Audio Engineering Society

Conference Paper

Presented at the International Conference on
Audio Forensics
2019 June 18–20, Porto, Portugal

This paper was peer-reviewed as a complete manuscript for presentation at this conference. This paper is available in the AES E-Library (http://www.aes.org/e-lib) all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Deep neural network based forensic automatic speaker recognition in VOCALISE using x-vectors

Finnian Kelly[1], Oscar Forth[1], Samuel Kent[1], Linda Gerlach[2], and Anil Alexander[1]

[1]*Oxford Wave Research Ltd., Oxford, United Kingdom.*

[2]*Philipps-Universität Marburg, Germany.*

Correspondence should be addressed to Author (finnian@oxfordwaveresearch.com)

ABSTRACT

In this article we present a Deep Neural Network (DNN)-based version of the VOCALISE (Voice Comparison and Analysis of the Likelihood of Speech Evidence) forensic automatic speaker recognition system. DNNs mark a new phase in the evolution of automatic speaker recognition technology, providing a powerful framework for extracting highly-discriminative speaker-specific features from a recording of speech. The latest version of VOCALISE aims to preserve the ‘open-box’ philosophy of its predecessors, offering the forensic practitioner flexibility in the configuration and training of all parts of the automatic speaker recognition pipeline. VOCALISE continues to support both legacy and state-of-the-art speaker modelling algorithms, the latest of which is a DNN-based ‘x-vector’ framework, a state-of-the-art approach that leverages a DNN to extract compact speaker representations. Here, we introduce the x-vector framework and its implementation in VOCALISE, and demonstrate its powerful performance capabilities on some forensically relevant data.

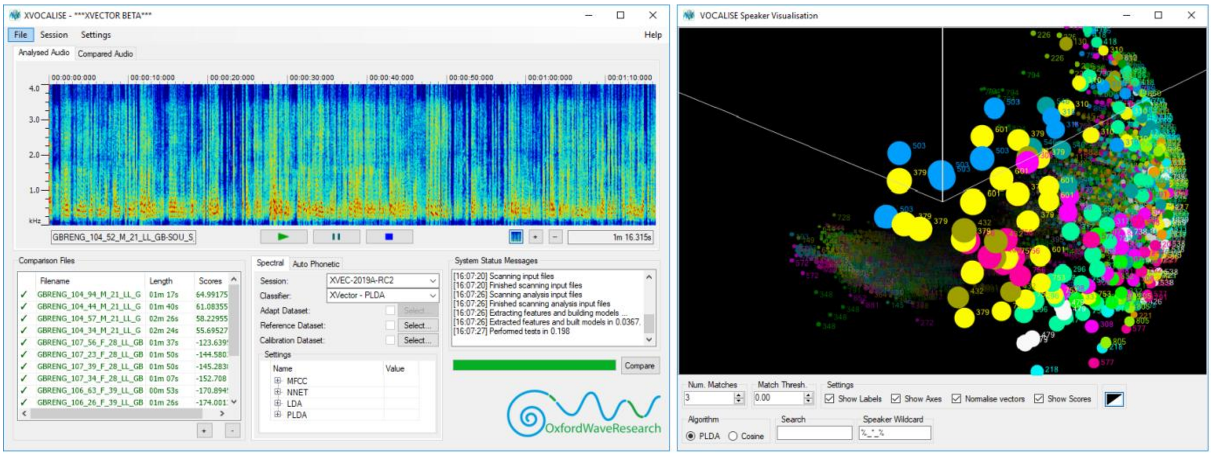
1 Introduction

VOCALISE [1] is an automatic speaker recognition system that allows the user to perform speaker comparisons using a variety of different features and algorithms in a flexible way. Coupled with the accompanying Bio-Metrics likelihood ratio and performance metrics software, VOCALISE enables the forensic practitioner to interpret the results of an automatic speaker comparison within a likelihood-ratio framework [2].

The steps involved in this process include the extraction of speaker-specific features from recordings of speech, the creation of speaker models using features, the comparison of speaker models to produce a score, and the evaluation of likelihood-ratios from the score distributions of same-speaker and different-speaker comparisons [2].

