ROBUST SPEECH ACTIVITY DETECTION IN MOVIE AUDIO: DATA RESOURCES AND EXPERIMENTAL EVALUATION

Rajat Hebbar, Krishna Somandepalli, Shrikanth Narayanan

Signal Analysis and Interpretation Laboratory, University of Southern California, Los Angeles
rajatheb@usc.edu, somandep@usc.edu, shri@ee.usc.edu

ABSTRACT

Speech activity detection in highly variable acoustic conditions is a challenging task. Many approaches to detect speech activity in such conditions involve an inherent knowledge of the noise types involved. Movie audio can offer an excellent research test-bed for developing speech activity models. A robust speech detection in movie audio is also a crucial step for subsequent content analyses such as audio diarization. Obtaining labels for supervision of such data can be very expensive, and may not be scalable. In this paper, we employ a simple, yet effective approach to obtain speech labels for movie data by coarse aligning the subtitles with movie audio. We compiled a dataset, called Subtitle-aligned Movie Corpus (SAM) of nearly 23 hours of data labelled as speech from ninety-five Hollywood movies. We propose convolutional neural network architectures that use log-mel spectrograms as input features to predict speech at a segment-level, as opposed to frame-level. We show that our models trained on SAM outperform existing baselines on two independent, publicly released movie speech datasets. We have made the SAM corpus and pretrained models publicly available for further research.

Index Terms— Speech activity detection, movie audio, convolutional neural networks

1. INTRODUCTION AND BACKGROUND

Audio streams from movies represent a rich source of data for developing and testing technologies such as speech activity detection (SAD) and speaker/audio event diarization. Such technologies can aid and scale up media content analyses. Some example applications include understanding speaker representations in movies [1], analysing content via social network constructs [2] and constructing story narratives. This enables a more robust infrastructure for improving information retrieval and content analysis.

The specific area of application that we are interested in this paper is toward estimating speaking time with respect to attributes such as gender and age in movie audio. There has been a growing interest in examining movie data by looking at speaking time from audio and screen-time from video for the characters in a movie with respect to gender, age, etc. This analysis at scale helps us understand gender and race diversity and inclusion [1]. For example, *female speaking time*–an estimate of the amount of time a female person speaks in a movie, has been analyzed from the top Box Office grossing movies over years[1]. A first step toward obtaining reliable estimates of such measures at scale is to be able to extract the speech regions robustly, for subsequent modeling such as gender identification in movies [3].

Speech Activity Detection (SAD) is the task of automatically detecting speech and non-speech regions in a given audio segment. This is a vital preprocessing step for many downstream applications such as speaker recognition [4] and ASR [5] as these systems operate over the detected speech regions in an audio stream. For these tasks, SAD is not the final objective, but often a first step to obtain speech segments from audio. As such, errors in speech detection can propagate through the pipeline which can impact the final accuracy of the overall system resulting in sub-optimal performance.

Although SAD has been extensively studied for tasks such as speaker recognition (See [4] for detailed surveys), most of these methods work with speech where the background noise is known [6, 7, 8]. SAD is arguably a straightforward task in case of clean speech with low background noise. Simple energy based methods [9, 10] are often sufficient for SAD in this context. However, SAD remains a challenging task in several domains where the acoustic conditions are highly variable and unknown and the noise conditions are generally not stationary [11, 12, 13].

Our domain of interest in this work is movies, specifically Hollywood movies. Movie audio consists of varied acoustic backgrounds that include ambient noise, music, environmental sounds, etc, all acquired under different contexts. Additionally, movie audio is edited during post-production stage of a movie[14], and this sound editing and mixing is stylistically motivated, often to induce certain emotions in the viewer. Furthermore, speech in movies often differs from regular conversational speech due to presence of atypical speech such as whispering, shouting, singing, and electronically modified speech. These factors make SAD a challenging task for movie audio, and drive the need for models to be trained and evaluated on domain-matched data. As a by-product, movie audio offers a challenging test-bed for speech technology research and development.

