Improving Voice Activity Detection in Movies

Bernhard Lehner, Gerhard Widmer, Reinhard Sonnleitner

Department of Computational Perception
Johannes Kepler University of Linz, Austria
{bernhard.lehner,gerhard.widmer,reinhard.sonnleitner}@jku.at

Abstract

Voice Activity Detection in movies is a non-trivial and challenging task. The different emotional states of the speakers, as well as the variety of soundscapes and noises contribute to the complexity of the task. In this paper, we propose a set of lightweight features that are specifically designed to perform under such conditions, while at the same time preventing confusions of singing voice with speech. For evaluation, we use four full-length movies, previously unseen to the system and painstakingly annotated. We compare our detector to a state-of-the-art reference system. The new approach performs better, yielding just about half the Equal Error Rate (EER). Furthermore, since the ground truth annotation task is extremely tedious, and to help with advancing in this topic, we release the annotations of all four movies to the research community.

Index Terms: Voice Activity Detection, Speech Detection

1. Introduction

Voice Activity Detection (VAD), sometimes also referred to as Speech Activity Detection (SAD), is an important preprocessing method to increase the performance of many applications, like Automatic Speech Recognition (ASR).

In movies, one of the challenges for VAD is due to the various emotional states of the speakers, ranging from anger to fear to sadness and everything in between. Those emotional states correspond to vocal effort categories like whisper, soft voice, normal voice, loud voice, and shouting [1], and directly impact the performance of VAD and ASR systems. Another factor that contributes to the difficulty, is the extremely diverse soundscape of movies. Music, wind, rain, traffic, battle, and crowds are just a few examples of noises that occur in movies (audio examples are available at `http://www.cp.jku.at/misc/is2015vad/`).

This paper presents a method that is specifically designed to detect the presence of speech under such difficult conditions. This could be useful for tasks like automatic subtitle alignment, and pave the way to an automatic subtitle generator system.

Furthermore, we will show, that promising results can be achieved despite a relatively small training set of just 4 h audio material. This is useful, since publicly available data sets are scarce, and annotating ground truth is a tedious task. By using radio broadcasts (see Sec. 4.1) for the development of our method, we were additionally able to consider the capability to discriminate between speech and singing voice. We consider this ability important, because a speech recogniser fed with singing voice, will most probably behave erroneous [1]. After all, in many movies there is singing voice present, be it as part of the soundtrack, or even sung by the actors themselves, like in movie musicals. Finally, besides music, vocals, and speech, radio broadcasts contain ads, which can be a good resource of speech mixed with highly non-stationary noise.

Another contribution, besides the proposed feature set, is the release of the ground truth annotations of four full-length movies to the research community (see Sec. 4.2).

2. Related Work

Often, VAD algorithms utilise features relating to energy [2], zero crossing rate [3], spectral flatness [4], or periodicity [5]. However, the challenging conditions in movies render such features less useful. Therefore, more sophisticated – or a fusion of several – features could improve VAD results. Recently, the methods from the data-driven category (where a classifier compares acoustic features to a previously trained model) yielded promising results, especially under highly corrupted conditions.

Our starting point and baseline is the method by Eyben et al. [6], where results on four Hollywood movies were published, and compared to three different state-of-the-art VAD algorithms [7, 8, 9], which they clearly outperformed.

Supposedly to avoid the tedious task of annotating more movies to be used as training data, they synthesise audio data by mixing speech data from the Buckeye [10] and the TIMIT [11] corpus with the following types of noise: babble, city, white and pink noise, and music. The training set comprises 34:54 h of audio, where 15:08 h are speech. The validation set comprises 3:00 h, where 1:22 h are speech. The strategies involved to create such a large and diverse amount of data are described in detail in [6], and a discussion of these would go beyond the scope of this work.

As features, they use standard RASTA-PLP coefficients 1-18, along with their first order derivatives (deltas), yielding a 36 dimensional feature vector. The features are extracted with the open-source toolkit openSMILE [12], with a frame size of 25 ms, and a step size of 10ms. As classifier, they use *Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs)* [13], which have access to the complete past, and are capable of modelling long-range temporal context.

