INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

NTCD-TIMIT: A New Database and Baseline for Noise-robust Audio-visual Speech Recognition

Ahmed Hussen Abdelaziz[*]
International Computer Science Institute, Berkeley, USA
ahmedha@icsi.berkeley.edu

Abstract

Although audio-visual speech is well known to improve the robustness properties of automatic speech recognition (ASR) systems against noise, the realm of audio-visual ASR (AV-ASR) has not gathered the research momentum it deserves. This is mainly due to the lack of audio-visual corpora and the need to combine two fields of knowledge: ASR and computer vision. This paper describes the NTCD-TIMIT database and baseline that can overcome these two barriers and attract more research interest to AV-ASR. The NTCD-TIMIT corpus has been created by adding six noise types at a range of signal-to-noise ratios to the speech material of the recently published TCD-TIMIT corpus. NTCD-TIMIT comprises visual features that have been extracted from the TCD-TIMIT video recordings using the visual front-end presented in this paper. The database contains also Kaldi scripts for training and decoding audio-only, video-only, and audio-visual ASR models. The baseline experiments and results obtained using these scripts are detailed in this paper.

Index Terms: Audio-visual speech recognition, audio-visual speech corpus, noise-robustness, visual front-end, ASR

1. Introduction

Many experiments demonstrated that humans' ability to understand speech is slightly affected by noise even without training and under unnatural noise conditions [1, 2]. Despite the breakthroughs achieved by deep learning in the realm of automatic speech recognition (ASR), such systems still can not imitate the competence and noise robustness of human speech recognition. In order to enhance the performance of ASR systems in noisy environments, a non-traditional approach comes from the fact that speech perception is bi-modal (audio-visual). Video recordings of speakers' mouths can be very helpful for ASR systems in uncontrolled noisy conditions, as the visual modality is almost unaffected by acoustic noise.

Although many successful audio-visual ASR (AV-ASR) systems with carefully designed fusion schemes were proposed in the last few decades [3–6], there are still two major challenges that prevent the field of AV-ASR from attracting its deserved research interest. The first challenge is the lack of publicly available audio-visual large vocabulary continuous speech recognition (LVCSR) corpora [7]. Most of the reported AV-ASR experiments have been conducted using small vocabulary tasks such as GRID [8] and CUAVE [9] or using non-released LVCSR corpora like IBM ViaVoice™ [10]. The second major challenge is that conducting AV-ASR experiments requires combining two fields of knowledge: Computer vision and ASR.

To address the first challenge, the authors of [7] have recently released the TCD-TIMIT corpus, which is a free audio-visual continuous speech corpus. Despite the small vocabulary nature of the TCD-TIMIT phone recognition task, applying standard LVCSR approaches to such a corpus can give an insight into the behavior of the explored audio-visual LVCSR methods [11]. Driven by the same motivation in [7], the TCD-TIMIT corpus is extended in this study by creating additional noisy versions of the acoustic signals[1]. The noisy utterances have been generated by mixing the clean signals of TCD-TIMIT with a selection of 6 noise types at a range of signal-to-noise ratios (SNRs). In addition to the noisy utterances, the noisy TCD-TIMIT (NTCD-TIMIT) corpus includes visual features that have been extracted from the video recordings of the TCD-TIMIT speakers. Kaldi [12] scripts for training and testing audio-only, video-only, and audio-visual ASR systems have also been made available in NTCD-TIMIT. The visual features in NTCD-TIMIT will allow ASR researchers to focus on improving acoustic, visual, and fusion models for AV-ASR without worrying about developing a front-end for visual feature extraction. NTCD-TIMIT will also help computer vision researchers to test new visual front-ends using the AV-ASR baseline scripts. In this paper, the NTCD-TIMIT database, visual feature extraction, and the baseline experiments and results are introduced.

The remaining paper is organized as follows: In Section 2, the noisy TCD-TIMIT corpus is presented. Next, in Section 3, the visual front-end is introduced. After a brief summary of the audio-visual fusion models in Section 4, the baseline experiments and results are detailed in Section 5. Finally, the paper is concluded in Section 6.

