

# **Robust Techniques for Generating Talking Faces from Speech**

by

Sefik Emre Eskimez

Submitted in Partial Fulfillment of the  
Requirements for the Degree  
Doctor of Philosophy

Supervised by Professors Wendi Heinzelman and Zhiyao Duan

Department of Electrical and Computer Engineering

Arts, Sciences and Engineering

Edmund A. Hajim School of Engineering and Applied Sciences

University of Rochester

Rochester, New York

2019

*To my beloved family...*



# Table of Contents

<b>Biographical Sketch</b>	<b>vii</b>
<b>Acknowledgments</b>	<b>ix</b>
<b>Abstract</b>	<b>x</b>
<b>Contributors and Funding Sources</b>	<b>xiii</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.1.1 Benefits of Visual Component in Speech Comprehension . . . . .	3
1.1.2 Augmented/Virtual Reality Agents . . . . .	3
1.1.3 Movie Dubbing . . . . .	4
1.2 Limitations of Existing Systems . . . . .	5
1.3 Related Areas . . . . .	6
1.3.1 Speech Enhancement (SE) . . . . .	6
1.3.2 Speech Animation (SA) . . . . .	7
1.3.3 Automatic Speech Emotion Recognition (ASER) . . . . .	7
1.4 Contributions . . . . .	8
1.5 Thesis Structure . . . . .	11
<b>Chapter 2: Speech Enhancement</b>	<b>12</b>
2.1 Front-End Speech Enhancement for Commercial Speaker Verification Systems . . . . .	12
2.1.1 Introduction . . . . .	12
2.1.2 Related Work . . . . .	13
2.1.3 Network Architecture . . . . .	17

---

2.1.4	Experiments . . . . .	21
2.1.5	Conclusions . . . . .	38
2.2	Adversarial Training for Speech Super-Resolution . . . . .	38
2.2.1	Introduction . . . . .	38
2.2.2	Related Work . . . . .	40
2.2.3	Proposed SSR System . . . . .	44
2.2.4	Experiments . . . . .	49
2.2.5	Results . . . . .	54
2.2.6	Noise Analysis . . . . .	58
2.2.7	Computational Complexity . . . . .	63
2.2.8	Conclusions . . . . .	64
<b>Chapter 3: Generating Talking Faces From Speech: Shape-Based Methods</b>		<b>65</b>
3.1	Generating Talking Face Landmarks From Speech . . . . .	65
3.1.1	Introduction . . . . .	65
3.1.2	Related Work . . . . .	66
3.1.3	Proposed Method . . . . .	68
3.1.4	Experiments . . . . .	72
3.1.5	Conclusions . . . . .	74
3.2	Noise-resilient Training Method For Face Landmarks Generation From Speech . . . . .	75
3.2.1	Introduction . . . . .	75
3.2.2	Related Work . . . . .	77
3.2.3	Method . . . . .	80
3.2.4	Experiments . . . . .	87
3.2.5	Conclusion . . . . .	98
<b>Chapter 4: Generating Talking Faces From Speech: Image-Based Methods</b>		<b>99</b>
4.1	End-to-End Talking Faces from Speech . . . . .	99
4.1.1	Introduction . . . . .	99
4.1.2	Related Work . . . . .	100

---

4.1.3	Method . . . . .	101
4.1.4	Experiments . . . . .	108
4.1.5	Conclusion . . . . .	108
<b>Chapter 5:</b>	<b>Automatic Speech Emotion Recognition (ASER)</b>	<b>109</b>
5.1	Introduction . . . . .	109
5.2	Amazon Mechanical Turk Study . . . . .	109
5.2.1	Introduction . . . . .	109
5.2.2	Related Work . . . . .	111
5.2.3	LDC Dataset . . . . .	112
5.2.4	Automated Emotion Classification System . . . . .	112
5.2.5	Amazon’s Mechanical Turk Setup . . . . .	114
5.2.6	Evaluation . . . . .	115
5.2.7	Discussions . . . . .	119
5.2.8	Conclusions . . . . .	120
5.3	WISE: Web-based Interactive Speech Emotion Classification . . . . .	121
5.3.1	Introduction . . . . .	121
5.3.2	Related Work . . . . .	122
5.3.3	Web-based Interaction . . . . .	123
5.3.4	Automated Emotion Classification System . . . . .	124
5.3.5	Evaluation . . . . .	126
5.3.6	Conclusions . . . . .	130
5.4	Unsupervised Learning Approach to Feature Analysis for Automatic Speech Emotion Recognition . . . . .	130
5.4.1	Introduction . . . . .	130
5.4.2	Related Work . . . . .	131
5.4.3	Method . . . . .	133
5.4.4	Experiments . . . . .	140
5.4.5	Conclusions . . . . .	143

---

<b>Chapter 6: Generating Emotionally Expressive Talking Faces</b>	<b>144</b>
6.1 System Overview . . . . .	144
6.2 Speech Emotion Recognition Module . . . . .	144
6.3 Emotion Discriminator . . . . .	145
6.3.1 Experiments . . . . .	146
6.3.2 Conclusion . . . . .	147
<b>Chapter 7: Conclusions and Future Work</b>	<b>149</b>
7.1 Conclusions . . . . .	149
7.2 Future Work . . . . .	150