They introduce two different network topologies, from which we only reimplement the one giving the better results: the uni-directional RNN with one hidden layer and 50 LSTM units. For increased robustness, they train three networks with different weight initialisations, which are then combined by averaging their outputs. Since we use a relatively small training set, we adapt this strategy to increase robustness even further. In our implementation, we train four networks with different training and validation sets (according to a four-fold cross validation split), and combine them by averaging their outputs. Z-normalisation is applied per fold to avoid too optimistic results.

Thresholds resulting in an Equal Error Rate (EER) are fixed according to the validation set results, and the test set is left

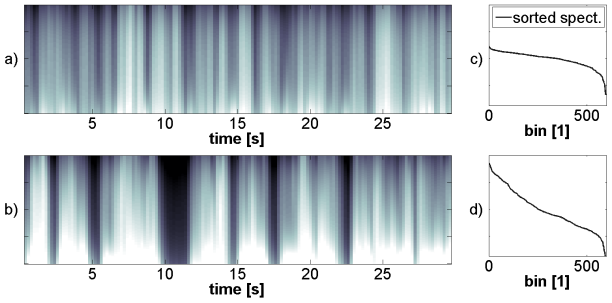

Figure 1: a) Sorted Spectrogram of male speech and b) of male singing. Brighter regions correspond to higher harmonicity; c) sorted spectrum of a single frame of male speech and d) of male singing. PSSC is a 3rd order polynomial fitted on such sorted spectra, and allows to distinguish speech and singing voice.

unseen. Non-speech segments shorter than 5 frames are then smoothed out to be speech segments.

3. Features

In order to detect speech when the additive noise is non-stationary, we propose a set of features, that target two different aspects of speech: First, the temporal characteristic descriptors, that quantify how the signal evolves in time (Fluctogram along with Spectral Flatness and Spectral Contraction). The Fluctogram reveals the variation of pitch, and was originally used for Singing Voice Detection [14]. However, the different intonations in speech and singing voice often enable a classifier to distinguish both, when trained with proper data sets. In order to allow standard classifiers like Random Forest (RF) or Support Vector Machine (SVM) to consider temporal characteristics of the signal, those feature values are summarised over time, by calculating means or variances over several frames.

Second, we propose a modified version of Spectral Contrast (Polynomial Shape Spectral Contrast), that serves as harmonicity descriptor, hence corresponding to timbre. The vowel-to-consonant duration ratio of speech is generally lower than that of singing voice, yielding a lower long-term (we use 800ms frames) harmonicity for speech. In Fig. 1, we compare Spectral Contrast of male speech and male singing, normalised w.r.t. energy. Clearly, the differences in harmonicity (brighter frames represent higher harmonicity) allow for discrimination between speech and singing voice, at least to a certain extent.

These temporal and harmonicity descriptors complement each other, and enable a classifier to deal with some of the challenging situations in movies.

3.1. Fluctogram

The harmonic (voiced) parts of speech are often clearly visible in the spectrogram, characterised by fluctuations of partials in a continuous manner. To detect such fluctuations without the necessity of error-prone pitch estimation, several attempts have already been made, but only in the context of speech processing, e.g. by Laskowsky and Jin [15], and Ma et al. [16]. In order to deal with the presence of musical accompaniment – which is an integral part of movies –, the *Fluctogram* [14] was introduced.It is a modification of the method of Sonnleitner et al. [17], where the *cross correlation* is utilised in an intuitive way to detect the presence of speech: Each spectrum of a time frame X_t is compared to the spectrum of its successor X_{t+1}, and the index of the maximum correlation when X_{t+1} is shifted $\pm n$ bins, is calculated.

We first compute the magnitude spectrum by performing a DFT on audio frames for every 20ms. The actual window over which the DFT is applied is 100 ms long, and always placed symmetrically around the current frame. A zero padding of 2^3 is applied, to assure the proper resolution in the lower frequency region of the spectrum. Afterwards, the spectrum is mapped to a scale that relates to pitch, where the range of one semitone comprises 10 bins. Our pitch scale comprises five octaves from E3 (164 Hz) to E8 (5274 Hz). We then divide the resulting 600-bin spectrum into 13 overlapping bands, each band 120 bins wide, resulting in a bandwidth of one octave. The distance from one band to the next is 40 bins, which equals four semitones. In order to reduce the influence of partials near the boundaries that are potentially leaving a frequency band, each band is then weighted by a triangle window that matches the bandwidth. The harmonic fluctuations within each band are then revealed by identifying the maximum cross correlation at shifts of ± 5 bins.