VOCALISE is built with an ‘open-box’ architecture, offering the user a choice of several feature extraction, speaker modelling, and speaker comparison approaches, and allowing them to introduce their own speech recordings at various training stages of the speaker comparison pipeline.

The open-box architecture further allows the user to tailor the system to their problem domain, by adapting the pre-trained models provided with VOCALISE, or by training their own custom models ‘from scratch’.



GBRENG_104_52_M_21_LL_GB-SOU_S

1m 16.315s

Comparison Files

	Filename	Length	Scores
✓	GBRENG_104_94_M_21_LL_G	01m 17s	64.99175
✓	GBRENG_104_44_M_21_LL_G	01m 40s	61.08355
✓	GBRENG_104_57_M_21_LL_G	02m 26s	58.22955
✓	GBRENG_104_34_M_21_LL_G	02m 24s	55.69527
✓	GBRENG_107_56_F_28_LL_GB	01m 37s	-123.639
✓	GBRENG_107_23_F_28_LL_GB	01m 50s	-144.580
✓	GBRENG_107_39_F_28_LL_GB	01m 50s	-145.283
✓	GBRENG_107_34_F_28_LL_GB	01m 07s	-152.708
✓	GBRENG_106_63_F_39_LL_GB	00m 53s	-170.894
✓	GBRENG_106_26_F_39_LL_GB	01m 26s	-174.001

Session: XVEC-2019A-RC2
Classifier: XVector - PLDA
Adapt Dataset: □
Reference Dataset: □
Calibration Dataset: □

Settings

Name	Value
⊞ MFCC	
⊞ NNET	
⊞ LDA	
⊞ PLDA	

System Status Messages

```
[16:07:20] Scanning input files
[16:07:20] Finished scanning input files
[16:07:26] Scanning analysis input files
[16:07:26] Finished scanning analysis input files
[16:07:26] Extracting features and building models ...
[16:07:26] Extracted features and built models in 0.0367.
[16:07:27] Performed tests in 0.198
```

Num. Matches: 3
Match Thresh.: 0.00
Settings: ☑ Show Labels ☑ Show Axes ☑ Normalise vectors ☑ Show Scores
Algorithm: ● PLDA ○ Cosine
Search:
Speaker Wildcard: %_*_%

Figure 1: The VOCALISE main interface running an x-vector comparison (left), and the VOCALISE interactive x-vector visualisation (right).

The architecture also helps to bridge the gap between traditional forensic phonetics and automatic speaker recognition, by supporting fully automatic speaker comparisons using phonetic features.

The first version of VOCALISE [3], developed in 2012, provided a speaker recognition pipeline based on Mel-frequency cepstral coefficients (MFCCs) and Gaussian Mixture Models (GMMs). Development has been ongoing since, with the inclusion of phonetic features, namely automatically extracted formant measurements in 2013 [4], and the expansion of its speaker modelling capabilities with the release of the i-vector based version (also known as iVOCALISE) in 2016 [1]. The latest version of VOCALISE (also known as xVOCALISE), further expands the speaker modelling pipeline with the addition of an x-vector-based framework. Figure 1 shows the latest VOCALISE interface and its speaker visualisation tool. Over the years, the development has benefitted from the support and collaboration of the German Bundeskriminalamt (BKA), the Netherlands Forensic Institute (NFI), as well as the UK Ministry of Defence.

In the following sections of this article we introduce the new VOCALISE x-vector framework. Section 2 provides an overview of x-vectors within the automatic speaker recognition pipeline. Section 3 provides some specific implementation details. Finally, Section 4 highlights the performance of the VOCALISE x-vector framework using some forensically relevant data.

2 Automatic Speaker Recognition: The x-vector framework

The purpose of an automatic speaker recognition system is to compare two recordings of speech and return a score. The score can be used to make decisions about whether the same speaker is the source of the speech on both recordings, or in a forensic context, can be used to estimate a likelihood-ratio (LR) under the same-speaker and different-speaker hypotheses [2]. Figure 2 illustrates LR estimation using Bio-Metrics.