2. Noisy Audio-visual Speech Corpus

The TCD-TIMIT corpus [7] has been used as the source audio-visual speech for NTCD-TIMIT. TCD-TIMIT is a free newly published audio-visual continuous speech corpus based on the speech material of the TIMIT database [13]. The corpus contains audio and video recordings of 56 speakers with an Irish accent uttering 5488 different TIMIT sentences, i.e., 98 sentences per speaker. It also contains sentences that are uttered by 3 speakers with non-Irish accents and other utterances spoken by 3 professional lip speakers. However, the utterances of those 6 speakers are not used here. Like the TIMIT corpus, the task of the TCD-TIMIT is phone recognition in continuous speech.

In addition to a down-sampled version of the clean TCD-TIMIT utterances, 36 noisy versions have been created. In order to keep the NTCD-TIMIT corpus free for research use like the original TCD-TIMIT, the noise signals have been chosen from noise corpora that have no restrictions on remixing and redistribution. Six noise types have been used: White, babble,

[*] This work was supported by the Federal Ministry of Education and Research (BMBF) through the Fit World Wide "FITweltweit" program, administered by the German Academic Exchange Service (DAAD).

[1] The author would like to show his gratitude to Naomi Harte who has approved republishing the audio files of TCD-TIMIT after mixing them with noise.

http://dx.doi.org/10.21437/Interspeech.2017-860

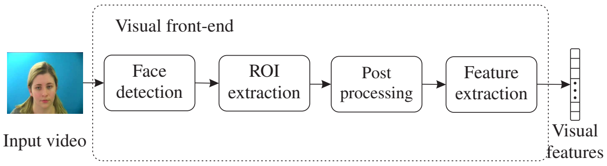

Figure 1: *Common stages of visual feature extraction.*

and car noise from the RSG-10 database[2] [14] (also known as the NOISEX-92 database [15]), living room noise from the second CHiME challenge [16], street and cafe noise from the third CHiME challenge [17]. Each noise type has been added to the clean utterances at 6 SNRs from 20 dB down to −5 dB.

Both the clean and noisy signals of NTCD-TIMIT are sampled at a sampling rate of 16 kHz. Since all noise signals are typically longer than the speech signals, noise segments of the appropriate lengths have been randomly extracted and mixed with the speech signals at the intended SNRs. The noise and speech signals have been mixed at the appropriate SNRs using the filtering and noise adding tool (FaNT) [18] that was used to create the Aurora-2 [19] and Aurora-4 [20] databases. FaNT computes the signal level from the silence-free active part of the speech signals only. The noise level is computed as the root mean squared (RMS) of the extracted noise segments.

The TCD-TIMIT corpus is originally divided into a 70% - 30% training-testing split, i.e., 39 speakers for the training set (almost 5 hours) and 17 speakers for the test set [7]. Here, the test set is further split into an 8-speaker development set and a 9-speaker test set (almost 1 hour each). The development set is reserved for tuning hyper parameters such as stream weights.

3. Visual Front-end

There are no visual features that are commonly used for AV-ASR. Each research group uses different algorithms to extract their own visual features. However, Figure 1 summarizes the four common stages that are mostly included in every visual front-end. The function of the first stage is to localize the speakers' faces in the image sequence acquired from the video stream. From the detected face region, the speech-related part of the face, the region of interest (ROI), is extracted. Next, the ROI is occasionally post-processed by rotating, scaling, and compensating the frame rate difference between the acoustic and visual features. Finally, the visual features are extracted from the post-processed ROI. In the following, the algorithms used in this study to implement each block in Figure 1 are detailed.

3.1. Joint Face and ROI Detection

In this paper, face detection and ROI extraction have been jointly implemented. The algorithm used for achieving these two tasks is composed of three main phases: An initialization, a tracking, and a combination phase.

In the initialization phase, the Viola-Jones algorithm [21] is used to detect bounding boxes (BBs) that circumference the speaker's face and mouth in the first video frame. If a face and a mouth are detected, their BBs are passed to the tracking phase. Otherwise, the current video frame is skipped and the Viola-Jones algorithm is applied to the next frame.