Finally, each frame of audio is then characterised by computing the variance over a window of 40 successive Fluctogram values, centered on the current frame, separately for each of the 13 frequency bands.

3.2. Spectral Flatness and Spectral Contraction

When we analyse e.g. the signal of a pitch-discrete instrument – like a piano –, the Fluctogram should reveal no fluctuations whatsoever. However, a small amount of fluctuation is still being detected, mostly due to a percussive onset characteristic of the sound source. Therefore, two reliability indicators were introduced to accompany the Fluctogram. First, the Fluctogram is most reliable when the signal is not noise-like, which is characterised for each frequency band by the *Spectral Flatness (SF)* measure [18].Each frame of audio is then represented by the mean over a window of 40 successive values, centered on the current frame, yielding another 13 feature values.

Second, the most appropriate estimations of fluctuations are provided, when the trajectory of the partial that dominates the result of the cross-correlation, resides near the center of the frequency band. To account for that, the *Spectral Contraction (SC)* [14] relates the energy in the center to the total amount of energy in the spectrum. The SC feature is simply the energy-wise ratio of a Chebyshev-windowed spectrum to the spectrum itself, computed separately for each frequency band. Each frame is then quantified by the variance over a window of 40 successive values, centered on the current frame, yielding again 13 feature values.

3.3. Polynomial Shape Spectral Contrast

Spectral Contrast relates the peaks to the valleys of the spectrum in several sub-bands, and could be considered a harmonicity descriptor.

We suggest a modification of the already existing Octave Based Spectral Contrast (OBSC) [19] and Shape-Based Spectral Contrast (SBSC) [20], both of which were successfully used for a music genre classification task. In [19], the authors suggest the following procedure to compute OBSC: Each frame of audio is transformed into the frequency domain by utilising the DFT. Afterwards, the resulting spectrum is divided into six sub-bands (0-200 Hz, 200-400 Hz, 400-800 Hz, 800-1600 Hz, 1600-3200 Hz, and 3200-8000 Hz), and the bins of each sub-band are sorted according to their magnitude in descending order. The

neighbourhood parameter α is then used to control the percentage of bins that are used to compute a log-scaled mean of the magnitudes at the beginning as well as the end of the sorted spectrum. Those means are referred to as *Peak* and *Valley* respectively, and their difference is the actual Spectral Contrast feature. The final feature vector is then composed from both the Spectral Contrast and the Valley for each band separately, resulting in 12 attributes. Additionally, a Karhunen-Loeve Transform (KLT) is applied to de-correlate the raw feature values, which requires the training of orthogonal base vectors.

SBSC itself is a modification of the OBSC, to increase robustness and appropriateness of the spectral contrast representation, and the following procedure is suggested in [20]: Similar to the computation of OBSC, every frame is transformed with the DFT, and the resulting spectrum is divided into sub-bands, but with a different scheme. For six bands, this results in boundaries at 20 Hz, 330 Hz, 704 Hz, 1256 Hz, 2303 Hz, 4729 Hz, and 11 kHz. Equation 1 is then suggested to calculate Spectral Contrast, where k is the index of the sub-band, μ is the mean of the whole sub-band, P and V the peak and valley values respectively.

$$C_k = \left(\frac{P_k}{V_k} \right)^{1/log\mu_k} \quad (1)$$

By including the mean of the sub-band μ in the equation, the characteristic of the sorted spectrum between peak and valley is also considered. This allows to differentiate between shapes that would result in the same Spectral Contrast, when computed with the procedure for OBSC. The final feature vector is then composed from both the Spectral Contrast and the Valley for each band separately, resulting in 12 attributes. Compared to OBSC, the KLT is also applied for de-correlation, but the required covariance matrices are not based on the complete data set, but computed for each individual song.