There have been two notable works in the context of SAD for movie data [15, 16]. Both of these have enabled the release of detailed SAD labels for two distinct movie datasets. Manually annotated labels on a set of four hollywood movies were released in [15]. They proposed a SVM based classifier trained on 63-dimensional hand-crafted spectral features. The training set used here was radio-data. Speech labels for a set of 160 movies available on YouTube were released by [16], with manually-annotated labels provided for about 15–30 min of each movie. They also provided audio quality labels for speech segments and presented results on convolutional neural network (CNN) models trained on mel-spectrogram features.

Our contributions in this work are: 1) We compile a dataset, SAM of about 117 hours of movie audio (~20% speech), aligned with the subtitles to obtain coarsely labelled speech and non-speech regions. 2) we propose CNN-based architectures to show that models trained on this dataset outperform existing methods on benchmark test data, and 3) we have released the audio features, pretrained models and related code for future research in this domain[2].

[1] seejane.org/research-informs-empowers/data

[2] github.com/usc-sail/mica-speech-activity-detection

2. MOVIE DATASETS

In this section, we first describe the details for generating speech and non-speech labels for the subtitle-aligned movie (SAM) corpus. We also describe the noise-augmented speech data that we created to test on movie data, followed by the benchmarking datasets where we evaluate all our models.

2.1. Subtitle-Aligned Movie (SAM) Corpus

SAM corpus comprises of ninety-five movies[3] from the year 2014 (movies purchased in-house). For transcripts, we used subtitles generated automatically[4]. These subtitles provide a list of approximate starting and ending time-stamps corresponding to a single utterance/dialogue. Some time-stamps also correspond to certain sounds which could be either vocal/non-vocal (e.g, shrieking, sobbing, bell ringing, door banging). These sounds are typically enclosed in parentheses in the subtitle-file. Since our goal is speech activity detection, we treat these sounds as non-speech data. The data was split into 82 movies for training and 13 for validation.

We extracted *labels* for speech/non-speech regions as follows:
1) **Non-speech labels:** We obtained audio segments between two successive timestamps corresponding to a speech utterance in a subtitle file. We also included audio regions with vocalized/non-vocalized sounds labelled in the subtitles.
2) **Speech labels:** We used an open-source Kaldi [17] based speech-to-text alignement tool, Gentle[5] to align speech segments at word-level given the subtitles of a movie.

Both the subtitle-generation and alignment are completely automated. Hence, they are prone to spelling-errors in the subtitles that may lead to to failure in alignment. One way to measure the completeness of gentle alignment is to examine the percentage of words from the subtitles that were successfully aligned. Overall, we were able to successfully align 76.4 ± 8.6 percentage of words across the 95 movie subtitles using our proposed system.

Next we obtained speech and non-speech *segments* as follows: Speech regions corresponding to consecutive gentle-aligned words were accumulated to form segments of length t_{seg}. First, a heuristic threshold of t_{break} seconds (duration of pause) was used to chunk consecutive aligned words into inter-pausal units (IPU) (e.g., [18]). Hence, two consecutive aligned words were considered to belong to the same IPU if they were no farther than t_{break} seconds apart. Finally, these IPUs were split into non-overlapping segments of t_{seg} seconds each. For our experiments, we chose $t_{seg} = 1.28s$ and $t_{break} = 0.5s$. We obtained a total of around 63,500 speech segments for this dataset. Similarly, we split the non-speech regions into segments of length 1.28 seconds. We used segments of length 0.64s to extract features in order to have square (64x64) input features to CNN models for convenience. More details on the number of words aligned and number of segments at different t_{seg} and t_{break} can be found here[6].