In the tracking phase, two sets of corner points, also known as interest points, are estimated from the image regions determined by the face and mouth BBs using the minimum eigenvalue algorithm [22]. The two sets of corner points are used to initialize two Kanade-Lucas-Tomasi (KLT) point trackers [23]. The face and mouth corner points are independently tracked from a video frame to another using the KLT trackers.

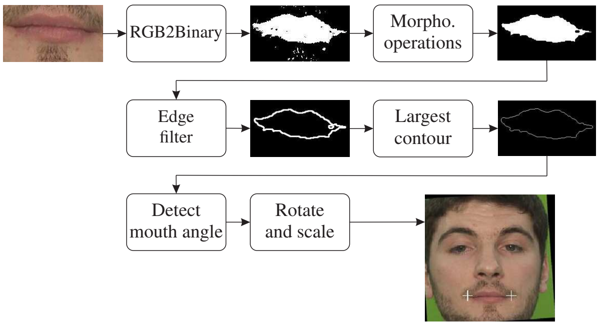

Figure 2: *ROI post processing.*

Given a new video frame, each KLT tracker is examined whether it can track at least two corner points from the previous frame. The tracked points of the successful tracker are then used to estimate an affine transformation, which is used to transform the mouth BB of the previous frame into the new mouth BB of the current frame. In the combination phase, if the two trackers succeeded to track two corner points from the previous frame, the two resulting mouth BBs are combined into a larger BB containing both of them. If only one tracker could track two corner points, its transformed BB is considered as the combined one. If both trackers failed to track two corner points, the algorithm returns to the initialization phase and the Viola-Jones algorithm is deployed to find new face and moth regions in the next frame. The algorithm returns also to the initialization phase to update the interest points with a refresh rate of 40 frames. The joint face and ROI detection algorithm has been implemented using Matlab's Computer Vision System Toolbox™.

3.2. ROI Post-processing

After finding the mouth region, the next stage is to post-process the detected ROI images by rotating them so that the speaker's mouth within the ROI images lies on a horizontal line. The angle by which the ROI should be rotated can be determined using the approach shown in Figure 2. The first step in this approach is to create a binary image from the mouth image by setting all values above a threshold to one and all other values to zero. The threshold is empirically determined based on the ratio of the red value to the green and blue values in the RGB image of the speaker's mouth. After applying two morphological operations, i.e, dilation and erosion, an edge filter is applied to find contours in the binary image. Finally, the contour that contains the largest area is considered as the contour of the mouth.

The angle between the x-axis and the line connecting the rightmost and leftmost points of the mouth contour is the angle by which the ROI should be rotated. Moreover, the length of this line is used to determine the factor by which the ROI should be normalized to a predefined size (here 67 × 67 pixels). The rotation and scaling processes described above significantly simplifies the training of automatic lip reading models.

The difference between the acoustic frame rate (100 frame/s) and the visual frame rate (30 frame/s) is compensated by repeating visual frames according to the digital differential analyzer (DDA) algorithm [24].

[2] The author would also like to thank Herman Steeneken, who gave the permission to redistribute the RSG-10 noise signals.

3.3. Visual Feature Extraction

The final stage of the visual front end is to extract the actual visual features from the post-processed ROI images. The discrete cosine transform (DCT) and the principal component analysis (PCA) are used to get 32-dimensional spectral feature vectors. The dimension has been empirically determined. In addition to the PCA visual feature vectors, the first and second derivatives are also used. In order to enhance the discrimination capability of the visual feature vectors while decreasing the correlation between their components, linear discriminant analysis LDA [25] is applied. The feature-space maximum likelihood linear regression (fMLLR) [26] approach is finally used to estimate speaker-adapted features. The dimension of the visual feature vectors after applying the linear transformations is 40. In order to include context information, 11 consecutive fMLLR frames (5 previous, current, and 5 future) are concatenated to form the 440-dimensional visual feature vectors. The raw DCT features are available for download in Matlab format, while the linearly-transformed features are available in Kaldi format.

4. Audio-visual Fusion

Two audio-visual fusion schemes have been used. The first one is the direct integration (DI), also called feature fusion and early integration. The second fusion model is the separate integration (SI), also called decision fusion or late integration.

