Speaker Diarization through Waveform and Neural Net

Rustam Latypov, Evgeni Stolov
Kazan Federal University
Kazan, Russia
Roustam.Latypov@kpfu.ru

***Abstract*—This paper presents an approach to the speaker diarization problem based on speech local waveform analysis. We assume that the recorded sound scene consists of a known number of sources and that the single microphone is utilized for recording. The research goal is to develop an algorithm for speaker diarization in online mode. The most significant attention is paid to limiting computer resources when solving the problem. We suppose that the speech file is already segmented so that any segment belongs to a single speaker. Our method is as follows. We divide each part into non-overlapping fragments of the constant length and change any sample in the piece to its absolute value. A particular technique is used to choose a threshold value *Thr*. After that, we select the portions of the fragments that exceed *Thr* and implement coding to describe the source signal's revealed parts as normalized cumulative sums containing the same number of items. These sums are used as input vectors for two types of neural networks. For comparison, we also developed a simple algorithm that does not leverage the neural net but fits the problem. The experiment shows that the end-to-end neural classification of the fragments brings acceptable results.**

I. Introduction

Diarization is fulfilled by corresponding a segment of speech into a space representing the speaker's features and then clustering. A fundamental problem is how to do this conformity. In recent research, neural nets are used for this mapping, guiding vital improvements in diarization correctness. Speaker diarization has applications in many critical scenarios, such as:

- During a conference, where the camera must be directed to the active speaker.
- Creation subtitles for audio recordings of productions.
- Implementation of unique tools for enhancing the quality of some speaker's speeches.

There are different requirements for the leveraged algorithms depending on the area of application. Production of subtitles is possible in offline mode, whereas conference services suppose work online. The specificity of the diarization problem is a small number of speakers. However, the database with speech templates of the presented speakers in the list is missing, and it should be created very fast before the meeting's beginning.

The speaker diarization problem consists of segmenting a discussion involving multiple speakers into homogeneous pieces which contain the voice of only one speaker, labeling the obtained segment, and grouping together all the portions that correspond to the same speaker. The first part of the process is also called speaker change detection, while the second one is known as the clustering process. After the labeling segmentation is realized, the next step is to extract features of the fragment for the following clusterization. The papers dedicated to the diarization problem vary in these aspects. In this paper, we consider the latter task, the task of clusterization.

Let us consider the different kinds of features extracted from fragments. The probabilistic approach in earlier work consists of the following. All samples are considered random values, so an approximation of PDF (probability density function) is constructed. Investigating the local maximum of PDF provides an estimation of the unknown number of speakers but requires significant calculation [1]. A review of the diarization methods, supported by the probabilistic model, is collected in [2]. The most attention is dedicated to the technique based on Hidden Markov Model (HMM). Usage of Mel-Frequency Cepstral Coefficients (MFCC) of a fragment as a feature is a natural solution when investigating the diarization problem [3]. The further efforts were directed to extract information from speech files that can be seen as a hash vector, which can not be described in purely probabilistic terms: i-vector [4], [5], d-vector [6], and others. The authors use the notion of the intrinsic dimension of the set of random values where any value can be produced by a function of the restricted number of parameters. The direct obtaining of the mentioned parameters supposes calculation eigenvectors of big size matrices, and this procedure needs significant resources. Later, a neural net is used to this end. A graph, where various fragments of segments are nodes, is used for labeling [4]. The d-vectors are the primary feature of fragments leveraged for the creation of an edge between two nodes.

From our point of view, the form of signal-based methods is of particular interest. Diarization on the base of analysis of the signal spectrum is given in [5]. Studies based on the straight waveform investigation are recently proposed in [6] and [7]. In [6], a novel convolutional architecture, *SincNet* developed. The approach in [7] uses *SincNet* to obtain distinguishing speaker peculiarities. In [8], the analysis of waveform was allied for distinguishing the language of the speaker. The wave is approximated employing a pair of parabolas. The parameters of the parabolas can be seen as a compressed description of the chunk. In this paper, we also developed a method of diarization

based on direct extraction features of the wave by a particular algorithm.