2.2. Noise augmented data: MUSAN + Audioset noise

Typically, noise-robust SAD systems have been developed using audio from a clean-speech dataset corrupted with different noise types (e.g., [7]). Thus a competitive baseline to compare the performance of models trained using SAM corpus would be to recreate them on noise-augmented data. For this purpose, we used MUSAN corpus [19], which consists of music, speech and noise data from various sources. For clean speech, we used 20 hrs of audiobook data from MUSAN. We augmented this data (by a factor of 3) by re-sampling the speech at 0.9 and 1.1 times the original sampling rate following recommendations in [20]. Thus we created a total of 60 hrs of speech data. For non-speech data, we used music without lyrics (~24 hrs) and noise data (~6hrs) from MUSAN. Additionally, we randomly sampled 15hrs of noise data from four Audioset categories (animal sounds, sounds of things, natural sounds and background)[7]. The 60 hours of speech was noise-augmented with the non-speech data at seven SNR levels of -15, -10, -5, 0, 5, 10 and 15 dB.

2.3. Current movie benchmarking datasets

Hollywood movie dataset:
A set of four movies (I am Legend (2007), Kill Bill Vol I (2003), Saving Private Ryan (1998) and The Bourne Identity(2002)) were first used by [7] and then [15] as benchmarks for voice activity detection (VAD) in movies. Ground truth voiced region annotations were released in [15] with around 2 hrs of speech and 6 hrs of non-speech. There is a subtle difference between VAD and SAD. Not all voiced utterances can be considered as speech (e.g, shriek, groan). It is important to note that there was no such distinction made in [15] during annotation. Hence, some of the labels may correspond to such voiced utterances. The dataset consists of speech in the presence of a wide range of acoustic backgrounds (e.g, gun-fire, rainstorms, loud music) and hence a good benchmark for the movie domain.
AVA-speech:
A set of 192 movies available on YouTube were manually annotated for 3 speech classes (clean speech, speech+music, speech+noise) and no-speech [16], out of which annotations for 160 movies were released. The dataset as reported in the publication [16] had 40 hours of audio with 40,000 annotated speech/non-speech segments. However, at the time we started our experiments (August 2018) only 147 of the 160 movies were available on YouTube (~37 hours). We present our results on this dataset of ~37 hours assuming that it is a representative sample of the original dataset.

3. METHODS

Features:
In all our experiments, we use log-mel filterbank energies (log-mel) as features. Log-mel features differ from the traditional mel frequency cepstral coefficient (MFCC) features by a single discrete-cosine transform (DCT). The purpose of DCT is to decorrelate spectral features to compress them, often to a lower dimension. Due to the advances in computational power and memory, as well as the ability of CNN models to utilize correlated features, log-mel features have become popular recently [16, 21, 22].

We used 23 dimensional log-mel features (extracted using Kaldi [17] with default parameters) for a BLSTM-based baseline experiment trained with frame-level data. Since we use a dual-context of 15 frames, we chose to restrict the feature dimension to 23. For the CNN experiments, we used 64 dimensional features extracted for 64 frames (640ms), resulting in a square input feature.

3.1. Network Architectures

We briefly discuss two popular neural network architectures adopted in our work, i.e, recurrent and convolutional networks. We train all

[3] boxofficemojo.com/yearly/chart/?yr=2014
[4] github.com/ruediger/VobSub2SRT
[5] github.com/lowerquality/gentle
[6] github.com/usc-sail/mica-speech-activity-detection/blob/master/sam_stats.xlsx
[7] research.google.com/audioset/ontology/index.html