In the DI fusion scheme, fusion is applied at the feature level. The audio and video feature vectors are concatenated to form new audio-visual feature vectors. The new feature vectors are then used to train an AV-ASR system. In this type of fusion, both the audio and video features are assumed to have the same utility and reliability at each time frame.

In the SI fusion models, the audio and video streams are first recognized separately. The recognition results are then combined using a voting scheme. Unlike DI, SI can control the contribution of the audio and video streams to the overall recognition results according to their reliabilities using stream confidence measures. DI, however, does not account for the natural temporal dependencies between the audio and video streams.

5. Baseline Experiments and Results

5.1. Models

The speaker-independent *acoustic* deep neural network/hidden Markov model (DNN/HMM) hybrid models have been trained as follows: Following the convention, mono-phone Gaussian mixture model/hidden Markov model (GMM/HMM) models have been firstly trained, where 3-states mono-phone HMMs have been used. The model parameters have been initialized with global parameters (a flat start procedure). Next, tri-phone GMM/HMM models with 1953 tied tri-phone states have been trained. The features used for training the initial mono- and tri-phone models are the 13-dimensional mel-frequency cepstral coefficients (MFCC) cascaded with their first and second derivatives. The MFCC features have been extracted from the clean signals of the training set. The tri-phone GMM/HMM models have then been used to estimate a linear discriminant analysis (LDA) transformation matrix, which is applied to the raw acoustic feature vectors to create new LDA-based acoustic features. The LDA features have been used to train new tri-phone GMM/HMM models. The final tri-phone GMM/HMM models are speaker-adaptive-trained (SAT) models, which have been trained using 40-dimensional fMLLR feature vectors.

Table 1: *Viseme error rates (VERs) and phone error rates (PERs) of the automatic lip reading models.*

Model	VER		PER	
	Dev Set	Test set	Dev Set	Test set
Spk-dep, GMM, Mono [7]	66.0	66.0	–	–
Spk-ind, GMM, Mono	63.6	63.7	77.0	77.1
GMM, Tri	61.5	61.4	76.3	75.9
GMM, Tri (LDA)	60.7	61.5	74.8	75.1
GMM, Tri (SAT)	57.9	58.2	72.2	72.4
DNN, Tri (CE)	54.0	53.5	68.3	67.4
DNN, Tri (sMBR)	52.5	51.7	66.7	65.4

The frame-state alignments required for training the tri-phone models have been obtained by applying the forced alignment algorithm to the previously trained model. The last frame-state alignments generated using the SAT tri-phone GMM/HMM models are used as the DNN's training labels.

The DNN has 6 hidden layers, each of which consists of 1024 neurons with sigmoid activation functions. The number of units in the output softmax layer is 1953, which is the number of the tied tri-phone states [27]. Each 11 consecutive (5 previous, current, and 5 future) 40-dimensional acoustic fMLLR frames have been cascaded to form the 440 input units of the DNN.

The DNN weights have been tuned using the stochastic gradient descent and back propagation algorithms so that the cross entropy (CE) objective function is minimized. Finally, the DNN weights have been re-tuned to minimize the sequential minimum bias risk (sMBR) objective function [28].

The speaker-independent *visual* DNN/HMM hybrid models have been trained similarly to the acoustic ones. However, the frame-state alignments used for training the initial visual mono-phone GMM/HMMs and the final visual DNN/HMMs have been obtained from the corresponding acoustic models.

A simple bi-phone language model has been used for decoding. The phone-level transcription of the training set has been used for training the bi-phone language model.

Acoustic feature extraction as well as training of acoustic, visual, and language models have been conducted using the Kaldi speech recognition toolkit.

5.2. Results

5.2.1. Automatic Lip Reading

Table 1 shows the evolution of the viseme error rate (VER) and the phone error rate (PER) from their highest values obtained using the mono-phone GMM/HMM visual models to their lowest ones achieved using the tri-phone sequentially-trained DNN/HMM models. In order to estimate the VERs, viseme-level transcriptions of the reference and recognized phones have been obtained using the phoneme-to-viseme mapping table in [7]. As can be seen, there are only slight differences between the development and test set results. Table 1 also shows that the average VER of the speaker-independent baseline proposed here is significantly better than the corresponding speaker-dependent VER reported in [7]. This can be attributed to the adopted visual front-end, the weighted-finite-state-transducers (WFST)-based decoder of Kaldi, and the bi-phone language model.