In our modification, we simplify this approach by removing the necessity to compute both peak and valley values, as well as removing the neighbourhood parameter α. Similar to the previously discussed methods, we transform every frame of audio (window length=800 ms, symmetrically placed around a 200 ms frame) with a DFT, and subdivide the resulting spectrum in six bands (using the same boundaries as with OBSC), and sort the bins of every sub-band according to their magnitudes in descending order. Afterwards, we fit a third-order polynomial to the resulting shape. Therefore, we refer to this feature as *Polynomial Shape Spectral Contrast (PSSC)*. It is computed for every band, yielding a feature vector with 24 attributes (we include the offset of the resulting polynomial). Contrary to the procedure to compute OBSC and SBSC, we don't utilise any de-correlation measures to the raw features, hence reducing the complexity even further. According to the results of a four-fold cross validation on our internal data set (see Sec. 4.1), the suggested approach outperforms OBSC and SBSC, despite its simplicity. OBSC performed at 0.753, SBSC at 0.761, and the suggested PSSC at 0.803 F-measure for the class of interest, *speech*. More traditional features like PLP, MFCC, and LPC reached F-measures of 0.725, 0.664, and 0.720 respectively, hence supporting the choice of PSSC even further.

3.4. Complete Feature Set and Classifier

Our final feature vector contains 63 attributes, 13 Fluctogram variances, 13 Spectral Flatness means, and 13 Spectral Contraction variances, and 24 PSSCs. The units of audio to be classified are 200 ms frames, resulting in five classifications per second. As classifier, we choose the SVM implementation of the Weka toolkit [21] ($complexity = 0.4, \gamma = 0.9$).

4. Datasets

4.1. Internal Data Set for Training

In order to determine a proper feature set and classifier parameters, we use an internal set of 4 h radio broadcasts from four different stations, exactly 1 h each. The amount of speech is 75 min (31.3%), and the language spoken is mainly German with occasional English. The decision to use this was based on two considerations: First, we do not have ground truth annotations for additional in-domain data besides the movies, which will be described in the next section, and radio broadcasts are the next best thing available to us. Second, they allow for an – at least coarse – estimation of the capability to discriminate speech and singing voice, since both are an integral part of radio.

4.2. Movie Data Set

We manually annotated the ground truth on four full-length movies in the original English versions by labeling *speech* and *non-speech* segments as precisely as possible. These annotations were used exclusively for testing the final detector, and are made openly available to the research community. In Table 1, the statistics regarding length and speech content are given. We also include the statistics reported from the authors of the baseline in [6] (col. EYB) to demonstrate the difficulties in obtaining a proper ground truth: clearly, their amount of speech is much higher than in our annotations (col. NEW), most certainly due to the coarse annotation style, which the authors mention. Depending on what the annotator considers speech, screaming and singing could possibly also be labeled as speech. Therefore, it is important to provide the annotator with a set of rules, preferably with an application already in mind. Unfortunately, although the authors responded quickly to our requests, and were very supportive by offering to share the annotations, we could not get our hands on their data in time to include a more detailed analysis of the differences. Additionally, there seem to be differences regarding the length of the movies. Obviously, if one wants to use the annotations provided by us, the audio tracks need to be extracted from the same version of the movie to assure proper aligned segment boundaries.

While annotating, we followed a set of rules, mainly with an ASR system in mind. Thus, we did not annotate singing, non-articulated screams, laughing, and breathing as speech. We tried to annotate speech, as soon as we were able to recognise it as such, and it would make sense to show subtitles along, even when it was barely perceivable. Pauses, even small ones, were annotated as often as possible.

5. Results

We present the results of two experiments. First, we perform a leave-one-out cross-validation (CV) on the before mentioned internal data set of 4 h radio broadcasts, with the baseline method (Eyben et al. [6], as well as our suggested approach. The thresholds resulting in equal false negative rate (FNR), and false positive rate (FPR), are then fixed and left unchanged for the second experiment. Here, we train the baseline system, and our novel approach with the complete (previously splitted for cross-validation), 4 h radio data set (see Sec. 4.1). The resulting classifiers are then presented with previously unseen audio

	[hh:mm:ss]		speech [%]	
	EYB	NEW	EYB	NEW
Bourne Id.	1:53:–	1:58:24	40.7	26.7
I Am Leg.	1:36:–	1:40:22	39.2	18.3
Kill Bill 1	1:46:–	1:46:08	33.9	19.2
Saving P.	2:42:–	2:42:27	48.6	32.1
ALL	7:57:–	8:07:21	41.6	25.2

Table 1: Statistics of the four full-length movies, used as challenging test set. Clearly, the amount of speech we annotated (col. NEW) differs from what was reported in [6] (col. EYB)

	AUC	FNR	FPR	EER	ACC	PREC	REC	F
EYB	.857	.202	.202	.202	.798	.643	.799	.712
NEW	.979	.059	.059	.059	.941	.880	.941	.910

Table 2: CV-Results of the proposed method on four radio broadcasts, compared to those of Eyben et al. in [6], always trained and tested with the same audio data. A fixed threshold corresponding to the EER is used.

tracks from four movies (see Sec. 4.2).