## Biographical Sketch

The author was born in Diyarbakir, Turkey. He attended Sabanci University and graduated with a Bachelor of Science degree in Mechatronics Engineering in 2011. He began graduate studies in the Department of Mechatronics Engineering at Sabanci University in 2011 and received a Master of Science degree in 2013. He began graduate studies in the Department of Electrical and Computer Engineering at the University of Rochester in 2014 and received a Master of Science degree in 2015. He pursued his research in speech processing and deep learning under the direction of Wendi Heinzelman and Zhiyao Duan.

The following publications were a result of work conducted during doctoral study:

**S. E. Eskimez**, R. K. Maddox, C. Xu, Z. Duan, “*Noise-resilient training method for face landmarks generation from speech,*” under Review.

**S. E. Eskimez**, K. Koishida and Z. Duan, “*Adversarial training for speech super-resolution,*” IEEE Journal of Selected Topics in Signal Processing, vol. 13, no. 2, pp. 347-358, May 2019.

**S. E. Eskimez** and K. Koishida, “*Speech super resolution generative adversarial network,*” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, United Kingdom, pp. 3717-3721, 2019.

L. Chen, **S. E. Eskimez**, Z. Li, Z. Duan, C. Xu, R. K. Maddox, “*Toward a visual assistive listening device: Real-time synthesis of a virtual talking face from acoustic speech using deep neural*

*networks*,” The Journal of the Acoustical Society of America, vol. 143, no. 3, pp. 1813-1813, 2018.

**S. E. Eskimez**, R. K. Maddox, C. Xu, Z. Duan, “*Generating talking face landmarks from speech*,” In International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA), pp. 372-381, Springer, Cham, 2018.

**S. E. Eskimez**, Z. Duan and W. Heinzelman, “*Unsupervised learning approach to feature analysis for automatic speech emotion recognition*,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 5099-5103, Apr 2018.

**S. E. Eskimez**, P. Soufleris, Z. Duan and W. Heinzelman, “*Front-end speech enhancement for commercial speaker verification systems*,” Speech Communication, vol. 99, pp. 101-113, 2018.

**S. E. Eskimez**, M. Sturge-Apple, Z. Duan and W. Heinzelman, “*WISE: Web-based interactive speech emotion classification*,” in Proceedings of International Joint Conference on Artificial Intelligence (IJCAI) - 4th Workshop on Sentiment Analysis where AI meets Psychology (SAAIP), pp. 2-7, New York City, USA. July 2016. <http://system.wise.audio/>

**S. E. Eskimez**, K. Imade, N. Yang, M. Sturge-Apple, Z. Duan and W. Heinzelman, “*Emotion Classification: How Does an Automated System Compare to Naive Human Coders?*,” in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 2274-2278, Shanghai, China, Mar 2016.

## Acknowledgments

I would like to thank my advisors, Professor Wendi Heinzelman and Professor Zhiyao Duan, for the opportunity to work with them, and their mentorship over the past few years. During my time in the Wireless Communication and Networking Group (WCNG) and Audio Information Research Group (AIR), both spent endless amounts of time proofreading my papers and providing excellent suggestions and insights for my work.

I would like to thank Professors Wendi Heinzelman and Zhiyao Duan from the Department of Electrical and Computer Engineering, Professor Ross K. Maddox from the Department of Biomedical Engineering and the Department of Neuroscience, and Professor Chenliang Xu from the Department of Computer Science, for acting as members of my committee.

I would like to thank the CEO and founder of Voice Biometrics Group (VBG) Pete Soufleris for his continued support and providing me with resources through my Doctoral studies.

I would like to thank Kazuhito Koishida for his support and providing me with resources during my internship in Microsoft Research.

I would like to thank all of my lab-mates and colleagues in the WCNG and AIR labs for their support.

I would also like to thank all of my colleagues at the University of Rochester for their support, as well as VBG Corporation for funding a part of my Doctoral studies.

Finally, I would like to thank my family and friends for their love and support.

# Abstract

Speech is a fundamental modality in human-to-human communication. It carries complex messages that written languages cannot convey effectively, such as emotion and intonation, which can change the meaning of the message. Due to its importance in human communication, speech processing has attracted much attention of researchers to establish human-to-machine communication. Personal assistants, such as Alexa, Cortana, and Siri that can be interfaced using speech, are now mature enough to be part of our daily lives. With the deep learning revolution, speech processing has advanced significantly in the fields of automatic speech recognition, speech synthesis, speech style transfer, speaker identification/verification and speech emotion recognition.

