Local binary patterns as features for speaker recognition

Waad Ben Kheder, Driss Matrouf, Moez Ajili and Jean-François Bonastre

LIA laboratory
University of Avignon, France
{waad.ben-kheder,driss-matrouf}@univ-avignon.fr
{moez.ajili,jean-francois.bonastre}@univ-avignon.fr

Abstract

The i-vector framework witnessed great success in the past years in speaker recognition (SR). The feature extraction process is central in SR systems and many features have been developed over the years to improve the recognition performance. In this paper, we present a new feature representation which borrows a concept initially developed in computer vision to characterize textures called Local Binary Patterns (LBP). We explore the use of LBP as features for speaker recognition and show that using them as descriptors for cepstral coefficients dynamics (replacing Δ and $\Delta\Delta$ in the regular MFCC representation) results in more efficient features and yield up to 15% of relative improvement compared to the baseline system performance in both clean and noisy conditions.

keywords: local binary patterns, feature extraction, i-vector.

1. Introduction

Front-end feature extraction is one of the three building blocks of speaker recognition systems (along with modeling and back-end scoring). Over the past years, a lot of research effort focused on speaker modeling and scoring resulting in highly efficient frameworks such as UBM/i-vectors [1, 2, 3] (for speaker modeling) and PLDA [4] (for scoring). Despite the effort dedicated to the improvement of the feature extraction process, the Mel-Frequency Cepstral Coefficients (MFCC) [5, 6] are still, to this day, the leading approach in speaker recognition applications (along with perceptual linear prediction coefficients (PLP) [7, 8]). In this context, more robust and discriminative features for speaker recognition are yet to be developed. Usually, the dynamic features are computed over MFCC coefficients (Δ and $\Delta\Delta$ [9, 10]) and used to improve the recognition performance. Formally, they correspond to the first and second derivatives of the Cepstral coefficients with respect to time (speed and acceleration respectively).

Dynamic features contain information that complements the one provided by static features and allows them to become less sensitive to channel and environment distortion. They also play an important role in identifying the speaking styles and pauses of a particular speaker. Indeed, using MFCC+Δ+$\Delta\Delta$ offers an additional 20% of relative improvement in recognition performance when compared to a system using only static MFCC features. Due to the importance of such parameters, many researchers focused on improving them and trying to capture dynamic information more efficiently. One example is the use of regression features [11, 12] instead of first and second order derivatives. Another recently proposed technique that uses DCT-based contextualization [13] proposes to replace MFCC features and their derivatives by a 2D-DCT transform applied on the Mel filter bank outputs. This technique achieves a 25% improvement in recognition performance.

Also, a recent paper [14] proposed a technique that combines static and dynamic features. In this system, MFCC coefficients are computed using mel spaced Gaussian filter banks then combined with their delta derivatives (Δ and $\Delta\Delta$) and energy using principal component analysis (PCA). Then, a probabilistic neural network is used for speaker modeling. This technique improves the system's accuracy by up to 14%. A different approach which makes use of information that is not contained in cepstral features or their derivative, namely the phase spectra and instantaneous frequency in feature extraction has lately been introduced in [15].

Finally, with the rise of deep learning in the last years, a different kind of approaches is being investigated for features extraction making use of deep neural networks. Unlike classical approaches, these techniques rely on the capability of a deep neural network (DNN) to learn underlying structures from acoustic data (either in the spectral, cepstral or temporal domain) in a non-supervised fashion [16]. After training, the DNN is used to extract features which can be used as input to a regular i-vector-based SR system.

In this paper, we introduce a new set of features based on MFCCs and borrowing a concept used in facial recognition called Local Binary Patterns (LBP). Usually, LBP features are used to describe textures for facial recognition and have been proven to be efficient and highly discriminative [17]. Lately, LBP descriptors were adapted to speech applications and were successfully used in the cepstral domain to build an anti-spoofing system (for speaker recognition) [18].