5.1. Cross Validation Results on Internal Data Set

In Table 2 the results of the leave-one-out CV are listed. The columns AUC, FNR, FPR, and EER list the area under ROC curve, false negative rate, false positive rate, and equal error rate, respectively. Furthermore we added the results regarding accuracy, precision, recall, and F-measure in the columns ACC, PREC, REC, and F respectively. The row EYB contains the results from the baseline method [6], and row NEW contains the results from our proposed method.

Compared to the results of our suggested method, one can observe a lower FPR in general (20.2% vs. 5.9%). This supports our claim, that the feature set we propose, is indeed better capable to discriminate between speech and other harmonic sources like vocals and music. The high amount of the latter two classes in radio broadcasts reveals such a potential weakness.

From previously conducted experiments, we experienced that RASTA-PLPs (as used in the baseline method) are not suited to discriminate between speech, singing voice and some instruments, even pure piano music produces some false positives. In [6], the authors of the baseline method also report a relatively high combined error rate (FNR+FPR=31%) on their synthetic test set for the segments, where they mixed speech and music.

5.2. Results on Movie Data Set

In Table 3, we can see the results on the previously unseen four full-length movies. The columns EYB and NEW refer to the baseline [6] and our proposed method respectively. The rows list the results of the four movies, as well as the averaged results (row ALL), weighted by the length of the movies respectively.

Compared to the results on the cross-validated radio broadcasts, the performance of both methods drops considerably. This is expected, and partly due to out-of-domain training, and was also reported by the authors of the baseline method [6], where the EER dropped from 10.4% on the synthetic, unseen test set down to 33.2% on the same four movies.

The proposed approach outperforms the baseline in every aspect, reducing the overall equal error rate (col. EER, row

	AUC		FNR		FPR		EER	
	EYB	NEW	EYB	NEW	EYB	NEW	EYB	NEW
Bourne Id.	.658	.897	.720	.297	.094	.074	.262	.134
I Am Leg.	.688	.922	.655	.225	.098	.062	.200	.092
Kill Bill 1	.688	.914	.480	.186	.240	.109	.286	.124
Saving P.	.644	.880	.707	.306	.127	.084	.313	.155
ALL	.652	.895	.665	.271	.139	.082	.271	.130
	ACC		PREC		REC		F	
	EYB	NEW	EYB	NEW	EYB	NEW	EYB	NEW
Bourne Id.	.738	.866	.520	.776	.280	.703	.364	.738
I Am Leg.	.800	.908	.443	.738	.345	.775	.388	.756
Kill Bill 1	.715	.877	.340	.640	.521	.814	.411	.717
Saving P.	.687	.845	.522	.797	.293	.695	.375	.742
ALL	.729	.870	.447	.748	.335	.729	.383	.738

Table 3: Results of the proposed method on four full-length movies, compared to those of Eyben et al. in [6], trained with the same audio data. Thresholds for EER were ascertained after cross-validation of the training set, hence FPR and FNR are not equal. Results in row ALL are length-weighted.

ALL) to less than half (27.1% vs. 13%). In terms of FNR, the movies Bourne Identity and Saving Private Ryan seem to be the most challenging. This is due to the high amount of noise, like shooting, tanks or cars driving, and rain, which deteriorates speech to an extend, that it is often not recognised as such. In our opinion, simply adding similar examples to the training set would not solve this problem. We think, specific sets of features need to be composed, to specialise on smaller problems, currently not covered by the proposed method. To give an example, we see room for improvement for screaming and whispering, which is clearly different than speech w.r.t. harmonicity. On the other hand, one has to be careful with the design of a possible solution, since every specific detector added to the system holds a potential risk to further increase the FPR (e.g. whispering vs. wind).

6. Conclusions

We have presented a set of light-weight acoustic features, specifically designed for VAD in the challenging conditions in movies. First, we performed a CV on four different radio broadcasts to demonstrate the capability to detect speech without confusing it with singing voice or music. With the result of this CV, the threshold at the EER was fixed.