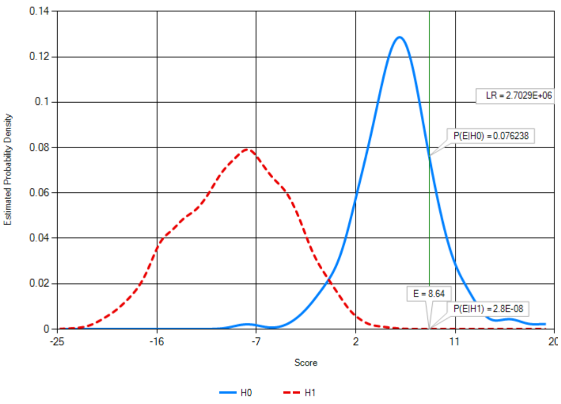


Figure 2: Estimating the LR in Bio-Metrics from the same-speaker (*H0*, blue curve) and different-speaker (*H1*, red curve) score distributions for a given score (*E*, green line).

Automatic speaker recognition technology has evolved steadily over the last three decades, accompanied by continuous improvements in recognition accuracy [5, 6]. The core processes involved in the automatic speaker recognition pipeline remain the same however. The first step is to extract the information most salient to speaker recognition from the recording of speech. This ‘feature extraction’ process is followed by speaker modelling, or speaker representation, which uses a set of features to create a general representation of the speaker. The final stage is the comparison of two speaker representations resulting in a score. This general pipeline is summarised in Figure 3.

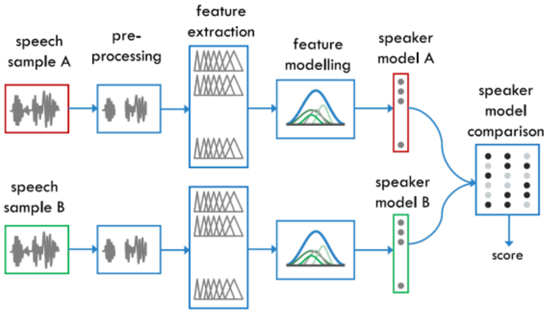


Figure 3: An automatic speaker recognition pipeline

A recent promising proposal for incorporating DNNs into the speaker recognition pipeline is the x-vector framework [7]. The following sections detail the core components of the VOCALISE x-vector framework, drawing comparisons with the existing state-of-the-art i-vector approach, and earlier approaches based on GMMs.

2.1 Feature Extraction

The feature extraction process converts a recording of speech into a set of features that are effective for speaker discrimination. Effective features have high between-speaker variability and low within-speaker variability, occur frequently and naturally in 'normal' speech, and are not strongly affected by noise or distortion.

MFCCs [8] have been the dominant feature in automatic speaker recognition technology since the early days of the field, and remain so in state-of-the-art systems today. MFCC extraction involves measuring the short-term power spectrum of speech over short overlapping windows throughout the recording, and applying a weighting according to the Mel scale, which mirrors the scale of human hearing. VOCALISE supports MFCC extraction, with flexibility in the frequency range for analysis and selection of the number of Mel filterbanks and the number of coefficients.

Following extraction, MFCC features can be processed in several ways with VOCALISE. To incorporate temporal information, MFCCs can be appended with their first and second derivatives (often referred to as delta and delta-delta coefficients) calculated across several adjacent feature frames [9]. Cepstral Mean Subtraction (CMS) [10] is a way of removing convolutional noise, such as the effect of channel. This can be extended to Cepstral Mean and Variance Normalisation (CMVN). An important final stage of feature processing is to discard any feature frames corresponding to regions of silence and non-speech in the speech recording. This process, referred to as Voice Activity Detection (VAD), discriminates between speech and non-speech based over short windows of the original speech recording.

In traditional forensic phonetics, estimations relating to fundamental frequency and formants have played a key role in making judgements about the speaker's identity. To bridge the gap between this form of traditional analysis and automatic analysis, VOCALISE also supports automatic extraction of such phonetic features. The 'auto-phonetic' features currently offered consist of any combination of formants F1 to F4. Other auto-phonetic features, such as pitch, can be included in certain set-ups by modifying an external Praat [11] script. In this paper, the focus will be on MFCC features; however, the full VOCALISE pipeline can also operate with auto-phonetic features.