Although speech contains rich information about the message that is being transmitted and the state of the speaker, it does not contain all the information for speech communication. Facial cues play an important role in establishing a connection between a speaker and a listener. It has been shown that estimating emotions from speech is a hard task for untrained humans; therefore most people rely on a speaker's facial expressions to discern the speaker's affective state, which is important for comprehending the message that the speaker is trying to convey. Another benefit of the availability of facial cues during speech communication is that seeing the lips of the speaker improves speech comprehension, especially in environments where background noise is present. This can be observed mostly in cocktail-party scenarios, where people tend to communicate better when they are facing each other but may have trouble communicating when talking over the phone.

This thesis describes my work in the fields of speech enhancement (SE), speech animation (SA), and automatic speech emotion recognition (ASER). For SE, I have proposed long short-term memory (LSTM) based and convolutional neural network (CNN) based architectures to compensate for



the non-stationary noise in utterances. My proposed models have been evaluated in terms of speech quality and speech intelligibility. These models have been used as pre-processing modules to a commercial automatic speaker verification system, and it has been shown that they provide a performance boost in terms of equal-error rate (EER).

I have proposed a speech super-resolution (SSR) system that employs a generative adversarial network (GAN). The generator network is fully convolutional with 1D kernels, enabling real-time inference on edge devices. The objective and subjective studies showed the proposed network outperforms the DNN baselines.

For speech animation (SA), I have proposed an LSTM network to predict face landmarks from first- and second-order temporal differences of the log-mel spectrogram. I have conducted objective and subjective evaluations and verified that the generated landmarks are on-par with the ground-truth ones. Generated landmarks can be used by the existing systems to fit texture or 2D and 3D models to obtain realistic talking faces to increase speech comprehension. I extended this work to include noise-resilient training. The new architecture accepts the raw waveforms and processes them through 1D convolutional layers that output the PCA coefficients of the 3D face landmarks. The objective and subjective results showed that the proposed network achieves better performance compared to my previous work and a DNN-based baseline. In another work, I have proposed an end-to-end image-based talking face generation system that works with arbitrarily long speech inputs and utilizes attention mechanisms.

For automatic speech emotion recognition (ASER), I have compared human and machine performance in large-scale experiments and concluded that machines could discern emotions from speech better than untrained humans. I have also proposed a web-based automatic speech emotion classification framework, where the user can upload their files and can analyze the affective content of the utterances. The framework adapts to the user's choices over time since the user corrects the wrong labels. This allows for large-scale emotional analysis in a semi-automatic framework. I have proposed a transfer learning framework where I train autoencoders using 100 hours of neutral speech to boost the ASER performance. I have systematically analyzed four different autoencoders, namely denoising autoencoder, variational autoencoder, adversarial autoencoder and adversarial variational Bayes.

This method is beneficial in scenarios where there are not enough annotated data to train deep neural networks (DNNs).

Pulling all of this work together provides a framework for generating a realistic talking face from noisy and emotional speech that has the capability of expressing emotions. This framework would be beneficial for applications in telecommunications, human-machine interaction/interface, augmented/virtual reality, telepresence, video games, dubbing, and animated movies.

## **Contributors and Funding Sources**

This work was supported by a dissertation committee consisting of Professors Wendi Heinzelman (co-advisor) and Zhiyao Duan (co-advisor) from the Department of Electrical and Computer Engineering, Professor Ross K. Maddox from the Department of Biomedical Engineering and the Department of Neuroscience, and Professor Chenliang Xu (chair) from the Department of Computer Science.

The speech enhancement (SE) research was supported by the Voice Biometrics Group (VBG) and the National Science Foundation grant No. 1617107.

The speech super-resolution (SSR) research was completed during an internship at Microsoft Research.

The speech animation (SA) research was supported by a University of Rochester Pilot Award Program in AR/VR and the National Science Foundation grant No. 1741472.

All work in this dissertation was completed independently by the author.

## List of Figures

2.1	Proposed BLSTM network architecture for speech enhancement. Input vector $\mathbf{v}_t$ is the concatenation of the normalized log-amplitude spectra of $2c + 1$ frames centered around the $t$ -th time frame, where $c$ is the short-term context window parameter. Hidden layer outputs are denoted as $\mathbf{h}_t^n$ , where $n$ is the layer index. $\hat{\mathbf{m}}_t^s$ is the predicted mask for the speech. . . . .	18
2.2	Proposed convolutional encoder-decoder (CED) network architecture for speech enhancement. The numbers of filters in the convolution and deconvolution layers are 128, 256, 512, 1024, 512, 256, and 128, respectively. The input is an L-frame magnitude spectrogram, where the output are estimated L-frame mask of speech and noise spectrograms. The red arrows represent the skip connections. . . . .	19
2.3	An example of speech enhancement results. Magnitude spectrograms of the noisy speech signal corrupted by motorcycle noise at 0 dB, the ground-truth clean speech, and enhanced speech of six speech enhancement methods, namely <i>SS</i> , <i>Log-MMSE</i> , <i>RNN</i> , <i>R-CED</i> , <i>BLSTM</i> and <i>CED</i> . . . . .	23
2.4	PESQ comparison between the proposed methods ( <i>BLSTM</i> and <i>CED</i> ) and baseline traditional methods ( <i>SS</i> [1] and <i>Log-MMSE</i> [2]) and baseline DNN-based methods ( <i>RNN</i> and <i>R-CED</i> [3]) for different noise types and SNRs. . . . .	24
2.5	STOI comparison between the proposed methods ( <i>BLSTM</i> and <i>CED</i> ) and baseline traditional methods ( <i>SS</i> [1] and <i>Log-MMSE</i> [2]) and baseline DNN based methods ( <i>RNN</i> and <i>R-CED</i> [3]) for different noise types and SNRs. . . . .	25
2.6	PESQ and STOI comparisons averaged over all noise types. . . . .	26