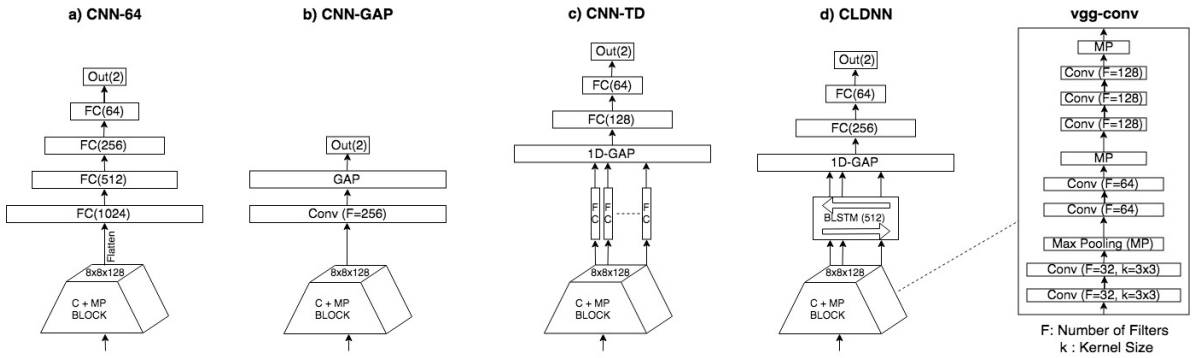

Fig. 1. Block description of CNN architectures trained on SAM.

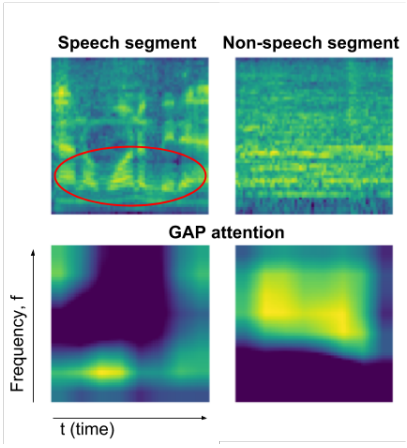

Fig. 2. Visualizing attention for speech and non-speech segments.

our models using the python-based Keras API and TensorFlow.

BLSTM architecture: LSTM is a popular neural network architecture for speech-related tasks (e.g., [7]) due to its ability to model both short-term and long-term context in speech extremely well. Bidirectional-LSTM (BLSTM) is an extension of LSTM, wherein both forward and backward context are utilized.

We trained a BLSTM network with log-mel features. We denote this model as **blstm-23**. The network consists of a single BLSTM layer with 300 nodes (150 in each direction), followed by a sequence of fully connected (FC) layers with 256, 128 and 64 nodes respectively (ReLu activation). The final layer has 2 nodes with softmax activation for speech/non-speech classification.

CNN-based architectures: CNNs have recently been shown to be extremely powerful for image based classification tasks. Popular CNN architectures such as VGG [23] and Res-net [24] have been replicated with some modifications for various speech-based tasks with impressive results [16, 25, 26].

We present four CNN-based architectures in our work, each trained on 64x64 segment-level features. All architectures have the same modified VGG convolutional block (**vgg-conv**). See **Fig.** 1 for details on the number of layers, filters, strides and kernel size. The output of the **vgg-conv** is of dimension 8x8x128 (TxFxC), where T is the number of time-frames, F, the number of frequency bins and C, the number of filters of the final layer (channels).

Our first architecture (**cnn-64**) uses FC layers on the flattened output to perform SAD. Motivated by the success of class activation mapping for object localization [27, 28], we used a global average pooling (GAP) layer in our 2nd architecture (**cnn-gap**). For the final two architectures, we flatten the FxC dimensions of **vgg-conv**. For **cnn-td** (See Fig. 1), we used a time distributed (TD) FC layer, and for **cldnn** we used a BLSTM layer. We then performed temporal pooling (1D-GAP) in both **cnn-td** and **cldnn** before input to FC layers for classification. We used batch normalization (BN) after each convolutional/FC layer (before applying ReLu). CLDNNs have been effectively used for other speech-related tasks [22, 29, 30].

Our design choices in the **cnn-td** architecture were motivated to model frame-level predictions when training on segment-level data. The convolutional blocks effectively capture local spatial context. Complementary to this, the time-distributed FC layers, which share their weights, can be viewed as analogous to performing "frame-level" operations – which are then aggregated in the 1D-GAP layer.