5.2.2. Audio-only ASR

Tables 2 and 3 show the PERs of the audio-only ASR system obtained using the development and test sets in all noise conditions. The results shown in Tables 2 and 3 are obtained from a clean-train-noisy-test setup. This explains the massive increase in the PERs in all noisy environments. However, the noisy train-

Table 2: *PERs of the development set in clean and noisy test conditions obtained using an audio-only ASR system.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	23.5	32.3	40.0	50.0	62.0	73.6	83.8	52.2
	White	23.5	46.6	58.7	70.8	81.2	89.0	93.4	66.2
	Babble	23.5	47.1	59.7	71.5	82.2	90.0	93.9	66.8
	L. Room	23.5	52.8	64.8	75.3	83.4	89.0	91.7	68.6
	Street	23.5	56.1	67.3	77.7	85.8	91.4	94.4	70.9
	Cafe	23.5	65.5	71.4	76.9	82.2	88.7	94.7	71.8
Average		23.5	50.1	60.3	70.4	79.5	87.0	92.0	66.1

Table 3: *PERs of the test set in clean and noisy test conditions obtained using an audio-only ASR system.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	21.6	28.7	35.2	44.6	56.2	69.4	80.4	48.0
	White	21.6	40.9	53.1	66.7	77.7	86.7	91.4	62.6
	Babble	21.6	42.2	54.6	68.3	79.2	88.8	92.7	63.9
	L. Room	21.6	47.0	60.3	72.6	81.5	87.7	91.1	66.0
	Street	21.6	50.7	63.2	74.8	84.5	90.8	93.6	68.5
	Cafe	21.6	62.6	69.9	75.2	81.4	89.0	94.5	70.6
Average		21.6	45.4	56.0	67.0	76.8	85.4	90.6	63.3

Table 4: *PER of the development set obtained using an audio-visual ASR system with a direct integration fusion scheme.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	22.1	29.1	35.3	44.3	56.0	68.5	79.9	47.9
	White	22.1	40.9	52.1	64.8	76.4	85.9	91.3	61.9
	Babble	22.1	40.7	52.9	65.0	76.8	85.8	91.1	62.1
	L. Room	22.1	45.4	57.0	68.5	78.6	85.6	89.9	63.9
	Street	22.1	48.2	59.5	71.3	81.3	88.7	93.1	66.3
	Cafe	22.1	55.2	62.2	69.9	78.2	86.2	93.2	66.7
Average		22.1	43.3	53.2	64.0	74.6	83.4	89.8	61.5

Table 5: *PER of the test set obtained using an audio-visual ASR system with a direct integration fusion scheme.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	20.4	25.1	29.6	37.5	48.8	63.3	76.2	43.0
	White	20.4	34.4	45.9	58.8	72.2	83.1	89.9	57.8
	Babble	20.4	35.3	46.4	60.1	73.0	84.1	89.3	58.4
	L. Room	20.4	39.6	51.3	64.1	75.4	84.4	89.2	60.6
	Street	20.4	42.0	54.6	67.3	79.2	87.5	92.3	63.3
	Cafe	20.4	51.3	59.6	67.3	76.0	85.4	92.8	64.7
Average		20.4	37.9	47.9	59.2	70.8	81.3	88.3	58.0

ing signals of all test conditions are available for download to allow for conducting matched- and multi-condition-training experiments.

5.2.3. AV-ASR Results

Tables 4 and 5 show the development and test set PERs of an AV-ASR system trained using a DI fusion scheme. First, the fMLLR-based audio and video features have been cascaded. The resulting 80-dimensional audio-visual feature vectors have then been used to train a new DNN/HMM AV-ASR model.

SI has been implemented by combining the recognition results of the audio-only and video-only ASR systems using ROVER (Recognition Output Voting Error Reduction) [29]. The combined results are shown in Tables 6 and 7. All hyper parameters required for ROVER have been tuned for each noise condition using the corresponding development sets.