2.7	PESQ and STOI comparisons averaged over all noise types for different numbers of hidden units (64, 128, 256, 512 and 1024) per layer in the BLSTM network. . . . .	27
2.8	PESQ and STOI comparisons averaged over all noise types for different numbers of filters ( $M = 8, 16, 32, 64$ and 128) in the first convolutional layer in the CED network. The numbers of filters in the other convolutional and deconvolutional layers are powers-of-two times of $M$ , following the same symmetric pattern shown in Fig. 2.2.	28
2.9	PESQ and STOI comparisons averaged over all noise types for different numbers of hidden layers (1, 2 and 3) in the BLSTM network. . . . .	29
2.10	PESQ and STOI comparisons averaged over all noise types for different numbers of layers (3, 5 and 7) in the CED network. . . . .	29
2.11	PESQ and STOI comparisons averaged over all noise types for mean-squared error (MSE) and binary cross-entropy (BCE) loss functions in BLSTM and CED networks.	30
2.12	PESQ and STOI comparisons averaged over all noise types between log-mel spectrogram (MEL) and log-linear spectrogram (LIN) inputs for BLSTM and CED networks.	31
2.13	Histogram of SNR estimation of <i>VBG RANDNUM</i> files. . . . .	34
2.14	Histogram of SNR estimation of <i>RedDots</i> files. . . . .	34
2.15	<i>VBG RANDNUM</i> EER results. Note that the y-axis starts from 5.0%. . . . .	36
2.16	EER results for <i>RedDots</i> Dataset. Note that the y-axis starts from 13.0%. . . . .	36
2.17	<i>RedDots</i> dataset EER results for different noise types and SNRs. . . . .	37

- 2.18 Overview of the proposed SSR system during test time. The Log-Power Spectra (LPS)  $X^{NB}$  and the phase spectrogram  $X_P$  are calculated from the input narrow-band waveform  $x$  through Short-Time Fourier Transform (STFT).  $X^{NB}$  is fed to the speech super-resolution generative adversarial network (SSR-GAN) to obtain the estimated high-frequency range LPS  $\hat{X}^{WB}$ , which is then concatenated with the original narrowband LPS. The phase of the high-frequency range is artificially produced by flipping and repeating the narrowband phase  $X_P$  and adding a negative sign. For fractional super-resolution factors, the last repeat is truncated to match the frequency range. Finally, the estimated wideband LPS and artificial phase are used to reconstruct the time-domain signal  $\hat{y}$  by Inverse STFT (ISTFT) and overlap-add. . . . . 43
- 2.19 The proposed network architectures for the generator (middle) and the discriminator (right). Each rectangular block is a convolutional layer with structures color coded and detailed on the left subfigure. The generator is an autoencoder with concatenating skip connections, predicting the high-frequency range of the input narrowband magnitude spectrogram. It is then concatenated with the original low-frequency range to generate the full wideband magnitude spectrogram. The input to the discriminator is the full wideband spectrogram of either a real sample or a generated sample. We do not use batch normalization in the discriminator. Notations: *BN* - batch normalization layer, *FC* - fully connected layer, *LReLU* - LeakyReLU activation, and *PShuffle* - pixel shuffle or sub-pixel layer, *LPS* - log-power spectrogram. . . . . 45
- 2.20 The adversarial training procedure for the proposed method. The generator contains the concatenation of narrowband LPS and high-frequency LPS. . . . . 49
- 2.21 Spectrogram examples for 2x and 4x, shown in (a) and (b), respectively. The samples are randomly selected from the WSJ0 corpus (unseen speakers). The first row in each Figure shows the ground truth high-frequency range spectrograms. The second and third rows show the generated high-frequency range spectrograms of the proposed network trained with only the LSD loss (second rows) and with both LSD and GAN losses (third rows). . . . . 52