Since working with LBP requires a 2D arrangement of features (like pixels in a 2D image), [18] used an horizontal stacking of cepstral features as input to the LBP algorithm. The success of this representation in an anti-spoofing context motivated us to utilize them as input features to a regular i-vector-based SR system and test their discriminative power. Moreover, while the dynamic delta features describe solely the temporal variation of successive speech frames, LBP features describe both temporal and cepstral variation since they are computed over two axis using a circular neighborhood. In consequence, this new feature can be richer in speaker-specific information and might improve the performance of a speaker recognition system.

We will first investigate the use of LBPs as features (computed over cepstral coefficients) compared to the standard MFCC-based systems (MFCC + Δ + $\Delta\Delta$). Then, we show that a combination of the two representations (MFCC + LBP descriptors as an alternative to the classical dynamic descriptors Δ and $\Delta\Delta$) yields up to 15% of relative improvement compared to the baseline system performance. Finally, it would be interesting to test the response of the new proposed features in presence of additive noise and compared it to the classical MFCC+Δ+$\Delta\Delta$ configuration.

This paper is structured as follows: Section 2 summarizes the use of i-vector framework in SR systems. Section 3 introduces Local Binary Patterns and their use in computer vision then in speaker recognition context. Finally, Section 4 specifies the experimental protocol and Section 5 presents the experiments conducted using LBP and the related analysis.

2. The i-vector paradigm

The i-vector paradigm has become in the past years a standard in speaker recognition applications [1, 2, 3]. It was motivated by the existing super-vector-based joint factor analysis (JFA) approach [19]. While the JFA approach models the speaker and channel variability space separately, i-vectors are formed by modeling a single low-dimensional total-variability space that covers both the speaker and channel variability [3]. An i-vector extractor converts a sequence of acoustic vectors into a single low-dimensional vector representing the whole speech utterance. The speaker- and session-dependent super-vector s of concatenated Gaussian Mixture Model (GMM) means is assumed to obey a linear model of the form:

$$s = m + Tw \tag{1}$$

where:

- m is the mean super-vector of the Universal Background Model (UBM)
- T is the low-rank variability matrix obtained from a large dataset by MAP estimation [19]. It represents the total variability subspace.
- w is a normally distributed latent variable called "i-vector".

3. Local Binary patterns for speaker recognition

The LBP operator is a non-parametric 3x3 kernel which summarizes the local spacial structure of an image. It was first introduced by Ojala et al. [20] who showed the high discriminative power of this operator for texture classification. At a given pixel position (x_c, y_c), LBP is defined as an ordered set of binary comparisons of pixel intensities between the center pixel and its eight surrounding pixels (Figure 1). The decimal form of the resulting 8-bit word (LBP code) can be expressed as follows:

$$LBP(x_c, y_c) = \sum_{n=1}^{8} s(i_n - i_c)2^n \tag{2}$$

where i_c corresponds to the grey value of the center pixel (x_c, y_c), i_n to the grey values of the 8 surrounding pixels, and function s is defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{3}$$

Later, Ojala et al. [17] extended their original LBP operator to a circular neighborhood of different radius size as shown in Figure 2. Their $LBP_{P,R}$ notation refers to P equally spaced pixels on a circle of radius R. In [17], they also noticed that most of the texture information was contained in a small subset of LBP patterns. These patterns, called uniform patterns, contain at most two bitwise 0 to 1 or 1 to 0 transitions (circular binary code). 11111111, 00000111 or 10000111 are examples of uniform patterns (they contain respectively 0, 1 and 2 transitions). Using uniform LBP in practice implies that all non-uniform patterns are discarded.

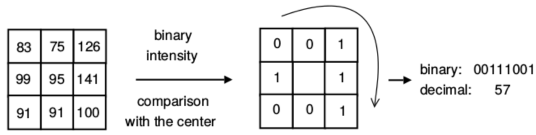

Figure 1: The LBP operator.

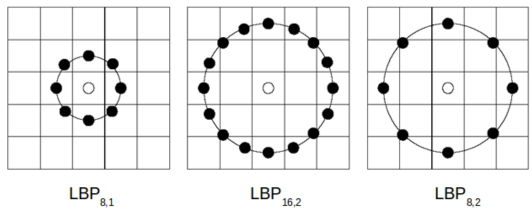

Figure 2: LBP neighborhood for different (P, R) values.