2.2 Speaker Modelling

The next stage of the automatic speaker recognition pipeline is to create a speaker model, or speaker representation, given a set of features. Progress in automatic speaker recognition performance has been driven primarily by advances in the speaker modelling part of the pipeline [5, 6]. VOCALISE supports 'classical' approaches to speaker modelling based on GMMs [12], as well as recent approaches based on i-vectors [13] and x-vectors [7].

This paper focuses on the state-of-the-art x-vector and i-vector operating modes of VOCALISE. The x-vector and i-vector approaches are conceptually similar, in that they both use a set of features, together with some pre-trained models, to extract a compact, fixed-size representation of a speaker (x-vectors typically contain 512 values; i-vectors typically contain 400 values). These new speaker representations can be directly compared to obtain a score. This form of speaker representation sets x-vectors and i-vectors apart from their GMM predecessors, which rely on models of feature distributions to represent speakers. In the terminology of DNNs, these compact vector speaker representations are referred to as speaker embeddings. The key difference between the i-vector and x-vector approaches lies in the way in which the speaker embedding is extracted.

2.2.1 The x-vector framework

In the VOCALISE x-vector framework, the set of MFCC features[1] from a speech recording is processed to extract an x-vector [7], a compact, fixed-size vector that captures speaker-specific

[1] This framework is equally applicable to VOCALISE auto-phonetic features.

information. The x-vector is of fixed-size, regardless of the number of MFCC frames used for extraction (determined by length of the speech recording).

Obtaining an x-vector for a new speech recording requires an x-vector extractor based on an artificial neural network that was trained using MFCCs from a large, diverse set of training recordings.

Artificial neural networks are collections of connected units or nodes, each of which loosely models the behaviour of the neurons in the brain. Typically, each node in the network receives one or more inputs, and outputs a weighted sum of these inputs. Training an artificial neural network involves adjusting the weight assigned to each node so that errors are minimised on some training data (i.e., a set of inputs and outputs). *Deep* Neural Networks are artificial neural networks with multiple layers of connected nodes. This deep extension allows representations of data to be learned at multiple levels of abstraction, enabling complex relationships to be modelled [14].

The VOCALISE x-vector extractor adopts the architecture in [7] – a feed-forward DNN comprised of nine layers in total. A schematic of the architecture is provided in Figure 4.

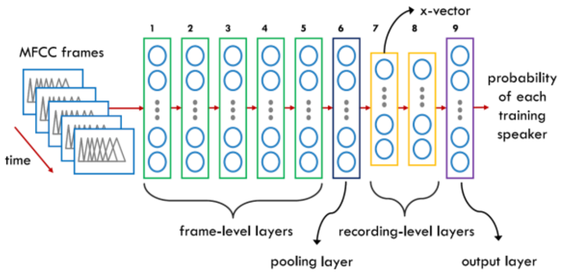


Figure 4: Schematic of the VOCALISE DNN architecture for x-vector extraction

The first five layers model both static and temporal characteristics of the input MFCC frames. The first layer takes the MFCC frame at time *t* as input, along with a small temporal context (i.e., a small number of additional MFCC frames centred on the current frame *t*). The second layer takes the output of the first layer as its input, and again includes a small temporal context, relative to the first layer. This process repeats through the five frame-level layers. Layers one to five operate at the frame-level, in the sense that they produce one set of outputs for every MFCC frame.

The output of the fifth layer is passed to the statistics pooling layer, which calculates the mean and standard deviation of the layer five outputs for the entire input speech recording. The pooling layer is a recording-level layer, in the sense that it produces just one set of outputs for every recording (regardless of duration).

These recording-level statistics are then passed through two lower-dimensional layers, before the final ‘softmax’ output layer, which has one output for every training speaker.

The purpose of the network is to provide a speaker embedding for an input speech recording of arbitrary duration. Either of the recording-level layers seven and eight are suitable for this purpose. Note that the final output layer nine is specific to the speakers used to train the network – since the intention is to use the system with *new* speakers, it is more effective to use the output of the more general layers seven and eight than the speaker-specific output of layer nine. In VOCALISE, the output of layer seven is taken as the speaker embedding, i.e., the x-vector.

The DNN is trained by extracting MFCCs from a large, diverse set of speech recordings and providing them as input to the network, along with their associated speaker labels at the output. To extract an x-vector for a new speech recording, a set of MFCCs and a trained DNN are required. DNN training is a data-hungry process, requiring tens of thousands of recordings from thousands of speakers.