4. EXPERIMENTS

In this section, we discuss the experiments performed and parameters chosen for training. We used binary cross-entropy as our loss criterion with Adam optimizer. We used a batch-size of 50 for BLSTM, and 64 for the CNN experiments. We trained the network for five epochs and then applied early stopping criteria (stop training if validation loss does not decrease by 1e-3 for 3 consecutive epochs).

For the baseline **blstm-23** model, we tuned the number of BLSTM nodes, number of FC layers/nodes, BN and dropout on the validation set. For the CNN architectures, we experimented with filter shapes of 3x3, 5x5 and full-spectrum rectangular filters [22]. We found that 3x3 filters performed best. Finally, we tuned the number of nodes in the BLSTM layer for the **cldnn** model, and the number of TD-FC layers and nodes in the case of the **cnn-td** model.

All models were trained on the SAM corpus. We first tested the models on the Hollywood movie dataset. We compared results obtained using the baseline **blstm-23**, and the four CNN models – **cnn-64**, **cnn-gap**, **cnn-td** and **cldnn**, with the performance reported for the four movies in [15]. Due to class imbalance (see Sec. 2.3), we used F1-score as the metric to compare performance.

We then tested our models on AVA-speech dataset. In order to compare our results with those reported in [16], we examine TPR (or recall) at an FPR=0.315. We also look at the ROC curve to analyze the recall rate of the models for different FPR values.

A widely used approach for SAD is to train a model with clean speech that is artificially corrupted with different noise types. We

Models	Accuracy	Precision	Recall	F1 score
Lehner et al. [15]	0.87	0.75	0.73	0.74
blstm-23	0.87	0.8	0.63	0.7
cnn-64	0.86	0.69	**0.83**	0.75
cnn-gap	0.87	**0.82**	0.65	0.73
cnn-td	0.88	0.74	0.81	**0.77**
cldnn	**0.89**	0.79	0.75	0.77

Table 1. Results on the 'four movie' dataset (Sec. 2.3), averaged across 4 movies.

trained a version of the **cnn-td** model with the noise-augmented dataset (described in Sec. 2.2). This experiment allowed us to assess the effectiveness of SAM corpus for detecting speech in movies.

Finally, we performed an experiment to justify our approach of predicting SAD labels at segment level vs. frame-level. For this, we generated predictions for our best performing model (**cnn-td**) with overlapping input segments. We then performed majority voting to decide the SAD label for a frame spanned by multiple windows. We varied the percentage overlap in the range of 12.5, 25.0,...,87.5 to examine the effect of this parameter on frame-level performance.

5. RESULTS AND DISCUSSION

5.1. Hollywood movie dataset

The **blstm-23** model falls short of the baseline in [15] in terms of F1-score (Table 1). The CNN models outperform the **blstm-23** baseline (F1: 77% vs. 70%). However, the overall performance with respect to [15] is comparable (F1: 77% vs. 74%). This could be attributed to the fact that our models were trained for the specific task of SAD, whereas, annotations provided in [15] may include voiced-segments (not necessarily speech) labelled as speech. The CNN models show an increase in recall as compared to the baseline models. Since we allow up to 0.5s of noise/silence in a speech segment of length 0.64s (see Sec. 2.1), the network can learn to detect short utterances within a segment, and thus be robust to silence/noise within these segments.

5.2. AVA speech dataset

The **blstm-23** model performs slightly worse overall than the tiny_320 architecture (Table 2), despite performance gains on speech segments with music. Our CNN models outperform [16] in terms of overall SAD performance. The improved performance is especially noticeable in segments of speech with music. Since many movies include speech with background music, our models trained on SAM can robustly detect speech in the presence of music. This highlights one of the benefits of using domain-specific data for training. One caveat, however, is that the models in [16] were trained on over 500 classes (from which they pick speech class outputs), whereas we trained for the specific task of SAD. This distinction reflects in the shape of the ROC curves in Fig. 3, and those shown in [16]. Furthermore, the results of our models are not sensitive to the fixed FPR value. For example, the **cnn-td** model achieves a recall of 0.917 (~resnet_960) at a lower FPR of 0.22.

