Robust Speaker Identification by Fusing Classification Scores with a Neural Network

Kevin Wilkinghoff, Paul M. Baggenstoss, Alessia Cornaggia-Urrigshardt, Frank Kurth

Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE
Fraunhoferstraße 20, 53343 Wachtberg, Germany
Email: {kevin.wilkinghoff, paul.baggenstoss, alessia.cornaggia-urrigshardt, frank.kurth}@fkie.fraunhofer.de

Abstract

Score-based fusion of multiple independent models for the purpose of identifying speakers is widely used as it reduces the identification error rate significantly. In this work, a speaker identification system for low-quality speech which has been propagated through telephone and communication channels is proposed. The system consists of 15 models based on 5 features as well as a Neural Network structure for the task of fusing the classification scores resulting from the individual models. Its performance is evaluated in closed-set speaker identification experiments conducted on the Switchboard corpus. Furthermore, the proposed Neural Network architecture is compared to other fusion techniques such as taking the mean, a Majority Voting, an Evolutionary Algorithm and Logistic Regression.

1 Introduction

Classically, speaker recognition systems are based on a single feature extracted from given speech data, mostly Mel-Frequency Cepstral Coefficients (MFCCs) [1]. These feature vectors are used to train a so-called Universal Background Model (UBM) [2] which is a Gaussian Mixture Model trained on data of various speakers. The UBM is adapted towards single speakers in a step called "enrollment" to attain speaker-specific models. By concatenating the means of the adapted Gaussian components, one obtains high-dimensional supervectors which serve as a fixed-size representation of the speech data. To reduce the dimension of these supervectors, first an i-vector model [3], which is a simplification of Joint Factor Analysis (JFA) [4], and second Probabilistic Linear Discriminant Analysis (PLDA) [5–7], which is essentially JFA applied to i-vectors, is used. For a more detailed review of the named methods, the reader is referred to [8].

It is well known that the performance of a fused classification model is always better than the best individual model if the models are making independent errors (see e.g. [9]). Therefore, many successful attempts have been made to fuse multiple models based on different features or use additional discriminative classifiers as for example Support Vector Machines [10–13].

Scores of multiple classifiers are most commonly fused with a weighted sum, although several techniques to combine the scores exist (see e.g. [14, 15]). But when the number of statistical models involved increases, the weight space grows exponentially. Hence, naively searching in the weight space is computationally infeasible. To address this issue, different techniques such as Logistic Regression (LR) or shallow Neural Networks with no or only one single hidden layer [16, 17] have been used to find the right weights. Usually, deeper Neural Networks are not used for that purpose although they have been successfully applied in the context of speaker recognition by evaluating the posteriors of the UBM, extracting features or speaker embeddings and even doing end-to-end Speaker Recognition (see [16, 18–21]).

The goal of this work is to have a robust speaker identification system for telephone and communication channels especially focusing on a small number of speakers (e.g. 10). For this purpose, a speaker identification system using 5 different features and for each of them 3 statistical models will be presented. The three models are (i) the standard i-vector/PLDA model with (ii) the underlying UBM and (iii) dimensionally reduced versions of the Supervectors via Principal Component Analysis (PCA) and PLDA. As another main contribution, we propose to use a Feed Forward Neural Network with a speaker-independent and a speaker-dependent layer for fusing the obtained classification scores of the models. In speaker identification experiments conducted on the Switchboard corpus, we will evaluate the speaker identification system and compare the Neural Network to other score-based fusion techniques.

2 The Speaker Identification System

The speaker identification system we propose is based on 5 features. In addition to the widely used MFCCs, complementary features based on Pitch, Glottal Mixture Models (GLOMMs) [22, 23], Perceptual Linear Prediction (PLP) [24] and Spectral Subband Centroids (SSCs) [25] are all included into the system because a single feature cannot capture all speaker relevant information. Though it has been shown that fusing some of these features is highly beneficial (see [23, 26, 27]), this is the first speaker identification system utilizing all features together.

The distribution of each feature is captured by another UBM resulting in five high-dimensional supervectors for every utterance after enrollment. Therefore, applying dimension reduction techniques to the supervectors is even more important when using multiple features. However, none of them perfectly keeps all relevant information and dismisses the rest. Hence, combining multiple dimension reduction techniques with the information they are operating on, i.e., the UBM itself, intuitively seems to be beneficial. More concretely, for each of the 5 features, the standard i-vector/PLDA model is fused with the underlying UBM as well as dimensionally reduced versions of the supervectors to compensate for the loss of information when applying dimension reduction techniques. Fig. 1 shows our proposed score-extraction scheme.