2.2.2 The i-vector framework

In the VOCALISE i-vector framework, a set of MFCC features[1] is used to extract an i-vector [13], a compact vector that captures speaker-specific information, in a similar way to an x-vector.

Obtaining an i-vector for a new speech recording requires an i-vector extractor trained using MFCCs from a large, diverse set of training recordings. To train the extractor, MFCCs from the training recordings are pooled together and used to train a GMM with a large number of components, referred to as a Universal Background Model (UBM) [12]. Next, speaker models are generated for each of the training recordings by adaptation of the UBM toward each of the training recordings given the relevant MFCCs, resulting in a set of GMM-UBM models. This process replicates the classical GMM-UBM approach for training a speaker model [12]. The i-vector extractor builds on this classical approach by converting these large GMM-UBM speaker models into more compact and discriminative speaker embeddings. To achieve this conversion, the GMM-UBM training models are each converted into a ‘supervector’ representation [5] by stacking their mean components into a vector. The set of training supervectors is then factorised into (approximately) speaker-dependent and -independent components; the speaker-independent component (represented by the 'universal' UBM supervector), is subtracted from the training supervectors, and the remaining speaker-dependent component is then factorised into a low-rank Total Variability (TV) matrix and a set of total factors [13]. These total factors, or i-vectors, can be viewed as compact representations of GMM-UBM supervectors that capture most of the important speaker variability.

To extract an i-vector for a new speech recording, a previously-trained UBM and a TV matrix are required. As with the x-vector approach, this training process requires tens of thousands of recordings from thousands of speakers.

2.3 Comparing Speakers

In the i-vector and x-vector frameworks, speakers are represented by compact, fixed-size vectors. In both cases therefore, comparing speakers involves the comparison of two vectors of the same size, resulting in a score. The vector representation offers several advantages over GMM-based frameworks: comparisons are generally faster and are commutative, and the vectors are amenable to powerful post-processing techniques. In VOCALISE, the same algorithms for comparing speakers are applicable to both i-vector and x-vector frameworks.

Prior to comparison of i-vectors or x-vectors, it is beneficial to apply some post-processing; VOCALISE supports Linear Discriminant Analysis (LDA) [15], which is applied to enhance speaker separability, while reducing the vector dimensionality. LDA uses a set of labelled training vectors to find a subspace in which the between-speaker variability is maximised and the within-speaker variability is minimised. After projecting vectors into the new subspace, they can be compared.

In the i-vector and x-vector frameworks, the post-LDA vectors are discriminative enough that a simple distance metric can be used to produce a similarity score. VOCALISE supports cosine distance scoring [13] for this purpose.
A more powerful method of comparing speakers is Probabilistic Linear Discriminant Analysis (PLDA) [16], which leverages knowledge of the most discriminative parts of the vectors. VOCALISE supports the Gaussian variant of PLDA [17], in which vectors are length-normalised, then factorised into (approximately) speaker-dependent and -independent components. A set of labelled training vectors is used to learn the speaker-dependent subspace in which vectors are subsequently compared.

2.4 Adapting to new conditions

The recording conditions encountered in forensic casework vary widely. It is not conceivable that a commercial forensic automatic speaker recognition system will have had sight of all possible conditions, or combinations of conditions. For a previously unseen condition or variant within a condition (e.g. telephone intercept recordings saved in mp3 format) therefore, the system may not perform optimally.

VOCALISE supports adaptation of its pre-trained models with user-provided speech recordings that have similar conditions to the case under consideration. This adaptation allows for the fact that there is usually a much smaller quantity of data available to the forensic practitioner than would be required to train a speaker recognition pipeline from scratch. VOCALISE condition adaptation updates the LDA transformation with user-provided data to find a new subspace in which the within-speaker variability and *between-condition* variability (i.e. variability between original and the user-provided data) are minimised. Along with the adapted LDA model, an adapted PLDA model is trained by pooling the original and user-provided vectors after transformation with adapted LDA.

This powerful capability can provide significant improvements in performance using a relatively small number of speaker recordings in the new condition. For instance, we have observed performance improvements using recordings from just thirty or more speakers in the new condition.

