, go to stage 3, and the process continues (Fig. 10).

Vector quantization (VQ). Vector quantization (Gersho and Gray, 1991) is an efficient data compression method and has been successfully applied in various applications, including vector quantization coding and vector quantization recognition.

To generate codebooks, the LBG algorithm is used (Linde et al., 1980; Gersho and Gray, 1991). The steps of the LBG algorithm are as follows (Kekre and Kulkarni, 2010):

1. Development of a 1-vector code book; this is the centroid of the entire set of training vectors.
2. Doubling the codebook size by dividing each current codebook y_n according to the rule:

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

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

where n varies from 1 to the current codebook size, and ε is a splitting parameter.

3. Finding centroids for a shared codebook (i.e., the codebook is twice as large).
4. Repeat stages 2 and 3 until a table of the required size is developed.

Euclidean distance measure. Euclidean distance is used to measure the similarities or differences between two spoken words that occur after quantizing a spoken word in its codebook. The unknown word is compared by measuring the Euclidean distance between the feature vector of the unknown word and the model (code book) of known words in the database. The word with the smallest average minimum distance is selected as shown in the equation below

$$d(x,y) = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2} \quad (5)$$

where x_i – ith vector input features, y_i – ith vector features in the codebook, d - distance between x_i and y_i.

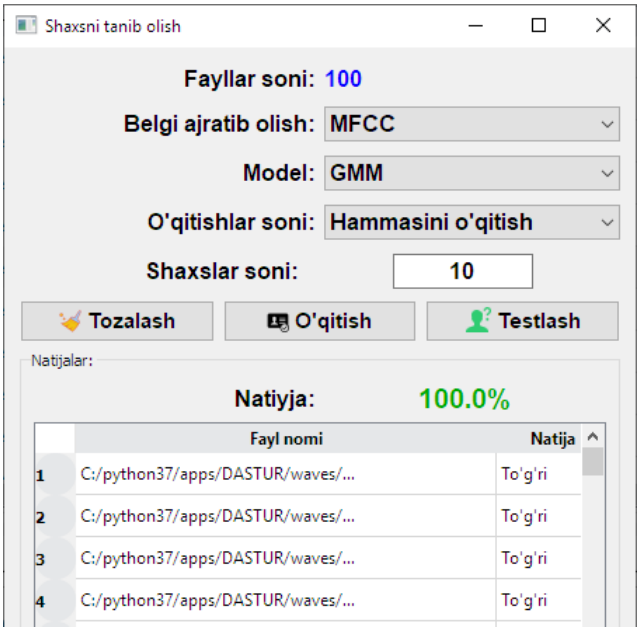

Fig. 11. Identification module

3. CONCLUSION

The article deals with the problem of creating software for preliminary processing of speech signals. For each stage, widely used methods and algorithms are selected. For the selected methods and algorithms, the corresponding software modules have been developed. In addition, to extract features, a pair quantization algorithm was proposed, in which the number of features obtained using this algorithm is at least two times less than the number of features obtained using existing algorithms. This speeds up identification. On the basis of the proposed algorithm, experimental studies have been carried out on the example of solving a practical problem. As a result of the work done, a software structure for preliminary processing of speech signals for the speech recognition and identification system is proposed. It is planned to develop an automatic speech recognition system based on the attributes of the pair quantization algorithm.

REFERENCES

Chakroborty, S., Roy, A., Saha, G. 2006. Fusion of a complementary feature set with MFCC for improved closed set text-independent speaker identification. In: IEEE International Conference on Industrial Technology, ICIT 2006. 387–390.

Chu, S., Narayanan, S., Kuo, C.C. 2008. Environmental sound recognition using MP-based features. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008. IEEE, 1–4.

El Choubassi, M.M., El Khoury, H.E., Alagha, C.E.J., Skaf, J.A., Al-Alaoui, M.A. 2003. Arabic speech recognition using recurrent neural networks. In: Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No.03EX795). Ieee, 543–547. DOI: 10.1109/ISSPIT.2003.1341178.

FFmpeg. https://ffmpeg.org/ffmpeg-filters. html#loudnorm.

Gersho, A., Gray, R.M. 1991. Vector quantization and signal compression. // Kluwer Academic Publishers, Boston, MA.

Github. https://github.com/wiseman/py-webrtcvad/

Hasan, M.R., Jamil, M., Rabbani, G., Rahman, M.G.R.M.S. 2004. Speaker identification using Mel frequency cepstral coefficients. In: 3rd International Conference on Electrical & Computer Engineering, ICECE 2004. 28–30.

Hermansky, H. 1990. Perceptual linear predictive (PLP) analysis of speech. The Journal of the Acoustical Society of America. 87, 1738–1752.

Holambe, R., Deshpande, M. 2012. Advances in non-linear modeling for speech processing. Berlin, Heidelberg: Springer Science & Business Media.

Kekre, H.B., Kulkarni, V. 2010. Speaker Identification by using Vector Quantization. // International Journal of Engineering Science and Technology. 2, 1325–1331.

Kumar, P, Chandra, M. 2011. Speaker identification using Gaussian mixture models. MIT International Journal of Electronics and Communication Engineering. 1, 27–30.

Linde, Y., Buzo, A., Gray, R.M. 1980. An algorithm for vector quantizer design. // IEEE Trans. Communication, COM-28, 84–95.

Narzillo, M., Abdurashid, S., Parakhat, N., Nilufar, N. 2019. Automatic speaker identification by voice based on vector quantization method, Int. J. Innov. Technol. Explor. Eng., 8, 2443–2445.

Narzillo, M., Abdurashid, S., Parakhat, N., Nilufar, N. 2019. Karakalpak speech recognition with CMU sphinx, Int. J. Innov. Technol. Explor. Eng., 8, 2446–2448.

Rabiner, L.R. 1981. Digital processing of speech signals. – M.: Radio and communications, –496 p.

Ravikumar, K.M., Rajagopal, R., Nagaraj, H.C. 2009. An approach for objective assessment of stuttered speech using MFCC features. ICGST International Journal on Digital Signal Processing, DSP. 9, 19–24.

Ravikumar, K.M., Reddy, B.A., Rajagopal, R., Nagaraj, H.C. 2008. Automatic detection of syllable repetition in read speech for objective assessment of stuttered Disfluencies. In: Proceedings ofWorld Academy Science, Engineering and Technology. 270–273.

Savitzky, A., Golay, M.J.E. 1964. Smoothing and differentiation of data by simplified least squares procedures // Anal. Chem. 36, 1627–1639.

SciPy.org. https://docs.scipy.org/doc/scipy0.15.1/reference/generated/scipy.signal. savgol_filter.html.

Wiedecke, B., Narzillo, M., Payazov, M., Abdurashid, S. 2019. Acoustic signal analysis and identification, Int. J. Innov. Technol. Explor. Eng., 8, 2440–2442.

Wu, Q.Z., Jou, I.C., Lee, S.Y. 1997. On-line signature verification using LPC cepstrum and neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 27, 148–153.