3 Score-Based Fusion Techniques

A frequently used method to perform score-based fusion is to take a weighted linear sum of all $M \in \mathbb{N}$ models' scores with one scalar weight $w_m, m = 1, ..., M$ for each differ-

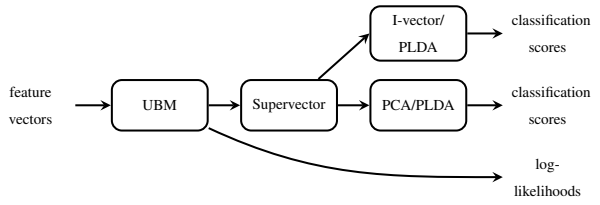

Figure 1: Extraction of the scores for a single feature

ent model $\theta_m, m = 1, ..., M$ (in our case $M = 15$). This corresponds to computing

$$O(s|x) := \sum_{m=1}^{M} w_m O(s|\theta_m, x)$$

where $O(s|\theta_m, x)$ denotes the classification score for target speaker $s \in \mathscr{S}$ given fixed feature vectors x from some feature space and a model θ_m . The actual classification is done by returning the argmax of the fused classification scores as the classification result.

Finding the weights that yield the best performance needs to be done experimentally. In practice, naively searching the resulting M-dimensional weight space is computationally infeasible, due to exponential growth, which is the reason why other techniques need to be applied.

Simple techniques that require no training do exist. One of them is a Majority Voting (MV) which means to classify with each model individually and use the class most models agree on. Another one is to simply take the mean which is equivalent to setting all fusion weights equal to 1. However, their general drawback is that they treat every model equally regardless of performance. In case that some models perform significantly worse than others, this leads to highly suboptimal results.

3.1 Evolutionary Algorithm

One possible solution to search the full weight space is to apply an Evolutionary Algorithm (EA) [28]. Its basic idea is to initialize a random population (of weight vectors) and optimize it by simulating evolution. We used a population of 2000 individuals uniformly distributed on the multidimensional open interval $(0, 1)^M \subset \mathbb{R}^M$. Then, we repeated the following steps for 100 iterations:

- Fitness evaluation: compute classification accuracy by fusing scores weighted with an individual
- External selection: keep best 10% of the entire population
- Inheritance: with a chance of 95% take the mean of two random parents, otherwise initialize the child uniformly distributed on $(0, 1)^M$
- Mutation (only applied to children): for each dimension independently, multiply in 40% of the cases with a uniformly distributed number in $(0, 2)$

At the end, one individual yielding the highest classification accuracy is kept as the result.

3.2 Logistic Regression

The basic idea of Logistic Regression (LR) [29] is to find the fusion weights by training a discriminative model for classification. In order to map the scores to class probabilities, the softmax function, which is a multi-dimensional generalization of the logistic function, is applied. It is

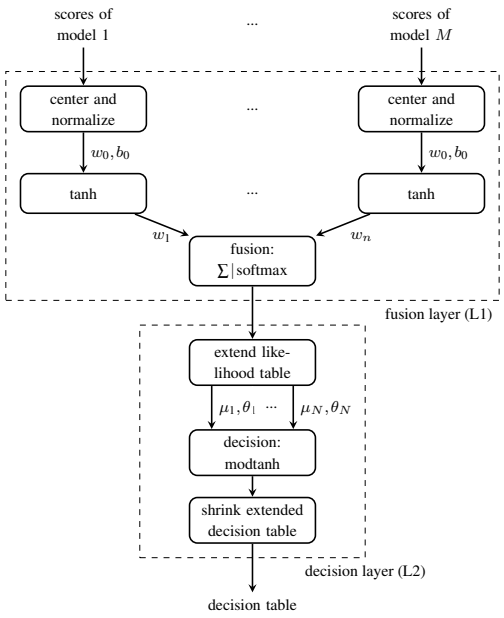

Figure 2: Structure of the Neural Network (NN)

given by

$$P(s|x) := \frac{\exp\left(\sum_{m=1}^{M} w_m O(s|\theta_m, x)\right)}{\sum_{t \in \mathscr{S}} \exp\left(\sum_{m=1}^{M} w_m O(t|\theta_m, x)\right)}$$

for speaker $s \in \mathscr{S}$. One can train this model to match the target distribution given by the categorical labels e.g. by minimizing the categorical crossentropy via gradient descent. In other words, the goal of training is to have $P(s|x) = 1$ if x belongs to speaker s and $P(s|x) = 0$ else.