3 The VOCALISE built-in x-vector and i-vector frameworks

In VOCALISE, speaker recognition models and parameters are stored as 'sessions' [1]. A session contains all of the information necessary to carry out a comparison for a given system configuration. VOCALISE is supplied with 'built-in' sessions containing pre-trained and optimised models.

Additionally, the user has the ability to create custom sessions by introducing their own data at any point in the system training pipeline (i.e., DNN, LDA, PLDA), or by training a new session from scratch [1].

This section details VOCALISE built-in x-vector and i-vector sessions in greater depth, and is followed by a sample evaluation using these sessions in Section 4.

3.1 VOCALISE built-in x-vector session

Here, we detail the VOCALISE built-in x-vector session '2019A-beta'. In this session, 22-dimensional MFCCs (including energy) are extracted over 25 ms Hamming windows with a 10 ms overlap, using 23 Mel filterbanks in the range 20 to 3,700 Hz. CMS is applied over a sliding window of 3 seconds, and silence frames are dropped according to VAD. Note that delta and delta-delta coefficients are *not* appended; the first five layers of the DNN provide sufficient temporal information.

The session is trained with a diverse set of speech recordings from several thousand speakers carefully selected from various corpora. The training set contains various channels (e.g. telephone and microphone) and multiple languages.

To expand the quantity and diversity of the training data, 'data augmentation' [18] is applied. In this procedure, copies of the training recordings are augmented with noise and reverb, before being combined with the original training set [18, 19].

The DNN architecture was the same as that in [7], with the x-vector speaker embedding taken at the output of the 512-dimensional seventh layer. The training set was used for the DNN, and LDA and PLDA models of 150 dimensions.

3.2 VOCALISE built-in i-vector session

Here, we detail the VOCALISE built-in i-vector session '2018A-Adaptable-15F'. In this session, 15-dimensional MFCCs are extracted over 32 ms Hamming windows with 50% overlap, using 24 Mel filterbanks in the range 1 to 4,000 Hz. Delta and delta-delta coefficients are appended, CMS is applied, and silence frames are dropped according to VAD.

Similar to the x-vector session, the i-vector session is trained with a diverse set of speech recordings from several thousand speakers carefully selected from various corpora.

The training set was used for training a UBM of 1024 components, a TV matrix of 400 dimensions, and LDA and PLDA models of 200 dimensions.

4 Sample experiments with forensically relevant data

In this section, we present x-vector speaker comparisons using VOCALISE and the GBR-ENG corpus[2].GBR-ENG consists of 6000 telephone recordings from 600 speakers. Each recording consists of one side of a landline or mobile telephone conversation of 3-6 minutes duration. All speech is in English, recorded across three regions in England, namely North, South and Midlands.

A set of 2134 landline recordings (from 387 speakers), and 3349 mobile recordings (from 534 speakers) were extracted for speaker comparison. Using the built-in x-vector session detailed in section 3.3, and the built-in i-vector session detailed in section 3.4, speakers were compared within and across telephone condition. The resulting Equal Error Rates (EERs) are provided in Table 1.

	Landline vs Landline	Mobile vs Mobile	Landline vs Mobile
i-vector	2.38 %	15.33 %	15.67 %
x-vector	0.94 %	1.68 %	3.30 %

Table 1: EERs for x-vector and i-vector GBR-ENG telephone comparisons.

To explore the effectiveness of VOCALISE condition adaptation (detailed in section 2.4), the GBR-ENG mobile vs mobile comparison was repeated, with the application of condition adaptation using another small subset set of GBR-ENG speakers. This adaptation set comprised of 236 recordings from 50 speakers. Note that there was no overlap between speakers or recordings in the adaptation set and the mobile recording test set (3349 recordings from 534 speakers). The application of the new condition adaptation method improved the x-vector mobile vs mobile comparison EER from 1.68% to 1.40% and the i-vector mobile vs mobile comparison EER from 15.33% to 5.80%.

5 Conclusions

The new DNN-based version of VOCALISE using x-vectors provides a powerful, flexible tool for automatic speaker recognition. It maintains an open-box philosophy and allows the forensic practitioner to interpret their speaker recognition results in a likelihood-ratio framework. Significant performance improvements are observed using the new VOCALISE x-vector framework, with further improvements observed using VOCALISE condition adaptation.