2.22	Objective evaluation results are presented for changing the number of layers in the encoder and decoder of the generator network. The results for 2x and 4x scales are shown in (a) and (b), respectively. The four sets of bars show <i>LSD HF</i> , <i>LSD Full</i> , <i>SegSNR</i> , and <i>PESQ</i> values, respectively. . . . .	55
2.23	Objective evaluation results are presented for changing the number of filters of the generator network. The results for 2x and 4x scales are shown in (a) and (b), respectively. <i>Half</i> and <i>Double</i> means that the number of filters shown in Table 2.1 has been halved and doubled, respectively. The four sets of bars show <i>LSD HF</i> , <i>LSD Full</i> , <i>SegSNR</i> , and <i>PESQ</i> values, respectively. . . . .	57
2.24	Objective evaluation results are presented for different loss weight parameters ( $\lambda$ ) and for <i>SSR-LSD</i> for comparison. The results for 2x and 4x scales are shown in (a) and (b), respectively. The four sets of bars show <i>LSD HF</i> , <i>LSD Full</i> , <i>SegSNR</i> , and <i>PESQ</i> values, respectively. . . . .	58
2.25	The subjective evaluation results (MUSHRA test) for 2x and 4x scales are shown in (a) and (b), respectively. The error bars show the 95% confidence intervals. . . . .	60
3.1	Examples of extracted face landmarks from the training talking face videos. Certain landmarks are connected to make the shape of the face easier to recognize. The first row shows unprocessed landmarks of five unique talkers. The second row shows their landmarks after outer-eye-corner alignment. The third row shows their landmarks after alignment and the removal of identity information. . . . .	68
3.2	The LSTM network architecture for generating landmarks of a talking face from the first and second order temporal differences of the log-mel spectrogram. $h_t^l$ are the hidden layers, where $t$ is the time step and $l$ is the hidden layer index. $y_t$ are the output face landmarks for the time step $t$ . . . . .	71
3.3	Pair-wise comparison between ground-truth landmarks (black solid lines) and generated landmarks (red dotted lines) on unseen talkers and sentences. The second image shows a failure case for “oh” sound. . . . .	72

3.4	Subjective evaluation results. The mean accuracy score and its standard deviation are averaged over all subjects. The mean confidence scores and their standard deviations are averaged over all subjects and videos. . . . .	74
3.5	Data preparation steps for face landmarks illustrated on six different speakers, where each column corresponds to a speaker. We draw lines between certain landmarks to form face shapes. The first, second, and third rows show raw face landmarks extracted from video images, landmarks after Procrustes alignment, and landmarks after identity removal, respectively. . . . .	81
3.6	The network architecture for (a) <i>ID_CNN</i> network and (b) <i>ID_CNN_TC</i> network. <i>ID_CNN_TC</i> is identical to <i>ID_CNN</i> , except that it accepts the previous frame's ASM weights as a condition to enforce temporal constraint. Raw waveform is fed to four convolutional layers, followed by a fully connected (FC) layer. . . . .	85
3.7	The noise-resilient training scheme. The networks on the left and right sides are the same, and their weights are shared. The clean and noisy speech goes through the left and the right networks, respectively, to reconstruct their face landmarks. A mean-squared error (MSE) constraint is applied to the latent representations to incorporate the supervised speech enhancement idea at the feature level. . . . .	86
3.8	System overview. A talking face is generated every 40 ms (frame hop size) from 320 ms (frame length) of audio. $t$ represents the time. . . . .	87
3.9	Single speaker objective evaluation results for the <i>BL1</i> [4], <i>BL2</i> [5], <i>ID_CNN</i> and <i>ID_CNN_TC</i> methods. We calculate the root-mean-squared error (RMSE) between generated and ground-truth 2D mouth landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation. . . . .	90
3.10	Multi speaker objective evaluation results for the <i>BL1</i> [4], <i>BL2</i> [5], <i>ID_CNN</i> and <i>ID_CNN_TC</i> methods. We calculate the root-mean-squared error (RMSE) between generated and ground-truth 2D full face landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation. . . . .	90



- 3.11 The example output showing the pronunciation of the word “ash”. The speech sample was taken from STEVI corpus. The first row shows the result generated by *ID\_CNN*. The second row shows the comparison of the result generated by *ID\_CNN* and the ground-truth (dotted red line). The third and fourth rows show the result generated by *ID\_CNN\_TC* and comparison with the ground-truth (dotted red line). Columns show every three frames. . . . . 92
- 3.12 Comparison of *ID\_CNN* configurations with different number of convolution layers. The number of filters for Layers 1 to 4 is shown in Table 3.2. The number of Layers 5 and 6 is both 512. We compare the root-mean-squared error (RMSE) between generated and ground-truth landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation. . . . . 93
- 3.13 The comparison of *ID\_CNN* configurations that has a different number of filters in convolutional layers is shown. The number of filters in the first layer is displayed, which are 16, 32, 64 and 128. After the first layer, the filters are doubled with each following convolutional layer. We compare the root-mean-squared error (RMSE) between generated and ground-truth landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation. . . . . 93
- 3.14 The comparison of results for different sizes of the input speech is shown for *ID\_CNN* network. The number of frames is displayed, which are 5, 7, 9. Each frame spans 40 ms speech. We predict the middle frame and use previous and past frames as context information. We compare the root-mean-squared error (RMSE) between generated and ground-truth landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation. . . . . 94
- 3.15 The results for the subjective test of speech-mouth match. The bars show the average score for the baseline method, proposed method (*ID\_CNN*) and ground-truth face landmarks. Error bars show the standard deviation. . . . . 96