The next step in the diarization task is distributing obtained features among a given number of clusters. In earlier works, standard methods, based on the probabilistic notion, were leveraged, such as the Gaussian Mixed Model, the Cross Likelihood Ratio, and others [2]. Recently, employment neural nets for clustering is the most popular approach. Since speech file is a series, the recurrent nets are the most natural instrument for investigation. Those are Long Short-Term Memory (LTSM) nets [9], [3], and the more complicated graph neural net [4]. As a rule, the networks are working in an end-to-end style. That means that the net directly defines the cluster containing the vector applied to the net's input.

The algorithms presented in this paper are intended for exploitation in online mode, basically controlling the camera directing to the active speaker. It means that all necessary calculations should be minimized. All the nets mentioned above require significant time for training if a regular computer is under hands to this end. That is why we restricted ourselves with simple structures, and all efforts are directed to the correct choice the features, having small dimensions extracted from the file. We use a local feature of signal for clustering. We choose a threshold *Thr* and create a chunk of the source signals which exceed *Thr* (see Fig.1). The completed part of the signal looks like a wave, and we have to find a standard method that is invariant to the signal's multiplication to a constant for the description of the waveform. We leveraged dataset [10] for experiments and [11] to obtain results.

The paper is organized as follows. In Section II, we elaborate on an algorithm to estimate the threshold *Thr*. In Section III, we present a description of the wave by cumulative sum, in Section IV, we present the networking in the end-to-end mode, and in Section V, we show the experiment results.

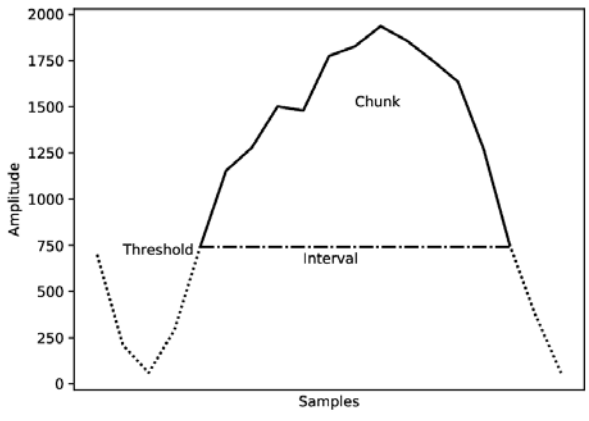

Fig.1. Wave as a part signal under investigation

II. Setting Threshold

Our program's first step is setting the thresholds *Thr* to create a step function of a fragment *Fragm*. Let *Step* be the result of the conversion of *Fragm* according to the equation (1), where *C* is a constant.

$$Step(t) = C \cdot \begin{cases} Thr, & if\ Frag(t) \geq Thr, \\ 0, & otherwise. \end{cases} \quad (1)$$

The most apparent selection criterion for *C* and *Thr* is

$$\|Frag\| = \|Step\|, \quad \|Frag - Step\| \to min. \quad (2)$$

A similar problem was investigated in [12]. The speech model has considered in which the samples are random values. Assuming all samples are independent and normally distributed. The proposed solution has theoretical value, and a solution that fits the case of a discrete signal is presented in [13]. The author utilizes the standard *k-means* procedure with two clusters. That procedure produces two centroids of the clusters

$$Cent_1, Cent_2,$$

plus the recipe is setting:

$$Thr = (Cent_1 + Cent_2)/2.$$

We used a different idea for the threshold. Consider the specific case of the fragment, when $Fragm(t) = \sin(K \cdot t),\ t \in [0, \pi/K]$, where *K* is a natural number. It is shown in [14] that the optimal threshold meaning in (2) is about 0.39. This value is independent of *K*. The mentioned independence can be considered a hint that a suboptimal estimate of the threshold is independent of the fundamental frequency and can be obtained based on the signal's power features. There is a piece of practical advice concerning the choice of threshold in the general case. That is a refinement of the procedure presented in [13]. Let us create the sequence $\{\sin\left(K \cdot n \cdot \frac{\pi}{K \cdot N}\right),\ n = 0,1,\dots,N\}$, apply the *k-means* procedure with two clusters to this sequence, and then construct the threshold

$$Thr = a \cdot Cent_1 + (1-a) \cdot Cent_2, \quad Cent_1 < Cent_2, 0 < a < 1. \quad (3)$$

Setting *Thr=0.39* in (3), one gets the value of *a*. After rounding to 2 digits after the decimal point, one gets the exact value *a*=0.83 for various *N*>100 and *K*<30.