The performance of the model trained on the noise-augmented dataset (acc: 72% and F1: 59% on Hollywood movies, 0.77 TPR for FPR=0.315 on AVA-speech) is significantly lower than that of the CNN models trained on SAM. This further justifies the need for domain-matched data for SAD in movies.

We notice a slight positive trend in performance as we increase the overlap percentage from 12.5% (87.4% accuracy and 0.75 f1-score) to 87.5% (88.6% accuracy and 0.77 f1-score). However, this performance gain may not justify the increased number of inferences associated with multiple overlaps per frame.

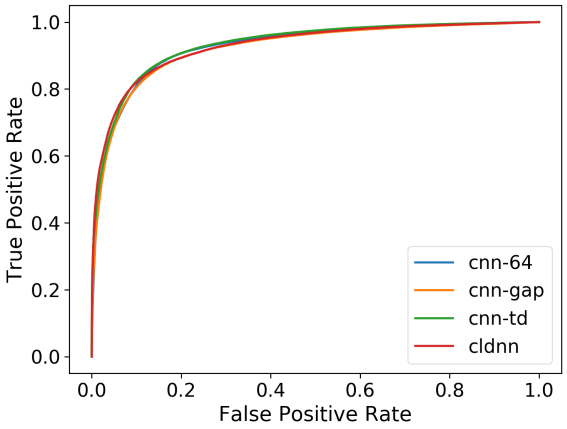

Fig. 3. ROC curve for CNN models evaluated on AVA-Speech

Models	Num Parameters	TPR for FPR = 0.315			
		Clean	+Noise	+Music	All
tiny_320* [16]	<1M	0.965	0.826	0.623	0.810
resnet_960* [16]	30M	**0.992**	**0.944**	0.787	0.917
blstm-23	300K	0.846	0.76	0.704	0.769
cnn-64	9.5M	0.988	0.933	0.91	0.942
cnn-gap	730K	0.986	0.934	0.874	0.933
cnn-td	**740K**	0.983	0.939	**0.917**	**0.945**
cldnn	1M	0.985	0.922	0.906	0.935

Table 2. True positive rate of our models on the AVA-speech dataset[10]

In Fig. 2, we visualize the attention of the **cnn-gap** model for given input segment (speech vs non-speech) using Grad-CAM [28]. The attention over speech/non-speech segments is evidently distinct. The model attends to lower frequency regions in the case of speech. These regions consist of formants, where most of the energy in speech signals is typically concentrated (circled in Fig. 2). However, in the case of non-speech, the attention is not sparse spatially. Moreover, in the case of non-speech, the model attends more to the higher frequency regions in the log-mel domain. This is consistent with the frequency ranges associated with speech/non-speech.

It is important to note that we did not conduct additional experiments to disambiguate the performance gains we have with respect to our corpus and the CNN architectures used in the AVA dataset (e.g., train resnet_960 on SAM). This will be part of our future work. We have also made our training data, features and code publicly available for the benefit of the research community[8].

6. CONCLUSION

We adopt a method of procuring segment-level speech labels from movie audio, without the need of manual annotations for robust SAD. We train CNN models on log-mel features for this data, and show results competitive with the state-of-the-art for two independent movie benchmarks. On probing the CNN models trained for SAD using attention mechanism, we observed results consistent wih time-frequency distributions of speech in log-mel domain.

[8] github.com/usc-sail/mica-speech-activity-detection

7. REFERENCES

[1] Tanaya Guha, Che-Wei Huang, Naveen Kumar, Yan Zhu, and Shrikanth S. Narayanan, "Gender representation in cinematic content: A multimodal approach," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, Seattle, WA, USA, November 09 - 13, 2015*, 2015, pp. 31–34.