3.1. LBP for speech:

LBP features can be applied to speech as in [18] where, instead of using an image, a horizontal stacking of cepstral features is given to LBP as input. The resulting features matrix is called "textrogram". In [18], LBP was applied to an horizontal stacking of MFCC features and an anti-spoofing system (for speaker recognition) was successfully built using the resulting features. This application motivated us to evaluate the discriminative power of such descriptors and explore the possibility to build better features using LBP in the cepstral domain. Compared to a system using MFCC+Δ+$\Delta\Delta$, LBP features describe the variation of cepstral information on two different axes by describing an entire neighborhood circle (as shown in Figure 3), whereas Δ and $\Delta\Delta$ focus solely (by definition) on the temporal variation.

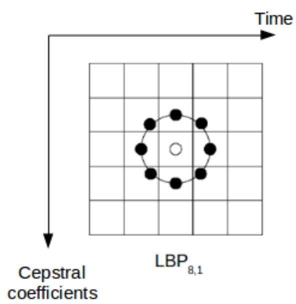

Figure 3: LBP neighborhood for $LBP_{8,1}$ when used on cepstral features. The first axis represents time (consecutive frames) and the second axis corresponds to the cepstral coefficients $(C_0, C_1, ..)$.

The study conducted in [18] found that uniform LBP descriptors are the most frequent in a cepstral features context.

For this reason, uniform LBP will be used throughout this paper and will simply be referred to as $LBP_{P,R}$ (where P indicates the number of points and R specifies the radius of the neighborhood circle). Four different textrograms are used in this work for each utterance based on the corresponding MFCC features: $LBP_{8,1}$, $LBP_{8,2}$, $LBP_{16,2}$ and $LBP_{16,4}$.

4. Experimental protocol

Our experiments operate on 19 Mel-Frequency Cepstral Coefficients (plus energy) augmented with 19 first (Δ) and 11 second ($\Delta\Delta$) derivatives. A mean and variance normalization (MVN) technique is applied on the MFCC features estimated using the speech portion of the audio file. The low-energy frames (corresponding mainly to silence) are removed.

Two gender-dependent GMM-UBMs (male model) having respectively 512 and 1024 diagonal components and a total variability matrix of low rank 400 are estimated (one T matrix for each GMM-UBM) using 15660 utterances corresponding to 1147 speakers (using NIST SRE 2004, 2005, 2006 and Switchboard data). The LIA_SpkDet package of the LIA_RAL/ALIZE toolkit is used for the estimation of the total variability matrix and the i-vector extraction. The algorithms used are described in [21]. Finally a two-covariance-based scoring [22] is applied. The equal-error rate (EER) over the NIST SRE 2008 male test data on the "short2/short3" task [23] is computed. The python library "scikit-image" [24] is used in our experiments to compute LBP descriptors from cepstral coefficients.

In order to test the recognition performance of the new proposed systems in adverse conditions, we use 4 noise samples from the free sound repository FreeSound.org [25] as background noises (air-cooling noise, crowd noise, wind noise and car-driving noise). The open-source toolkit FaNT [26] was used to add these noises to the full waveforms generating new noisy audio files for each noise / SNR level.

5. Experiments and results

5.1. Description of the developed systems

In order to investigate the discriminative power of LBP features and their robustness, we conduct four sets of experiments in clean and noisy conditions:

System 1: Using MFCC features + Δ + $\Delta\Delta$: This configuration represents the baseline system. 19 cepstral coefficients + energy + Δ + $\Delta\Delta$ are used to train the GMM-UBM and the i-vector extractor.

System 2: Using LBPs as features: This configuration uses LBP descriptors (computed over MFCC) as input features for an i-vector-based SR system. As shown in Figure 4, the construction of LBP features for a single utterance is done as follows:

1. The static MFCC features are extracted (19 cepstral features + energy).
2. $LBP_{8,1}$, $LBP_{8,2}$, $LBP_{16,2}$ and $LBP_{16,4}$ textrograms are computed over the resulting MFCCs. As a result, each one of the 4 textrograms has a size of: $20 \times$ *number of frames*.
3. The 4 textrograms are combined with a simple concatenation giving a set of feature LBP^{80} having the size: $80 \times$ *number of frames*.
4. The dimension of LBP^{80} features is further reduced to 40 using PCA. The resultant features LBP^{40} have the size: $40 \times$ *number of frames* (the reduction dimension used in the PCA was chosen in a way that preserves 99% of the data variance).