The next significant quantity associated with the found threshold for the Sine is the quality of approximation of this function by a step-function corresponding to the optimal threshold. That is the SNR calculated via (4) and equal to 10.6 dB.

$$SNR = \log_{10}\left(\frac{\sigma^2(Fragm)}{\sigma^2(Frag - Step)}\right). \quad (4)$$

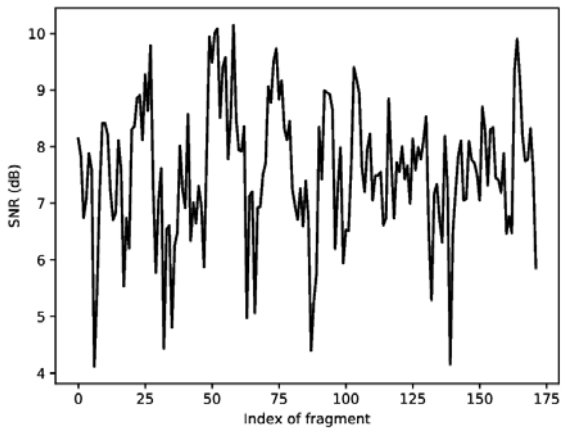

Fig.2. Suboptimal SNRs for various chunks in the file

This value can be seen as a baseline for SNRs, which can be gained for an arbitrary chunk of speech file. The distribution of SNRs for different fragments varies from one fragment to another (see Fig.2), although it does not reach the limit of 10.5 dB. To make the approximation quality more transparent, we created Table I, where medians of SNRs calculated overall speech files' fragments are placed. We leveraged four files belonging to two male and two female speakers. It follows from the table that the obtained values mainly depend on the length of the fragment but not on file choice .

Since our goal is to develop an algorithm for fast diarization, we propose a method for evaluation *Thr* that does not require the implementation *k-means* procedure. Let *Fragm* be a fragment of audio file and *AFragm* be a vector composed of absolute values of items in *Fragm*. Let $Mx = \max(AFragm)$ and $Std = std(AFragm)$ - standard deviation. We selected *Arg* = *Std*/*Mx* and used linear regression for obtaining coefficients in (5)

$$Thr = Mx \cdot (U \cdot Arg + V), \qquad (5)$$

where the left part in (5) is calculated according to (3). These coefficients depend on sample frequency and length of fragments and slightly depend on the speech file choice. It means that by obtaining the coefficients for one time, one can apply them for establishing thresholds in otter files written with the same sampling frequency. For example. We use the case of a sampling frequency of 16 kHz and a fragment length of 512, plus

$$Thr \approx Mx \cdot \left(1.2491 \cdot \frac{Std}{Mx} - 0.05047\right). \qquad (6)$$

In Fig.3, *Thr* values obtained by full search with step 20 and by linear regression (6) are shown. One can see that the values differ very slightly. Comparing SNRs, calculated by (4) based on those thresholds, one cannot distinguish the graphs and do not present the corresponding figures.

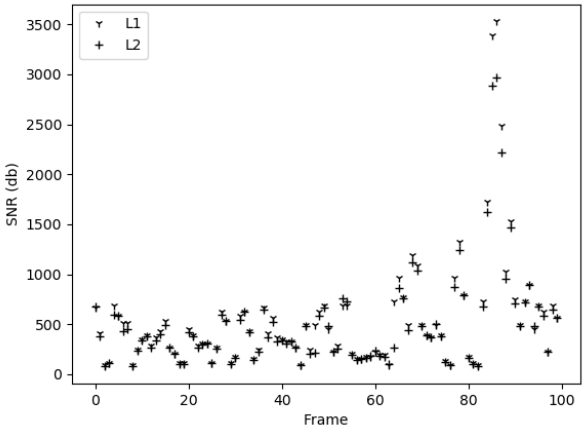

Fig.3. Compare optimal thresholds computed by full search and (5); sample frequency = 16 kHz, length of fragment = 512. L1 - full search for Thr with step = 20, L2 - linear regression