3.3 Neural Network

The Neural Network (NN) used for fusing all classification scores consists of 2 layers (see Fig. 2), a model specific fusion layer (L1) and a speaker specific decision layer (L2). Both layers are trained individually, one after another, to reduce the effect of overfitting to the validation data which easily happens when directly operating on the scores. Additionally, we applied early stopping by monitoring the training loss.

The first layer consists of two preprocessing steps: First the classification scores are centered and normalized to have a standard deviation of 1. Secondly, a hyperbolic tangent with a single global weight $w_0 \in \mathbb{R}$ and bias $b_0 \in \mathbb{R}$ is applied. This layer serves as a global boundary distinguishing between scores belonging to positive and negative examples. After that, the scores are fused with a weighted sum and the weights are found via LR. This is achieved by applying the softmax function and minimizing the categorical crossentropy via Backpropagation of Error. Note, that the standard notation of the term *layer* is abused because, strictly speaking, the fusion layer itself consists of two layers of a Neural Network.

To improve robustness, the idea of the second layer (the decision layer) is to differentiate between positive and negative examples by using another, individually trained, decision function for each speaker. Mathematically, the goal is that likelihoods of positive examples are mapped to a value of 1 and likelihoods of negative examples are mapped to -1. Thus, the speaker dependent transfer function decides whether the underlying speech data of a likelihood value

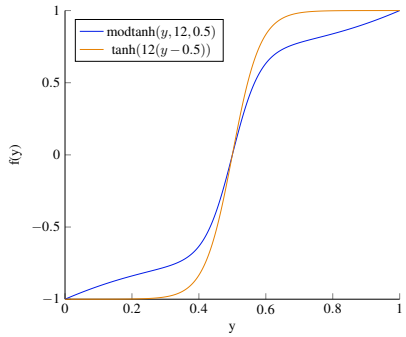

Figure 3: Modified hyperbolic tangent

belongs to the corresponding speaker or not. For this purpose, a modified hyperbolic tangent, which is given by

$$\begin{aligned} \text{modtanh}: \quad & [0,1] \to [-1,1] \\ & y \mapsto \tanh(\lambda(y-\mu))(y(y-1)+1) \end{aligned}$$

with decision boundary $\mu \in [0,1]$ and sharpness $\lambda \in \mathbb{R}_+$, is used as a decision function. Throughout the paper we will use $\mu = 0.5$ and $\theta = 12$ as initial values for the parameters. Note, that multiplying with the term $(y(y-1)+1)$ prevents that the gradient vanishes at the boundary of the interval $[-1,1]$. This is done because values are very likely to be chosen as the classification result when being large and very unlikely to be chosen when being small and thus are of particular interest. See Fig. 3, for a visualization of the decision function. All parameters $\mu_1, ..., \mu_N, \lambda_1, ..., \lambda_N$, $N \in \mathbb{N}$ of this layer are trained by minimizing the squared error via Backpropagation of Error.

As said before, we consider closed-set speaker identification tasks where $K \in \mathbb{N}$ speakers are selected out of a database consisting of $N >> K$ speakers. In order to have a single Neural Network which can be used regardless of the actual choice of these speakers, individual decision functions for each of the N speakers need to be trained. Thus, the likelihood table must be extended first by entering likelihoods with a value of 0 for all speakers who have not been considered for identification in a specific test. As an example, consider the case where one wants to classify among $K = 10$ speakers, which is the size of the output of the fusion layer, but has a database of $N = 212$ speakers. Then, a likelihood of 0 is entered for the remaining $N - K = 202$ speakers. This needs to be done, in order to know the speakers to whom the likelihood values correspond to and to be able to train speaker specific decision functions. After applying the decision functions, the likelihood table can be shrunk by removing the dummy zeros, which have been mapped to a value of -1, again.