The generated features (LBP^{40}) corresponding to train data are used to train a GMM-UBM (512 components) and an i-vector extractor (i-vector dimension = 400). In the test phase, the LBP^{40} features computed on enrollment and test data are used to generate the corresponding i-vectors.

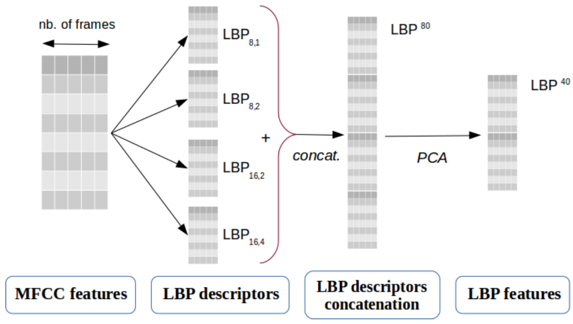

Figure 4: The extraction process of LBP features from MFCC.

System 3: Using MFCC + LBP features: In this configuration, we investigate the use of LBP to describe the variation of cepstral information (as an alternative for Δ and $\Delta\Delta$). To do so, MFCC and LBP^{40} features are computed then concatenated and used as features to train the GMM-UBM and the i-vectors extractor.

System 4: Using MFCC + Δ + $\Delta\Delta$ + LBP features: In this configuration, we investigate the use of both Δ+$\Delta\Delta$ and LBP to describe the variation of cepstral information. To do so, MFCC, Δ + $\Delta\Delta$ and LBP^{40} features are computed then concatenated and used as features to train the GMM-UBM and the i-vectors extractor.

5.2. Results

In this subsection, we present the performance of the four systems in clean and noisy conditions and evaluate the efficiency of LBP descriptors in these two contexts.

5.2.1. *Performance in clean conditions:*

First, we present the results given by the four systems in clean conditions on the eight conditions of the 2008 NIST SRE evaluation [23] noted *det1* to *det8*:

- *det1:* All trials involving only interview speech in training and test.
- *det2:* All trials involving interview speech from the same microphone type in training and test.
- *det3:* All trials involving interview speech from different microphones types in training and test.
- *det4:* All trials involving interview training speech and telephone test speech.
- *det5:* All trials involving telephone training speech and non-interview microphone test speech.

Table 1: Performance of the four systems in clean conditions using two different world models (512 and 1024 components) and an i-vectors size of 400 (a different UBM and i-vector extractor are built for each system depending on the used features).

System	GMM-UBM components	EER (%)							
		det1	*det2*	*det3*	*det4*	*det5*	*det6*	*det7*	*det8*
MFCC + Δ + $\Delta\Delta$	512	7.51	0.40	7.86	4.10	3.58	4.91	1.59	0.91
LBP^{40}		7.53	0.45	8.43	5.30	4.18	5.02	2.31	1.01
MFCC+LBP^{40}		**6.75**	**0.37**	**7.00**	**3.81**	**3.15**	**4.47**	**1.35**	**0.86**
MFCC+Δ + $\Delta\Delta$+LBP^{40}		7.49	0.39	7.84	4.09	3.56	4.89	1.57	0.89
MFCC + Δ + $\Delta\Delta$	1024	7.16	**0.40**	7.50	4.77	3.76	5.03	1.77	0.88
LBP^{40}		7.48	0.46	8.13	5.33	4.22	4.92	2.28	1.01
MFCC+LBP^{40}		**6.45**	0.41	**6.80**	**3.75**	**3.39**	**4.89**	**1.43**	**0.87**
MFCC+Δ + $\Delta\Delta$+LBP^{40}		7.15	0.39	7.48	4.75	3.76	5.02	1.76	0.87

Table 2: Performance of the four systems in mismatched conditions (clean enrollment and noisy test) with a 512 components GMM-UBM and an i-vector size of 400.