- 4.1 The proposed end-to-end talking face generation system overview. The input reference image and raw speech waveform are processed by the image encoder and speech encoder, respectively. For each frame, a normally distributed random noise vector is generated and fed to the noise encoder that contains an LSTM layer. The image, speech, and noise features are sent to the generator. During training, we use both the adversarial loss and the reconstruction loss. The frame discriminator improves the image quality, where the pair discriminator improves the mouth movements and speech synchronization. . . . . 102
- 4.2 An example of the mouth region mask is shown. The mouth is located using the mean point of mouth face landmarks. A 2D Gaussian is placed at the mean point to isolate the mouth. The first row shows the original frame, the second row shows the mouth region mask, and the third row shows the masked mouth. . . . . 102
- 4.3 The architecture of the speech encoder. The network accepts an arbitrarily long speech waveform and processes it frame by frame through five convolutional layers. The resulting embedding is a time-series corresponding to these frames. Past and future frames as context information are also fed to the network as input when the network processes each frame. For the beginning and ending frames of the waveform, we concatenate zeros as the context information. Every fifth frame is kept to form the final speech features. . . . . 103
- 4.4 The architecture of the generator. The noise, image, and speech features are concatenated at each frame and fed into a fully connected layer, and the output is reshaped. Then, the results are concatenated with skip connections from the image decoder and fed into a convolutional layer. This is repeated for all layers except for the last convolutional layer. . . . . 105
- 4.5 The architecture of the pair discriminator is shown. The input is the masked mouth frames, condition speech, and condition image. The image and speech encoders are identical to our main speech and image encoders, but the parameters are updated only during discriminator training. Each frame is classified as real or fake. . . . . 106

4.6	Example generation results shown along with the baseline results and the ground-truth. a) shows an utterance of “POINT” and b) shows an utterance of “RUSSIANS”.	107
5.1	Questions shown to Turkers. . . . .	115
5.2	Flow chart showing the operation of WISE. . . . .	124
5.3	WISE user interface screenshot. . . . .	125
5.4	The results of emotion category for Scenarios I-III. . . . .	127
5.5	The results of arousal category for Scenarios I-III. . . . .	127
5.6	The results of valence category for Scenarios I-III. . . . .	128
5.7	Proposed ASER system overview. The dashed red windows represent the sliding window with 50% overlap. From each window, emotion class probabilities ( $p_1$ , $p_2$ , $p_3$ , $p_4$ and $p_5$ ) are predicted and the average of these vectors is calculated over all windows is calculated for each utterance. The emotion that has the highest probability is predicted as the emotion of the utterance. . . . .	132
5.8	DAE network architecture: reconstructing the clean spectrogram from noisy input . .	135
5.9	VAE network architecture: variational inference on auto-encoder by constraining the latent representation to follow a normal distribution . . . . .	136
5.10	AAE network architecture: variational inference on auto-encoder by constraining the latent representation through adversarial training . . . . .	138
5.11	AVB network architecture: unifying VAE and generative adversarial networks (GANs)	139
5.12	The unweighted accuracy rating (UAR) and F1-score results for the baseline systems and the proposed systems. F1-score is calculated for each class, and their unweighted mean is presented. . . . .	141
6.1	The proposed end-to-end emotionally expressive talking face generation system overview. There are two modifications compared to the base system described in Chapter 4. The first modification is to add a speech emotion recognition module that classifies the input speech’s emotion. The second modification is to use another discriminator that checks if the video contains the given emotion. . . . .	145

---

6.2	The automatic speech emotion recognition module is shown. The module accepts speech features as input to two LSTM layers followed by a fully connected layer that outputs a probability for each emotion class. . . . .	146
6.3	The architecture of the emotion discriminator. The video frames are fed into the image encoder, and the resulting embeddings are concatenated with emotion embeddings and are fed into a BLSTM layer. The output is fed into an FC layer that classifies the frames as real or fake. . . . .	147
6.4	This example shows different emotions using the same condition image. . . . .	148

## List of Tables

2.1	Detailed parameters of the proposed network architecture. The number of channels and hidden units, filter sizes, strides, activations and output shapes are shown for each layer in the generator and discriminator networks. $K$ and $N$ are the narrowband and the high-frequency range LPS dimensions along the frequency axis, respectively. $K$ is 129 and 65 for 2x and 4x super-resolution scales, respectively. $N$ is 141 and 199 for 2x and 4x super-resolution scales, respectively. . . . .	47
2.2	The objective evaluation results for 2x and 4x SSR experiments. The bolded values show the best results. Our method ( <i>SSR-GAN</i> ) outperforms the baselines for all metrics. <i>LSD HF</i> shows the LSD value calculated only for the high-frequency range, where <i>LSD Full</i> shows the LSD value calculated for the whole spectrogram. . . . .	54
2.3	Objective evaluation results for noise analysis. . . . .	59
2.4	The intelligibility test results. The mean and standard deviation (std) of word error rate (WER) is shown for the 2x and 4x scale experiments using <i>SSR-GAN</i> . . . . .	62
2.5	Computational complexity in terms of floating point operations per second (FLOPS), FLOPS per generating 1 second of speech and number of parameters for the baselines (BL1 and BL2) and the proposed <i>SSR-GAN</i> method. . . . .	63
3.1	Objective evaluation results for different system configurations. The models are named according to the amount of delay and contextual information. For example, “D40-C5” describes a model trained with 40 ms delay and 5 frames of context. The lower value means better results, where the ideal result is zero. . . . .	73