Figure 3 compares the test set PERs of the audio-only, video-only, and audio-visual ASR. The PERs shown in Figure 3 are the average PERs over the test conditions. As can be seen, the performance of the DI-AV-ASR system is better than the best performance of the audio-only and video-only ASR systems in high SNRs. In low SNRs, however, DI-AV-ASR is outperformed by the video-only ASR system.

It can also be seen from Figure 3 that the performance of the SI-AV-ASR system follows the best performance of the audio-only and video-only ASR systems when the difference between their performance is very large. It only becomes slightly better than both audio-only and video-only ASR systems when their performances become comparable, e.g, at 10 dB.

Table 6: *PER of the development set obtained using an audio-visual ASR system with a separate integration fusion scheme.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	24.2	33.0	40.5	50.4	62.1	66.9	66.9	49.1
	White	24.2	47.0	58.9	66.8	66.9	66.9	66.8	56.8
	Babble	24.2	47.5	59.7	66.8	66.9	66.9	66.9	57.0
	L. Room	24.2	53.1	64.1	66.8	66.9	66.8	66.8	58.4
	Street	24.2	56.3	65.6	66.9	67.0	67.0	66.9	59.1
	Cafe	24.2	65.3	66.8	66.8	66.9	67.0	66.9	60.6
Average		24.2	50.4	59.3	64.1	66.1	66.9	66.9	56.8

Table 7: *PER of the test set obtained using an audio-visual ASR system with a separate integration fusion scheme.*

	SNR [dB]	Clean	20	15	10	5	0	-5	Average
Noise Type	Car	22.5	29.4	35.8	45.1	57.7	65.5	65.6	45.9
	White	22.5	41.4	53.3	65.5	65.6	65.7	65.6	54.2
	Babble	22.5	42.8	54.9	65.5	65.7	65.8	65.6	54.7
	L. Room	22.5	47.4	60.8	65.5	65.6	65.6	65.6	56.1
	Street	22.5	51.0	63.1	65.6	65.7	65.8	65.7	57.1
	Cafe	22.5	62.5	65.4	65.5	65.6	65.7	65.6	59.0
Average		22.5	45.8	55.6	62.1	64.3	65.7	65.6	54.5

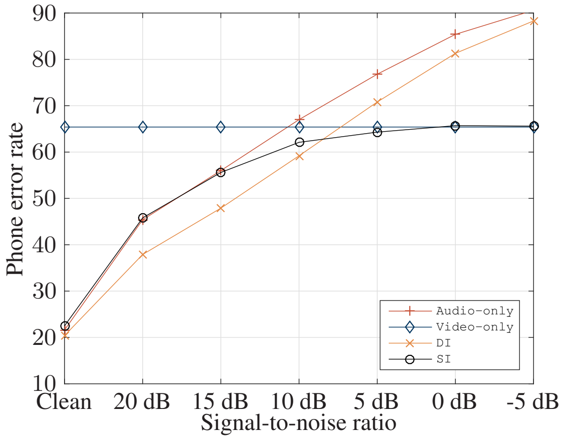

Figure 3: *Test set phone error rate of the audio-only ASR, video-only ASR, DI-based AV-ASR, and SI-based AV-ASR systems averaged over the test conditions.*

6. Conclusions

In this paper, the noisy TCD-TIMIT audio-visual speech corpus has been introduced. The clean audio-visual speech material of this database has been obtained from the recently published TCD-TIMIT corpus. The audio files of TCD-TIMIT have been distorted by a selection of 6 noise types over a range of signal-to-noise ratios. A front-end for visual feature extraction has been presented. Baseline audio-only, video-only, and audio-visual ASR experiments have been conducted using Kaldi and baseline results have been reported. The noisy audio files, the visual features, and the Kaldi baseline scripts are available for download from [30]. The database and the baseline scripts can attract more computer vision and ASR researchers to the AV-ASR field. ASR researchers can simply use the available visual features in this corpus to conduct AV-ASR experiments. Computer vision researchers can also use the available Kaldi ASR scripts to test new visual features for AV-ASR.

7. References

[1] R. Lippmann, “Accurate consonant perception without mid-frequency speech energy,” *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 66–69, 1996.

[2] J. Licklider and I. Pollack, “Effects of differentiation, integration, and infinite peak clipping upon the intelligibility of speech,” *The Journal of the Acoustical Society of America*, vol. 20, pp. 42–51, 1948.