TABLE I. MEDIANS OF SNRS (IN DB) DEPENDING ON THE LENGTH OF FRAGMENTS

Speaker	Length of fragment			
	128	256	512	1024
Sp1(m)	7.8	7.6	7.2	6.7
Sp2(m)	7.3	7.1	6.8	6.3
Sp3(f)	7.5	7.3	7.0	6.5
Sp4(f)	7.8	7.6	7.3	6.9

III. WAVEFORM AND ITS DESCRIPTION

Let *Step* be a result of converting the source fragment *Fragm* into a step-function and letting the chunk *Chunk* be the part of *Fragm* corresponding to the interval *Inter* of nonzero values of *Step* (see Fig.1). We associate a new function *Wave* with any chunk, which has the interval *Inter* as a definition domain, $Wave(t) = Fragm(t) - Thr, t \in Inter$. The *Wave* description must be invariant when changing *Wave* with $C \cdot Wave$ for any positive constant *C*. Let $CumWave(t)$ be a result of converting *Wave* into cumulative function. Dividing *CumWave* by its last item gives a normalized version of this function. It is a PDF analog of a random value and its distribution function. We keep the previous designations for the new function.

Direct leveraging *CumWave* as a hash vector of a chunk for further comparison is impossible since the intervals *Inter* can have different lengths for chunks in the fragment.

Using linear interpolation, we convert all chunks to ones having the same length *L* of the intervals *Inter*. Let *M* be the length of a given *Inter* and $Args = < a_0, a_1, a_2, \dots, a_L >$ where $a_0, a_1, \dots, a_L$ are uniformly distributed throughout the interval of length *M*. The new *NewWave* function is defined as

$$NewWave = interpolate(Args, Inter, Wave).$$

The cumulative function *CumNewWave* of the length *M*, built with *NewWave*, is the given chunk's hash vector. That is a function of *M* and *CumWave*:

$$CumNewWave = Fun(M, CumWave). \quad (7)$$

The selected length *M* of the interval for aligning hash vectors plays a vital role in all the algorithms below. This parameter depends on the sampling frequency. All the following data relate to files written with the frequency of 16000 kHz. If the interval's actual length significantly exceeds the prescribed length *L*, some essential features of the waveform can be lost. For the experiment, we selected two files of the same size, belonging to men and women, and created a histogram of the lengths. We fixed the prescribed bounds of bins: 0,5,10,15,20,25,30,35 and obtained the number of items inside each bin. The histogram is presented in Fig.4. One can see that most lengths do not exceed ten samples, so the universal *L* must be close to 10.

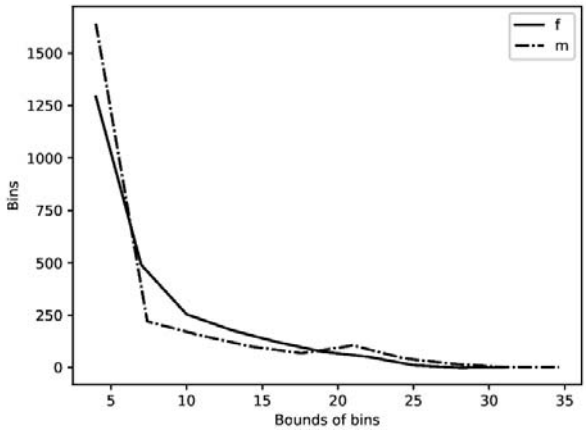

Fig.4. Distribution of intervals' lengths in step-functions; 'm' – male speaker, 'f' – female speaker

IV. Diarization Algorithms

In this section, we consider three versions of diarization based on the investigation of the waveform. We divide the source file into fragments of the length *FragmLen*, obtain the threshold *Thr* for each fragment and form a list *ListWave* containing all *CumNewWave* functions produced according to (7). Using the *ListWave* as source data, we create some algorithms for speaker diarization.

A. Shallow neural net

We use the *Keras* package for building all the neural nets in this section [15].

Algorithm 1 Diarization through the shallow neural net

Step 1. Choose the universal length *L*. For any chunk with interval length *M* aligns its length to *L* and creates *CumNewWave* according to (7).

Step 2. Add to *each CumNewWave* ratio *M/L* and create the vector *ExtdVect* of the length *L+1*.