3.2	Detailed parameters of the proposed network architecture. The number of filters and hidden units, filter sizes, strides, activations, and output shapes are shown for each layer. <i>ID_CNN_TC</i> is identical to <i>ID_CNN</i> ; further, it accepts condition input and concatenates it with the output of the fully connected (FC) layer that is shown in the last two rows of the table. This concatenated tensor is fed to another FC layer that outputs the final ASM weights. . . . .	84
3.3	Objective results for the <i>ID_CNN</i> method and noise-resilient (NR) version of it ( <i>ID_CNN_NR</i> ) for clean and noisy speech input. We present results for Babble, Factory, SSN, Motorcycle and Cafeteria noises at 5 and 10 dB SNRs, none of which were not included in the training noise corpus. Best results in each noise setting are bolded. . . . .	95
4.1	Detailed parameters of the proposed network architecture. The number of filters and hidden units, filter sizes, strides, activation functions, and output shapes are shown for each layer. . . . .	104
5.1	Number of samples classified by Turkers. . . . .	117
5.2	Accuracy values (%) for six emotions. . . . .	117
5.3	Confusion matrix for the automatic classification system (GT = ground truth). . . . .	117
5.4	Confusion matrix for the Turkers (GT = ground truth). . . . .	118
5.5	Accuracy values (%) for APN and PNN. . . . .	118
5.6	The architecture of the encoder, decoder, discriminator and emotion classifier networks. AEs share the encoder and decoder structures, except AVB where we modify the encoder to accept external noise input similar to AVB discriminator architecture. <i>Conv2D</i> is a 2-d convolution layer, where <i>Conv2DT</i> is a transposed 2-d convolution (or deconvolution) layer. <i>Concat</i> is the concatenation layer. <i>F. No</i> is the number of filters, where <i>F. Size</i> is the filter size. . . . .	134

## Chapter-1

# Introduction

Imagine you are in a noisy environment, where you need to talk to someone over the phone. It would be tough to understand what the other person is saying for both parties. In another scenario, you are in an environment where the Internet connection is limited; hence it is not possible to use video chat, but you are trying to talk to a person who is hard of hearing. Alternatively, you are an elder person talking with a machine, such as a personal assistant (Cortana, Siri, Alexa), and want to see a familiar human face that makes you comfortable.

An obvious way to improve speech comprehension in such scenarios is to use text messaging instead, at the cost of discarding the emotional state, tone, accent, articulation, co-articulation and facial expressions of the speaker. Discarding such information may impair successful communication between the parties. Speech communication involves a visual counterpart in addition to the acoustic signal. The facial cues carry essential information including emotions, where these emotions may influence the context of the speech.

For the hearing-impaired population, the presence of a visual signal is even more essential. The missing information due to the lack of an acoustic signal can be retrieved from the facial cues. Having an automated system that can generate a talking face from speech will enable the hearing-impaired population to access much of the available speech content online.

In this thesis, to address these issues, I work on the problem of generating a realistic talking face from speech that accounts for emotions to improve or even establish speech comprehension.

## 1.1 Motivation

Speech is the most common mode of communication in our society. The main purpose of speech is human-to-human and human-to-machine communication, where the speaker transmits a message to the listener [6]. Denes and Pinson describe the speech communication process as the “speech chain” [7]. The speech chain starts with the formulation of a message, usually represented by text, in the speaker’s brain. This is followed by converting the message into phonemes and prosody, i.e., the language code. The language code drives the neuro-muscular control mechanism that moves the speech articulators, i.e., the lips, teeth, tongue, jaw, eyes, eyebrows, and facial expressions, in sync with the desired speech. The next step in the speech chain is the vocal tract system that creates the acoustic waveform of the speech. The steps described in the speech chain up to this point are called *speech production*, and occur on the speaker’s side. Speech production works similarly for machines: the text is converted to phonemes and prosody. Then the intonation is determined according to the phonemes and prosody. This is followed by generating speech parameters and synthesizing the speech according to these parameters. However, automatic systems usually do not generate a visual signal.

The processes described after this point in the speech chain occur on the listener’s side and are called *speech comprehension*. The speech is converted from an acoustic waveform to a spectral representation in the inner ear. The next step is feature extraction by the neural transduction, which produces features that can be processed by the brain. At the same time, the human visual system processes the visual signal by extracting features from visual cues as described in the previous paragraph. Both audio and visual features are merged and synced. Then the features are converted into the language code, i.e., the phonemes, words, and sentences. The last step in the speech chain is the high-level understanding of the message. If the listener is a machine, the process is similar: the waveform is converted to the feature representation and piped into a statistical model, and the output of the model is post-processed to obtain the high-level understanding of the message.