An example of the outputs of the L1 and L2 layer can be found in Fig. 4. Applying the decision function without training only pushes all values towards 1 and -1 respectively (compare a) and b)). Therefore, the classification results are exactly the same when not training the decision layer because, initially, the decision function is the same for all speakers and strictly monotone increasing. The goal of training is to have fewer values in the upper left (false positives) as well as in the lower right area (false negatives) because those values certainly lead to misclassifications. As one can see when comparing b) and c), the training helps to thin out those areas i.e. reduce the number of false positives and false negatives and therefore lower the identification error rate.

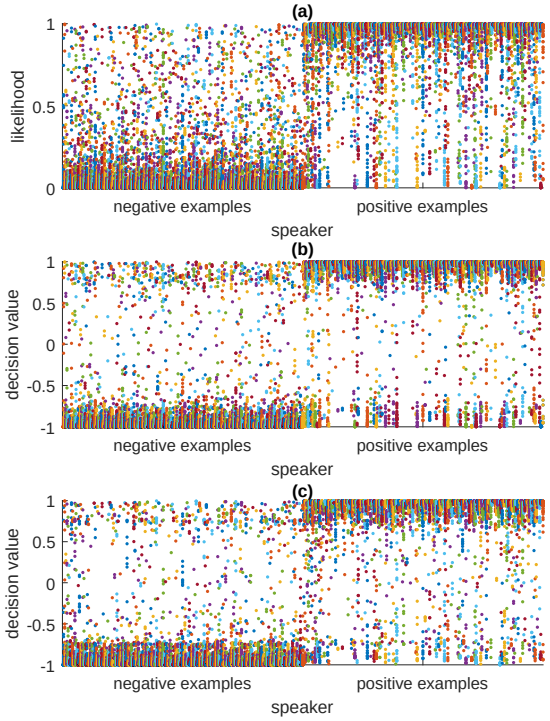

Figure 4: Outputs of the Neural Network at (a) the fusion layer, (b) the decision layer without training and (c) the decision layer after training. In this example, only GLOMM features have been used.

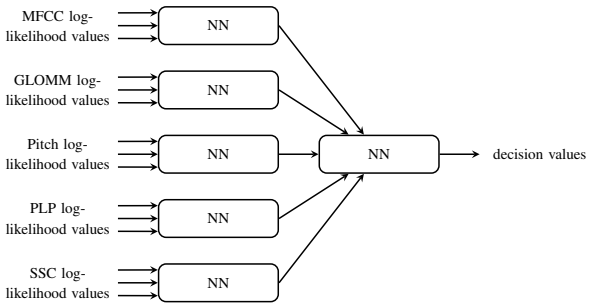

Figure 5: Structure of the Cascaded Neural Network

3.4 Cascaded Version of the Neural Network

A variation of the presented Neural Network (NN) is to first fuse the three models for each feature and fuse the resulting likelihoods afterwards by applying the NN structure in both cases. This variation will be called Cascaded Neural Network in the following and is visualized in Fig. 5. Again, this prevents the Neural Network from overfitting because all layers are trained individually and therefore only parts of the complete information can be accessed in each training phase as some of it is lost or not available yet.

4 Experiments

4.1 Experimental Setup

The performance of the proposed speaker identification system will be evaluated with closed-set speaker identification experiments on a subset of the Switchboard-1 Release 2 corpus [30]. It consists of data sent over telephone and communication channels which is suitable for closed-set

Table 1: Identification error rates obtained without fusion.

feature	UBM	i-vector/PLDA	PCA/PLDA
MFCC	5.56%	3.21%	4.30%
GLOMM	12.52%	10.55%	14.10%
pitch	42.45%	41.74%	40.84%
PLP	5.58%	2.50%	3.92%
SSC	17.84%	11.87%	13.76%

Table 2: Comparison of different fusion methods.