Step 3. Create neural net

$$inputs = keras.Input(shape = (L+1,))$$
$$x = layers.Dense(256, activation = 'relu')(inputs)$$
$$x = layers.Dropout(0.5)(x)$$
$$x = layers.Dense(128, activation = 'relu')(x)$$
$$x = layers.Dense(8, activation = 'relu')(x)$$
$$outputs = layers.Dense(1, activation = 'sigmoid')(x)$$
$$model = keras.Model(inputs = inputs, outputs = outputs)$$
$$model.compile(optimizer = 'rmsprop', loss = 'binary - crossentropy', metrics = 'acc')$$

Step 4. Training the net. The net is intended for processing the conversation of two persons. Correspond to *ExtdVect* 0 or 1, depending on the speaker. The training algorithm ignores the order of chunks in the source signal used for training. A permutation of data is leveraged before the training procedure.

Step 5. Applying any *ExtdVect* from the file belonging to one speaker to neural net inputs, one obtains a series of *OutValues*. Comparing the mean value of the obtaining responses with 0.5, one determines the active speaker.

B. Convolutional neural net

We create the input data as above. While building the net, we use the *Conv1D* layer. The structure of the net is as follows:

$$inputs = keras.Input(shape = (Batch, L+1))$$
$$x = layers.Conv1D(filters = 64, 3, activation = relu')(inputs)$$
$$x = layers.MaxPooling1D(pool - size = 2)(x)$$
$$x = layers.Conv1D(filters = 64, 3, activation = relu')(x)$$
$$x = layers.MaxPooling1D(pool - size = 2)(x)$$
$$x = layers.Dropout(0.4)(x)$$
$$x = layers.Flatten()(x)$$

$$x = layers.Dense(80, activation = 'relu')(x)$$
$$outputs = layers.Dense(Num, activation = 'softmax')(x)$$
$$model = keras.Model(inputs = inputs, outputs = outputs)$$
$$model.compile(optimizer = 'rmsprop', loss = 'categorical - crossentropy', metrics = 'acc')$$

Here *Num* is the number of speakers under observation, and *Batch* is the size of the batch of *ExtdVec* on net input. The prediction is organized by the detection of the position of maximal argument in the output vector.

C. 'Naive' approach

We suppose that the training of net is performed before the beginning of a conversation. Although the presented nets have small sizes, their training requires time. At this point, we

develop another approach to the diarization. The hash vector, used for describing an audio file fragment, is based on waveforms' direct employment. According to Fig.4, the lengths of chunk intervals are distributed non uniformly. What more, the structure of wave can depend on the length of the chunk. That is the property we leveraged in our approach. As before, the source file is the list *ListWave*. Choose four bounds $\mathrm{Lim}_0, \mathrm{Lim}_1, \mathrm{Lim}_2, \mathrm{Lim}_3$, three lengths $\mathrm{Len}_1 > \mathrm{Lim}_1, \mathrm{Len}_2 > \mathrm{Lim}_2, \mathrm{Len}_3 > \mathrm{Lim}_3$ and create three lists: $\mathrm{List}_1, \mathrm{List}_2, \mathrm{List}_3$. According to Algorithm 2, one creates a hash vector for any file belonging to a speaker. The algorithm's critical feature is equality of size of all hash vectors independent of the source file. So we can compare two hashes using the regular Euclidean distance. Before the beginning of the conversation, one creates template hashes for each speaker. During the conversation, one compares the current hash vector with template ones and determines the speaker by the template hash nearest to the current hash.

Algorithm 2 Creating hash vectors

for $CumWave \in ListWave$ **do**
  $Ln = length(CumWave)$
  $List_0.append(Ln)$
  if $Ln \geq Lim_0 \& Ln < Lim_1$ **then**
    $CumNewWave \leftarrow CumWave, Len_1$ {Aligning to length Len_1 (7)}
    $List_1.append(CumNewWave)$
  else if $Ln < Lim_2$ **then**
    $CumNewWave \leftarrow CumWave, Len_2$
    $List_2.append(CumNewWave)$
  else
    $CumNewWave \leftarrow CumWave, Len_3$
    $List_3.append(CumNewWave)$
  end if
end for
$Vect = \langle histogram(List_0), sum(List_1), sum(List_2), sum(List_3) \rangle$
$Hash \leftarrow Vect$ {Normalization}

V. Experiments

In this section, we present some results of experiments with data in the dataset [10]. We loaded four files: Bed003, Bed004, Bed005, Bed006. The files are written with a 16kHz sample frequency. We selected just those fragments marked 'segment', which means that the fragment contains a single person's speech. What more, we rejected all the fragments having a duration of less than 2 sec. For each speaker, we selected a series of files with 2 sec and used the training. We divided all other selected fragments into files duration about 750 msec. That is the dataset used in the experiments.