		EER (%)			
Test conditions		*MFCC* + Δ+$\Delta\Delta$	LBP^{40}	*MFCC* +LBP^{40}	*MFCC* + Δ+$\Delta\Delta$ + LBP^{40}
Air-cooling noise	*10dB*	3.62	3.59	**3.35**	3.61
	5dB	6.15	6.17	**5.65**	6.14
	0dB	10.67	10.52	**9.58**	10.65
Crowd noise	*10dB*	4.02	4.00	**3.73**	4.01
	5dB	7.65	7.70	**7.03**	7.64
	0dB	11.23	11.21	**10.08**	11.21

- *det6:* All trials involving only telephone speech in training and test.
- *det7:* All trials involving only English language telephone speech in training and test.
- *det8:* All trials involving only English language telephone speech spoken by a native U.S. English speaker in training and test.

Table 1 presents the performance of the four systems using 512 and 1024 components GMM-UBMs.

From Table 1, we see that a system using LBP^{40} as features does not perform better that a standard MFCC system (MFCC+Δ+$\Delta\Delta$). On the other hand, combining MFCC features with LBP^{40} descriptors perform consistently and produces more robust features in all conditions. The global system performance is improved in this context from 6% up to 15% (in terms of relative improvement). Finally, we observe that combining LBP^{40} features and $\Delta + \Delta\Delta$ preserves the original system performance and does not change much the resulting EER. This can be explained by the redundancy of dynamic information between the two dynamic representations (LBP and $\Delta + \Delta\Delta$).

We also observe that using a 512 components GMM-UBM is sufficient for this task and that no significant improvement is observed using a 1024 components world model. For this reason, only 512 component GMM-UBM will be used in the next subsection.

5.2.2. *Performance in noisy conditions:*

Now, we evaluate the 4 systems performance in noisy environments for the *det7* condition (all trials involving only English language telephone speech in training and test). Table 2 presents 4 systems performance in mismatched conditions (clean enrollment and noisy test data) and Table 3 presents the systems performance when both enrollment and test data are noisy (affected by different noises: {wind noise , car-driving noise } for enrollment and {crowd noise, air-cooling noise} for test).

We can see from Tables 2 and 3 that combining MFCC static features with LBP^{40} gives the best recognition performance and improves the baseline system performance by up to 15% of relative improvement. We also observe that combining the two dynamic features (LBP and $\Delta + \Delta\Delta$) does not improve the recognition performance compared to an MFCC+$\Delta + \Delta\Delta$ system. This can also be explained by redundant information

Table 3: Performance of the four systems in noisy conditions ({wind noise , car-driving noise } for enrollment and {crowd noise, air-cooling noise} for test) with a 512 components GMM-UBM and an i-vector size of 400.

		EER (%)			
Test/enrollment conditions		*MFCC* + Δ+$\Delta\Delta$	LBP^{40}	*MFCC* +LBP^{40}	*MFCC* + Δ+$\Delta\Delta$ + LBP^{40}
Wind noise in enrollment Crowd noise in test	*10dB*	10.53	10.64	**9.61**	10.51
	5dB	16.84	17.01	**15.45**	16.82
	0dB	26.96	27.24	**24.74**	26.93
Car-driving noise in enrol. Air-colling noise in test	*10dB*	8.32	8.40	**7.53**	8.31
	5dB	14.77	14.92	**13.38**	14.75
	0dB	25.61	25.87	**23.37**	25.58

in the two representations. It is also worth noting that LBP^{40} features can outperform MFCC+$\Delta + \Delta\Delta$ features in noisy conditions even though the MFCC+$\Delta+\Delta\Delta$ system performs better in clean conditions (see Table 1).

These results along with the clean system performance prove that using LBP^{40} features instead of the regular MFCC dynamic features provides a more robust representation and can be used as an alternative to MFCC+ Δ+$\Delta\Delta$ for speaker recognition.