[3] A. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, “Dynamic Bayesian networks for audio-visual speech recognition,” *EURASIP Journal on Applied Signal Processing*, vol. 11, pp. 1274–1288, 2002.

[4] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, “Recent advances in the automatic recognition of audiovisual speech,” *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1306–1326, 2003.

[5] A. H. Abdelaziz, S. Zeiler, and D. Kolossa, “Learning dynamic stream weights for coupled-HMM-based audio-visual speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 5, pp. 863–876, 2015.

[6] A. K. Katsaggelos, S. Bahaadini, and R. Molina, “Audiovisual fusion: Challenges and new approaches,” *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1635–1653, 2015.

[7] N. Harte and E. Gillen, “TCD-TIMIT: An audio-visual corpus of continuous speech,” *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 603–615, 2015.

[8] M. Cooke, J. Barker, S. Cunningham, and X. Shao, “An audio-visual corpus for speech perception and automatic speech recognition,” *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, Nov. 2006.

[9] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy, “CUAVE: A new audio-visual database for multimodal human-computer interface research,” in *Proc. ICASSP*, 2002, pp. II–2017–II–2020.

[10] I. Matthews, G. Potamianos, C. Neti, and J. Luettin, “A comparison of model and transform-based visual features for audio-visual LVCSR,” in *Proc. ICME*, 2001, pp. 825–828.

[11] T. N. Sainath, B. Ramabhadran, and M. Picheny, “An exploration of large vocabulary tools for small vocabulary phonetic recognition,” in *Proc. ASRU*, 2009, pp. 359–364.

[12] D. Povey *et al.*, “The Kaldi speech recognition toolkit,” in *Proc. ASRU*, 2011.

[13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, “TIMIT acoustic-phonetic continuous speech corpus,” *Linguistic data consortium, Philadelphia*, vol. 33, 1993.

[14] H. Steeneken and F. Geurtsen, “Description of the RSG-10 noise database.” TNO Institute for Perception, Tech. Rep., 1988.

[15] A. Varga and H. J. M. Steeneken, “Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems,” *Speech Communication*, vol. 12, no. 3, pp. 247–251, 1993.

[16] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, “The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines,” in *Proc. ICASSP*, 2013, pp. 126–130.

[17] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third ‘CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” in *Proc. ASRU*, 2015, pp. 504–511.

[18] H.-G. Hirsch, “F a N T - filtering and noise adding tool,” International Computer Science Institute, Niederrhein University of Applied Science, Tech. Rep., 2005.

[19] H.-G. Hirsch and D. Pearce, “The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” in *Proc. ISCA ITRW ASR2000*, 2000, pp. 29–32.

[20] N. Parihar, J. Picone, D. Pearce, and H.-G. Hirsch, “Performance analysis of the Aurora large vocabulary baseline system,” in *Proc. the 12th European Signal Processing Conference*, 2004, pp. 553–556.

[21] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. CVPR*, 2001, pp. I–511.

[22] J. Shi and C. Tomasi, “Good features to track,” in *Proc. CVPR*, 1994, pp. 593–600.

[23] B. D. Lucas, T. Kanade *et al.*, “An iterative image registration technique with an application to stereo vision.” in *Proc. IJCAI*, vol. 81, no. 1, 1981, pp. 674–679.

[24] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[25] D. Kolossa, S. Zeiler, R. Saeidi, and R. Astudillo, “Noise-adaptive LDA: A new approach for speech recognition under observation uncertainty,” *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1018–1021, 2013.

[26] M. J. Gales, “Maximum likelihood linear transformations for HMM-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[27] S. J. Young, J. J. Odell, and P. C. Woodland, “Tree-based state tying for high accuracy acoustic modelling,” in *Proc. workshop on Human Language Technology*, 1994, pp. 307–312.

[28] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks.” in *Proc. Interspeech*, 2013, pp. 2345–2349.

[29] J. G. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER),” in *Proc. ASRU*, 1997, pp. 347–354.

[30] A. H. Abdelaziz, “NTCD-TIMIT,” May 2017. [Online]. Available: https://doi.org/10.5281/zenodo.260228