[2] GBR-ENG: A telephonic speech database collected for the UK Government for evaluating speech technologies. Further details on application.

References

[1] A. Alexander, O. Forth, A. A. Atreya and F. Kelly, *VOCALISE: A Forensic Automatic Speaker Recognition System supporting Spectral, Phonetic, and User-Provided Features*, Odyssey 2016 (2016).

[2] A. Drygajlo, M. Jessen, S. Gfroerer, I. Wagner, J. Vermeulen and T. Niemi, *Methodological Guidelines for Best Practice in Forensic Semiautomatic and Automatic Speaker Recognition, Including Guidance on the Conduct of Proficiency Testing and Collaborative Exercises*, European Network of Forensic Science Institutes (ENFSI), Wiesbaden, Germany (2015).

[3] M. Jessen, O. Forth and A. Alexander, *VOCALISE: Eine gemeinsame Plattform für die Anwendung automatischer und semiautomatischer Methoden in forensischen Stimmenvergleichen*, Polizei & Wissenschaft vol. 4, pp. 2-19 (2013).

[4] M. Jessen, A. Alexander and O. Forth, *Forensic voice comparisons in German with phonetic and automatic features using VOCALISE software*, in proceedings of the Audio Engineering Society 54th International Conference 2014, pp. 28–35 (2014)

[5] T. Kinnunen and H. Li, An overview of *Text-Independent Speaker Recognition: From Features to Supervectors*, Speech Communication, vol. 52, no. 1, pp. 12-40 (2010).

[6] J. H. L. Hansen and T. Hasan, *Speaker Recognition by Machines and Humans: A tutorial review,* IEEE Signal Processing Magazine, vol. 32, no. 6, pp. 136–145 (2015).

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey and S. Khudanpur, *X-Vectors: Robust DNN Embeddings for Speaker Recognition*, In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018: 5329-5333 (2018).

[8] S. Davis and P. Mermelstein*, Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences*. IEEE Trans. Acoustics, Speech, and Sig. Proc., vol. 28, no. 4, pp. 357–66 (1980)

[9] S. Furui, *Speaker-independent isolated word recognition using dynamic features of speech spectrum*. IEEE Trans. Acoustics, Speech, and Sig. Proc., vol. 34, pp 52–59, (1986).

[10] S. Furui, *Cepstral analysis technique for automatic speaker verification*. IEEE Trans. Acoustics, Speech, and Sig. Proc., vol. 29, pp. 254–272 (1981).

[11] P. Boersma and D. Weenink, *Praat: doing phonetics by computer [Computer program], Version 5.3.42*, retrieved 2 June 2013 from http://www.praat.org/

[12] D. A. Reynolds, T. F. Quatieri and R. B. Dunn, *Speaker Verification using Adapted Gaussian Mixture Models*. Digital Signal Processing. 10 (1–3): 19–41 (2000).

[13] N. Dehak, P. J. Kenny, E. Dehak, P. Dumouchel and P. Ouellet, *Front-end factor analysis for speaker verification*. IEEE Trans. Acoustics, Speech, and Language Processing. 19(14): 788–798 (2011)

[14] Y. LeCun, Y. Bengio and G. Hinton, D*eep Learning*, Nature 521, pp 436—444 (2015).

[15] G, J. McLachlan, *Discriminant analysis and statistical pattern recognition.* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics (1992).

[16] S. Prince and J. Elder, *Probabilistic Linear Discriminant Analysis for Inferences about identity*. In IEEE 11th International Conference on Computer Vision (ICCV), 2007, pp 1–8 (2007)

[17] D. Garcia-Romero and C.Y. Espy-Wilson, *Analysis of i-vector Length Normalization in Speaker Recognition Systems.* In Interspeech 2011, pp 249—252 (2011).

[18] D. Snyder, G. Chen, and D. Povey, *MUSAN: A Music, Speech, and Noise Corpus*, arXiv: 1510.08484v1 (2015)

[19] M. McLaren, D. Castán, M. K. Nandwana, L. Ferrer, E. Yılmaz, *How to Train Your Speaker Embeddings Extractor*, In Odyssey 2018, pp 327—334 (2018)