[2] Chung-Yi Weng, Wei-Ta Chu, and Ja-Ling Wu, "Rolenet: Movie analysis from the perspective of social networks," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 256–271, 2009.

[3] Rajat Hebbar, Krishna Somandepalli, and Shrikanth Narayanan, "Improving gender identification in movie audio using cross-domain data," *Proc. Interspeech 2018*, pp. 282–286, 2018.

[4] Md Sahidullah and Goutam Saha, "Comparison of speech activity detection techniques for speaker recognition," *arXiv preprint arXiv:1210.0297*, 2012.

[5] Man-Wai Mak and Hon-Bill Yu, "A study of voice activity detection techniques for nist speaker recognition evaluations," *Comput. Speech Lang.*, vol. 28, no. 1, pp. 295–313, Jan. 2014.

[6] Javier Ramırez, José C Segura, Carmen Benıtez, Angel De La Torre, and Antonio Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech communication*, vol. 42, no. 3-4, pp. 271–287, 2004.

[7] 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 *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 483–487.

[8] Mohammad H. Moattar and Mohammad M. Homayounpour, "A simple but efficient real-time voice activity detection algorithm," *2009 17th European Signal Processing Conference*, pp. 2549–2553, 2009.

[9] Sree Hari Krishnan P, R. Padmanabhan, and Hema A Murthy, "Robust voice activity detection using group delay functions," *2006 IEEE International Conference on Industrial Technology*, 2006.

[10] Georgios Evangelopoulos and Petros Maragos, "Speech event detection using multiband modulation energy," in *INTERSPEECH*, 2005.

[11] Neville Ryant, Mark Liberman, and Jiahong Yuan, "Speech activity detection on youtube using deep neural networks," in *INTERSPEECH*, 2013.

[12] Jianjun Lei, Jiachen Yang, Jian Wang, and Zhen Yang, "A robust voice activity detection algorithm in nonstationary noise," *2009 International Conference on Industrial and Information Systems*, pp. 195–198, 2009.

[13] Waheeduddin Q. Syed and Hsiao-Chun Wu, "Speech waveform compression using robust adaptive voice activity detection for nonstationary noise," *EURASIP J. Audio, Speech and Music Processing*, vol. 2008, 2008.

[14] Elisabeth Weis and Belton John, *Film Sound - Theory and Practice*, Columbia University Press, 1985.

[15] Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner, "Improving voice activity detection in movies," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] Sourish Chaudhuri, Joseph Roth, Daniel P. W. Ellis, Andrew Gallagher, Liat Kaver, Radhika Marvin, Caroline Pantofaru, Nathan Reale, Loretta Guarino Reid, Kevin W. Wilson, and Zhonghua Xi, "Ava-speech: A densely labeled dataset of speech activity in movies," in *Interspeech*, 2018.

[17] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[18] Brigitte Bigi and Daniel Hirst, "Speech phonetization alignment and syllabification (sppas): a tool for the automatic analysis of speech prosody," in *Speech Prosody 2012*, 2012.

[19] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *CoRR*, vol. abs/1510.08484, 2015.

[20] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[21] Samuel Thomas, Sriram Ganapathy, George Saon, and Hagen Soltau, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2519–2523, 2014.

[22] Che-Wei Huang and Shrikanth Narayanan, "Characterizing types of convolution in deep convolutional recurrent neural networks for robust speech emotion recognition," *CoRR*, vol. abs/1706.02901, 2017.

[23] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] Samuel Albanie, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman, "Emotion recognition in speech using cross-modal transfer in the wild," *arXiv preprint arXiv:1808.05561*, 2018.

[26] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[27] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[28] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

[29] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4580–4584.

[30] Jaejin Cho, Raghavendra Pappagari, Purva Kulkarni, Jesús Villalba, Yishay Carmiel, and Najim Dehak, "Deep neural networks for emotion recognition combining audio and transcripts," *Proc. Interspeech 2018*, pp. 247–251, 2018.