feature	mean	MV	EA	LR	(L1)	NN
MFCC	3.85%	3.82%	3.10%	3.21%	2.86%	**2.68%**
GLOMM	9.68%	10.11%	9.11%	9.11%	9.04%	**8.65%**
pitch	39.09%	40.02%	39.16%	38.63%	38.55%	**38.45%**
PLP	3.37%	3.32%	2.30%	2.52%	2.34%	**2.27%**
SSC	11.15%	11.53%	10.49%	10.40%	**10.24%**	10.44%
all	2.65%	2.63%	1.91%	1.94%	1.90%	**1.83%**
all (cascaded)	-	-	2.07%	2.08	-	**1.76%**

speaker identification experiments because there is a sufficient number of files per speaker. As a first preprocessing step, a Voice-Activity-Detector (VAD) which combines the speech detection algorithm given in [31] with the audio features of kurtosis and spectral entropy has been applied to the data. Furthermore, only 212 speakers out of all available speakers, each with 10 audio files, have been used. To compensate for crosstalk, each file has been reduced to a length of 5 seconds composed of short segments where no energy has been detected in the other speaker's channel. As a result, it is ensured that each audio file captures only a single speaker.

For each speaker, 6 of the 10 files i.e. 30 seconds, have been used for training the UBM, i-vector and PLDA models which have been trained with the fastPLDA toolkit [32]. 2 of the 4 remaining files have been labeled as test data and the other 2 were used as validation data to tune the numerous hyperparameters of the models. Furthermore, the scores extracted from the validation data were used to train all fusion models and be able to check their generalization capablities with the test data.

We used the HTK toolkit [33] to extract MFCCs and PLP features. Both are 19 dimensional and were computed on 25ms long frames with 10ms overlap. The GLOMM features have been computed with the algorithm for telephone channels described in [23]. For extracting the pitch features, the signals were divided into overlapping Hanning-weighted frames and for each of them the autocorrelation function (ACF) was computed. Next, the highest ACF value in the range of human pitch was detected. Its position is the pitch period and its value normalized with the value at zero is the pitch amplitude. As humans perceive pitches logarithmically, we applied the logarithm to the pitch period. Then, each pitch feature is defined as a two-dimensional vector consisting of those two values. To extract the SSC features, the procedure described in [27] has been used.

To compute the classification scores, we sampled 500 sets consisting of 10 randomly chosen speakers and used the same fixed sets to evaluate the models with both validation and both test files. In conclusion, we conducted $500 \cdot 10 \cdot 2 = 10000$ independent speaker classification trials for each test. The error rates of all 15 models can be found in Table 1. Using the centered and normalized resulting scores extracted from the validation data, we then applied the different fusion techniques presented in section 3. We used the BOSARIS toolkit [34] (with the default objective function and a prior of 0.1) to apply LR. The results were then used to compute the corresponding identification error rates on previously unseen test data.

4.2 Experimental Results

The results obtained in the score-based fusion experiments can be found in Table 2. First, it is visible that fusing all three models significantly improved the identification error rates regardless of the actual features being used (compare Table 1 and 2). In addition to that, the EA, LR and the Neural Network performed much better than simply taking the mean. Moreover, the performance may actually get worse when fusing the scores by taking the mean (e.g. PLP: 2.50% $\rightarrow$ 3.37%). The same is true for the MV, as the performance was mostly even worse than when taking the mean. If all 15 models of all features were used, the error rates of the mean and MV were roughly the same and better than any individual model but still much higher than the other 3 fusion techniques. Comparing those 3 techniques, one sees that the Neural Network delivered the lowest error rates except for SSC where it was slightly worse than LR. To see the effect of the L2 layer on the error rate, we also state the error rates obtained with L1 only. In genereral, the full NN performed better than L1 alone with SSC being the only exception again. We also experimented with using additional layers but this did not improve the results. As expected, the cascaded fusion led to worse performance when using the EA or LR because only a subspace of the total weight space is covered. However, the performance of the cascaded NN was better than the original NN.

Since the pitch features alone performed much worse than any of the other features, we also evaluated the cascaded NN without pitch which led to an identification error rate of 1.94%. In conclusion, the pitch features are containing additional information and thus are an important component of the system.

5 Conclusions and Future Work

In this work, a Neural Network architecture used to fuse scores of a speaker identification system for applications with telephone quality has been presented. The system consists of three speaker identification models (UBM, I-vector/PLDA and Supervector/PCA/PLDA) based on five complementary features (MFCC, GLOMM, Pitch, PLP and SSC). As shown in closed-set experiments on Switchboard, fusing the three models always reduces the error regardless of the features being used. Additionally, the proposed Neural Network consisting of a model-specific fusion and a speaker-specific decision layer led to slightly better results than all other approaches when encountering previously unseen test data.