Second, we trained a SVM with the complete set, previously used in the first experiment, and reuse the threshold to make sure the test set is left unseen. As test set, we use four different movies, and compare the results to a state-of-the-art baseline system. The new approach outperforms the baseline, yielding just about half the EER (27.1% vs. 13%).

Furthermore, since the ground truth annotation task is extremely tedious, and to help with advancing in this topic, we release the annotations of all four movies to the research community.

7. Acknowledgements

This research is supported by the Austrian Science Fund (FWF) under grants TRP307-N23 and Z159.

8. References

[1] P. Zelinka, M. Sigmund, and J. Schimmel, "Impact of vocal effort variability on automatic speech recognition," *Speech Communication*, vol. 54, no. 6, pp. 732–742, 2012.

[2] G. Evangelopoulos and P. Maragos, "Speech event detection using multiband modulation energy." in *Proceedings of INTERSPEECH 2005*. ISCA, 2005, pp. 685–688.

[3] B. Kotnik, Z. Kacic, and B. Horvat, "A multiconditional robust front-end feature extraction with a noise reduction procedure based on improved spectral subtraction algorithm." in *Proceedings of INTERSPEECH 2001*. ISCA, 2001, pp. 197–200.

[4] M. Moattar and M. Homayounpour, "A simple but efficient real-time voice activity detection algorithm," in *Proceedings of the 17th European Signal Processing Conference, EUSIPCO 2009*. IEEE, 2009, pp. 2549–2553.

[5] V. Hautamäki, M. Tuononen, T. Niemi-Laitinen, and P. Fränti, "Improving speaker verification by periodicity based voice activity detection," in *Proceedings of the 12th International Conference on Speech and Computer, SPECOM 2007*, 2007, pp. 645–650.

[6] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, "Real-life Voice Activity Detection with LSTM Recurrent Neural Networks and an Application to Hollywood Movies," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2013*. IEEE, 2013, pp. 483–487.

[7] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *Signal Processing Letters, IEEE*, vol. 6, no. 1, pp. 1–3, 1999.

[8] J. Ramírez, J. C. Segura, C. Benítez, L. García, and A. Rubio, "Statistical voice activity detection using a multiple observation likelihood ratio test," *Signal Processing Letters, IEEE*, vol. 12, no. 10, pp. 689–692, 2005.

[9] S. Mousazadeh and I. Cohen, "AR-GARCH in Presence of Noise: Parameter Estimation and Its Application to Voice Activity Detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 916–926, 2011.

[10] M. A. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, "Buckeye corpus of conversational speech (2nd release)," *Columbus, OH: Department of Psychology, Ohio State University*, 2007.

[11] J. S. Garofolo, L. D. Consortium *et al.*, *TIMIT: acoustic-phonetic continuous speech corpus*. Linguistic Data Consortium, 1993.

[12] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor," in *Proceedings of the 18th International Conference on Multimedia, MM 2010*. ACM, 2010, pp. 1459–1462.

[13] A. Graves, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[14] B. Lehner, G. Widmer, and R. Sonnleitner, "On the Reduction of False Positives in Singing Voice Detection," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2014*. IEEE, 2014, pp. 7530–7534.

[15] K. Laskowski and Q. Jin, "Modeling instantaneous intonation for speaker identification using the fundamental frequency variation spectrum," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009*. IEEE, 2009, pp. 4541–4544.

[16] B. Ma, D. Zhu, and R. Tong, "Chinese Dialect Identification Using Tone Features Based on Pitch Flux," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2006*, vol. 1. IEEE, 2006, pp. 1029–1032.

[17] R. Sonnleitner, B. Niedermayer, G. Widmer, and J. Schlüter, "A Simple And Effective Spectral Feature For Speech Detection In Mixed Audio Signals," in *Proceedings of the 15th International Conference on Digital Audio Effects, DAFx 2012*, 2012.

[18] A. Gray Jr. and J. Markel, "A spectral-flatness measure for studying the autocorrelation method of linear prediction of speech analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 3, pp. 207–217, Jun 1974.

[19] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proceedings of the International Conference on Multimedia and Expo, ICME 2002*, vol. 1. IEEE, 2002, pp. 113–116.

[20] V. Akkermans, J. Serrà, and P. Herrera, "Shape-based spectral contrast descriptor," in *Proceedings of the 6th Sound and Music Computing Conference, SMC 2009*, 2009, pp. 143–148.

[21] M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.