A. Experiments with shallow neural net

As usual, we evaluate the net's quality by figures containing graphs of accuracy and validation accuracy produced during the training. It follows from Fig.5 that the quality of diarization gained by the shallow net is of the middle level. We trained the net using two files belonging to two persons and then processed all other files in the dataset belonging to those speakers. The results are presented in Table II. The form *A/B* records mean the number of correct recognized files (*A*) to the total number of the files belonging to the speaker (*B*).

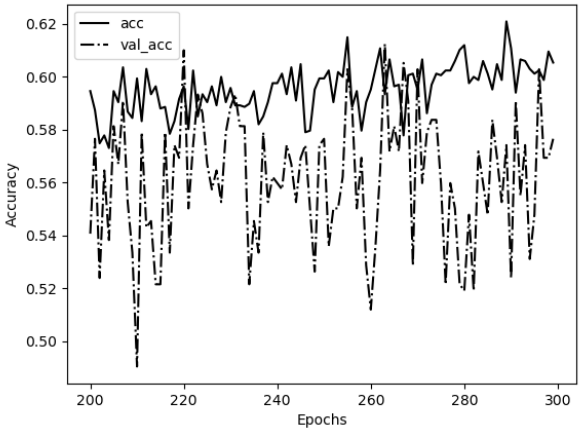

Fig.5. Accuracy and validation accuracy graphs produced during training shallow nets for two persons

TABLE II. RESULTS OF DIARIZATION OF TWO PERSONS BY SHALLOW NET

Speakers	The first	The second
Sp1,Sp2	$^{21}/_{36}$	$^{20}/_{32}$
Sp1,Sp3	$^{20}/_{36}$	$^{23}/_{34}$
Sp2,Sp3	$^{18}/_{32}$	$^{19}/_{34}$

All attempts to use this net to diarize more than two persons have failed. For example, in Sp1, Sp2, Sp3 speakers, we get the values $^{8}/_{36}, ^{12}/_{32}, ^{34}/_{24}$. Most of the speakers were marked as Sp3.

B. Experiments with convolutional net

It follows from Fig.6 that the results of training are more successful comparing to ones for the shallow net. Some results are placed in Table III. The results of the diarization of three persons are presented in Table IV.

TABLE III. RESULTS OF THE DIARIZATION OF TWO PERSONS BY CONVOLUTIONAL NET

Speakers	The first	The second
Sp1,Sp2	$^{36}/_{36}$	$^{24}/_{32}$
Sp4,Sp2	$^{55}/_{61}$	$^{28}/_{32}$
Sp4,Sp3	$^{61}/_{61}$	$^{32}/_{34}$
Sp1,Sp5	$^{31}/_{36}$	$^{28}/_{35}$

The authors of [9] claim percent errors in the range of 11%-15% in the case of two speakers. These figures are adjusted with our calculations, although the resources used for training are significantly less. One can suggest that the quality of the diarization depends on the files' choice, and our estimates support that statement.

TABLE IV. RESULTS OF THE DIARIZATION OF THREE PERSONS BY CONVOLUTIONAL NET

Speakers	The first	The second	The third
Sp5,Sp7,Sp3	$^{40}/_{40}$	$^{35}/_{35}$	$^{17}/_{34}$
Sp5,Sp1,Sp4	$^{36}/_{40}$	$^{33}/_{36}$	$^{61}/_{61}$
Sp8,Sp1,Sp6	$^{23}/_{34}$	$^{28}/_{36}$	$^{39}/_{41}$
Sp5,Sp8,Sp2	$^{40}/_{40}$	$^{19}/_{34}$	$^{23}/_{32}$

C. *Experiments with a 'naive' approach*

The result of the experiment is shown in Table V. If we compare with the results in Table II, we can see that this method exceeds the method based on the shallow net but requires less calculation. On the other hand, our experiments show that the naive method does not provide a possibility for the diarization of more than two persons.