For the future, we plan to complement the presented system with other deep Neural Networks which extract features or do end-to-end speaker recognition. By jointly training our proposed Neural Network used for fusing everything with the other ones which serve as an input, an additional improvement may be achieved. Furthermore, evaluating the system with a recent NIST Speaker Recognition Evaluation task will be helpful for comparing the performance of the speaker identification system to performances obtained with other systems.

References

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digit. Signal Process.*, vol. 10, pp. 19–41, Jan. 2000.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[5] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[6] P. Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Odyssey 2010 - The Speaker and Language Recognition Workshop*, p. 14, 2010.

[7] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Interspeech*, vol. 2011, pp. 249–252, 2011.

[8] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: a tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[9] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[10] D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, *et al.*, "The supersid project: Exploiting high-level information for high accuracy speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 784–787, IEEE, 2003.

[11] W. M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 161–164, IEEE, 2002.

[12] W. M. Campbell, D. A. Reynolds, and J. P. Campbell, "Fusing discriminative and generative methods for speaker recognition: experiments on switchboard and nfi/tno field data," in *ODYSSEY04-The Speaker and Language Recognition Workshop*, 2004.

[13] N. Brümmer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiát, D. A. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the stbu submission for the nist speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[14] S. Tulyakov, S. Jaeger, V. Govindaraju, and D. Doermann, "Review of classifier combination methods," *Machine Learning in Document Analysis and Recognition*, pp. 361–386, 2008.

[15] M. P. Ponti Jr., "Combining classifiers : from the creation of ensembles to the decision function," in *Graphics, Patterns and Image Tutorials (SIBGRAPI-T), 2011 24th SIBGRAPI Conference on*, pp. 1–10, IEEE, 2011.

[16] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[17] B. Xiang and T. Berger, "Efficient text-independent speaker verification with structural gaussian mixture models and neural network," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 447–456, 2003.

[18] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pp. 165–170, IEEE, 2016.

[19] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech 2017*, pp. 999–1003, IEEE, 2017.

[20] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 4930–4934, IEEE, 2017.

[21] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet based loss on short utterances," in *Proc. Interspeech 2017*, pp. 1487–1491, 2017.

[22] P. M. Baggenstoss, "Combining the glottal mixture model (glomm) with ubm for speaker recognition," in *Proceedings of the 24th European Signal Processing Conference (EUSIPCO)*, pp. 2156–2160, IEEE, 2016.

[23] P. M. Baggenstoss, K. Wilkinghoff, and F. Kurth, "Glottal mixture model (glomm) for speaker identification on telephone channels," in *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, pp. 2803–2807, IEEE, 2017.

[24] H. Hermansky, "Perceptual linear prediction (plp) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[25] K. K. Paliwal, "Spectral subband centroid features for speech recognition," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 617–620, IEEE, 1998.

[26] M. J. Alam, T. Kinnunen, P. Kenny, P. Ouellet, and D. O'Shaughnessy, "Multitaper mfcc and plp features for speaker verification using i-vectors," *Speech communication*, vol. 55, no. 2, pp. 237–251, 2013.

[27] N. P. H. Thian, C. Sanderson, and S. Bengio, "Spectral subband centroids as complementary features for speaker authentification," in *Biometric Authentification*, pp. 631–639, Springer, 2004.

[28] T. Bäck, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[29] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 1 ed., 2006.

[30] J. J. Godfrey and E. Holliman, "Switchboard-1 release 2 LDC97S62." Web Download, 1993. Philadelphia: Linguistic Data Consortium.

[31] F. Kurth and A. Cornaggia-Urrigshardt, "Evaluation of enhanced f0-trajectories for speech detection and classification in acoustic monitoring," in *Speech Communication; 12. ITG. Symposium; Proceedings of*, pp. 1–4, VDE, 2016.

[32] A. Sizov, K.-A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *Proc. S+SSPR*, 2014. Software available at https://sites.google.com/site/fastplda/.

[33] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, *et al.*, *The HTK book, Version 3.4*. Cambridge University Engeneering Department, 2006.

[34] N. Brümmer and E. de Villiers, "The bosaris toolkit user guide: Theory, algorithms and code for binary classifier score processing," *Documentation of BOSARIS toolkit*, 2011.