6. Conclusion

In this paper, we investigated the use of Local Binary Patterns descriptors in a speaker recognition context. Once computed over MFCC features, we used them as features to describe speakers identity then utilized them to describe the variation of cepstral information as an alternative to Δ and $\Delta\Delta$ in a classical MFCC representation. We showed that LBP textrograms capture short-time feature dynamics beyond that in conventional dynamic parametrization and yield up to 15% of relative improvement compared to the baseline system performance when used on both clean and noisy conditions. In a future work, It would be possible to use a DNN-based approach to combine more efficiently these features in order to achieve better recognition performances.

7. References

[1] John HL Hansen and Taufiq Hasan, “Speaker recognition by machines and humans: A tutorial review,” *Signal Processing Magazine, IEEE*, vol. 32, no. 6, pp. 74–99, 2015.

[2] Pulkit Verma and Pradip K Das, “i-vectors in speech processing applications: a survey,” *International Journal of Speech Technology*, vol. 18, no. 4, pp. 529–546, 2015.

[3] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, “Front-end factor analysis for speaker verification,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[4] Patrick Kenny, “Bayesian speaker verification with heavy-tailed priors.,” in *Odyssey*, 2010, p. 14.

[5] JS Bridle and MD Brown, “An experimental automatic word recognition system,” *JSRU Report*, vol. 1003, no. 5, 1974.

[6] Steven B Davis and Paul Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[7] Hynek Hermansky, “Perceptual linear predictive (plp) analysis of speech,” *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[8] Steven B Davis and Paul Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[9] Sadaoki Furui, “Speaker-independent isolated word recognition using dynamic features of speech spectrum,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 1, pp. 52–59, 1986.

[10] Brian Hanson and Ted Applebaum, “Robust speaker-independent word recognition using static, dynamic and acceleration features: Experiments with lombard and noisy speech,” in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 857–860.

[11] Ted H Applebaum and Brian A Hanson, “Regression features for recognition of speech in quiet and in noise,” in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*. IEEE, 1991, pp. 985–988.

[12] JS Mason and X Zhang, “Velocity and acceleration features in speaker recognition,” in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*. IEEE, 1991, pp. 3673–3676.

[13] Moray McLaren, Luciana Ferrer, Yun Lei, et al., “Effective use of dcts for contextualizing features for speaker recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4027–4031.

[14] Khan Suhail Ahmad, Anil S Thosar, Jagannath H Nirmal, and Vinay S Pande, “A unique approach in text independent speaker recognition using mfcc feature sets and probabilistic neural network,” in *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*. IEEE, 2015, pp. 1–6.

[15] Eliathamby Ambikairajah, “Emerging features for speaker recognition,” in *Information, Communications & Signal Processing, 2007 6th International Conference on*. IEEE, 2007, pp. 1–7.

[16] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” in *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[17] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[18] Federico Alegre, Ravichander Vipperla, Asmaa Amehraye, and Nicholas Evans, “A new speaker verification spoofing countermeasure based on local binary patterns,” in *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon: France (2013)*, 2013, p. 5p.

[19] Patrick Kenny, “Joint factor analysis of speaker and session variability: Theory and algorithms,” *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[20] Timo Ojala, Matti Pietikäinen, and David Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[21] Driss Matrouf, Nicolas Scheffer, Benoit GB Fauve, and Jean-François Bonastre, “A straightforward and efficient implementation of the factor analysis model for speaker verification.,” in *INTERSPEECH*, 2007, pp. 1242–1245.

[22] Niko Brümmer and Edward De Villiers, “The speaker partitioning problem.,” in *Odyssey*, 2010, p. 34.

[23] "The NIST year 2008 speaker recognition evaluation plan," http://www.itl.nist.gov/iad/mig//tests/sre/2008/, 2008, [Online; accessed 15-May-2014].

[24] Stefan Van Der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, pp. e453, 2014.

[25] "Freesound.org," http://www.freesound.org.

[26] H. Guenter Hirsch, "FaNT - Filtering and Noise Adding Tool," http://dnt.kr.hsnr.de/download.html, [Online; accessed 15-May-2014].