TABLE V. RESULTS OF DIARIZATION OF TWO PERSONS BY 'NAIVE' APPROACH

Speakers	The first	The second
Sp1,Sp2	$^{22}/_{36}$	$^{23}/_{32}$
Sp1,Sp3	$^{21}/_{36}$	$^{26}/_{34}$
Sp2,Sp3	$^{19}/_{32}$	$^{25}/_{34}$

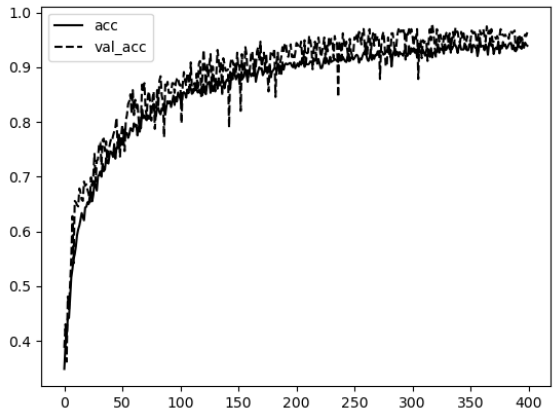

Fig.6. Accuracy and validation accuracy graphs produced during training convolutional nets for two persons

VI. CONCLUSION

It follows from our experiments that there are reasons for the selection method for diarization depending on the number of persons under processing, where available computing resources are restricted. The shallow net does not fit the problem. It is a pity, but the quality of the diarization depends on the speakers taking part in the conversation. If we are dealing with two persons, we start with the 'naïve' method since it can be tested very quickly. Whereas, for the case with the number of persons more than 2, a recurrent net method must be implied. Depending on the situation, some additional features must be included in the network model.

ACKNOWLEDGMENT

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University. This work is also supported by the Russian Science Foundation grant № 19-18-00202.

REFERENCES

[1] K. Fukunaga and L. D. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," in *IEEE Trans. on Inform. Theory*.IEEE, 1975, vol. rr-21, pp. 32–40.

[2] S. Tranter and D. Reynolds, "An overview of automatic speaker diarization systems," in *IEEE Trans. Audio, Speech, and Lang. Proc.* IEEE, 2006, vol. 14, pp. 1557–1565.

[3] L.Sun et al., "A novel LSTM-based speech preprocessor for speaker diarization in realistic mismatch conditions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5234–5238.

[4] J. Wang et al., "Speaker diarization with session level speaker embedding refinement using graph neural networks," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.IEEE, 2020, pp. 7109–7113.

[5] Q. Wang et al., "Voice filter: Targeted voice separation by speaker-conditioned spectrogram masking," in *Proc. Interspeech 2019*. IEEE, 2019, pp .2728-2732.

[6] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *Proc. Spoken Lang. Technol. Workshop*. IEEE, 2018, pp. 1021–1028.

[7] H. Dubey, A. Sangwan, and J.H. Hansen, "Transfer learning using raw waveform sincnet for robust speaker diarization," in *Proc. ICASSP 2019* . IEEE, 2019, pp. 6296–6300.

[8] R. Latypov, R. Nigmatullin, and E. Stolov, "Classification of speech files by waveforms," *Lobachevskii Journal of Math.*, vol. 36, 2015, pp. 496–502.

[9] Q. Wang, C. Downey, L. Wan, P. Mansfield, and I. Moreno, "Speaker diarization with LSTM," in *Proc. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,. IEEE, 2018, pp 5539-5543.

[10] ICSI corpus, "The meeting recorder project," Web:http://groups.inf.ed.ac.uk/ami/icsi/, 2006.

[11] F. Pedregosa et al., "Scikit-learn: machine learning in python," *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825–2830.

[12] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. on Inform. Theory*. IEEE, 1982, vol. IT-28, pp. 129–137.

[13] B. Girod, "Image and video compression," Web:https://web.stanford.edu/class/ee398a/handouts/lectures/05-Quantization.pdf, 2005.

[14] R. Latypov and E. Stolov, "Speaker diarization based on speech signal approximation by step-function," in *Proc. 28th Conference of Open Innovations Association(FRUCT)*, 2021, pp. 598-604.

[15] F. Chollet et al., "Keras," Web: https://github.com/fchollet/keras, 2015.