Speech processing has attracted the attention of researchers in order to develop automatic understanding (pattern recognition), efficient transmission, synthesis of the speech, speaker identification/verification and aids for the hearing impaired population. These problems have been extensively



studied using traditional machine learning algorithms, and most of them have been commercialized and are being used in our daily life. With the *deep learning* revolution, significant improvements have been obtained for these problems over traditional methods.

### 1.1.1 Benefits of Visual Component in Speech Comprehension

Speech comprehension, as described in the previous section, is the process of understanding the message that the speaker is trying to transmit. During speech production, the visual components, which are the movement of lips, teeth, tongue, jaw, eyes, eyebrows, and skin, are generated to reflect the affective state of the speaker that directly influences the meaning of the speech. In light of this information, it is necessary to obtain the information from the visual signal counterpart of the speech in order to achieve true speech comprehension.

Most commercial systems do not consider the visual counterpart of speech in their framework. Most of the research is focused on the automatic speech recognition (ASR) problem, which can be described as the conversion of speech to text. The text modality is important for human-machine communication. However, it is not sufficient to describe what the speaker is trying to convey. The intonation and sentiment pieces of information are discarded in such systems.

It is shown that when there is a complementary visual signal present during speech communication, the speech comprehension is significantly improved [8, 9, 10, 11]. This is more evident when the background is noisy. For example, in a cocktail party scenario, two people facing each other can communicate effectively regardless of the background noise since the visual signal is present; however, it would be challenging to talk over a phone in the same situation.

### 1.1.2 Augmented/Virtual Reality Agents

Text-to-speech synthesis systems can produce realistic speech, thanks to recent advances in deep learning. Some of these systems consider generating emotional speech [12, 13] to establish natural communication with the user. However, just using acoustic speech to interact with a computer might still feel unnatural for most people, especially for the elderly population.

Augmented reality (AR) and virtual reality (VR) technologies are still in their infancy. However, with rapidly increasing computational power, these technologies will be accessible to a large amount of the population soon. Interacting with the participant who is in the VR from the normal reality (NR) can break the immersion for the participant, and speech comprehension can be difficult due to the environment noise. For example, VR is already being used for training purposes in domains such as police enforcement, military, sports, flight schools, industrial machine operation, and driving. The communication between the student, who is in VR, and instructor, who is in NR, must be seamless. If these systems can generate a synthetic talking face of the instructor speaking in NR seamlessly, the student can better be immersed in the training scenario and have increased speech comprehension. Also, there are multi-person VR applications, where the avatars of the users interact. These avatars must have the capability to mimic the visual component of the speech when the users talk.

### 1.1.3 Movie Dubbing

When translating from one language to another, the number of words and syllables changes per sentence. Furthermore, the intonation varies for different languages. The facial expressions might also differ in time, for example in one language the facial expression for disgust may appear at the beginning of the sentence, while in another language it may appear at the end for the same sentence. These differences make it difficult to dub a movie while keeping the lips and facial expressions synced to the new speech.

A framework that can generate a talking face from speech, conditioned on the input face image/video, is useful in this scenario. A realistic, emotionally expressive talking face can be generated using the target actor's identity (provided as an image or video clip), and the dubbed speech. The original face can be replaced with the generated one, which will solve the lip movement and facial expression mismatch problem for dubbed movies.

## 1.2 Limitations of Existing Systems

There are a few existing works in the literature that generate emotionally expressive talking faces from speech [14, 15, 16, 17, 18]. However, it is important to note that it is hard to compare some of these older techniques with modern deep learning based methods. Cao et al. [14] proposed a method to generate an emotionally expressive face by first estimating the emotion from speech using a support vector machine (SVM) based system and then generating a face using a graph that represents the visual motion of a phoneme. Deng et al. [15] proposed an eigenspace based expression model, where personality vectors can be applied to other targets. The main limitation of these two approaches is that they highly depend on a reliable estimation of phonemes. In our method, we plan to use the raw audio, and let the network automatically learn the mapping between phonemes and the facial motions.

Pham et al. [16] proposed a method to predict 3D blend shape parameters from speech. They trained their system using an emotional audio-visual database and can generate emotionally expressive faces. They improved their work by directly predicting the 3D blend shape parameters from raw speech [17]. They tested their system using four speakers from the same dataset that is not seen during training. Large-scale, unseen speakers from other datasets must also be evaluated to draw conclusions. Karras et al. [18] proposed an end-to-end network to generate 3D vertex points from speech. Their proposed method does not rely on the categorical representation of emotion. Instead, they let the network learn the emotional expression from the raw data itself. Their network is designed for a single speaker, but is still able to generalize well to unseen speakers. However, a large-scale evaluation must be conducted with major categories of emotions to draw conclusions about the generalization capacity of this network. In our proposed method, we design our network to be speaker-independent, and we train with a wide variety of speakers. Also, different from these two approaches, we want to produce the image/video of the speaker to increase the realism, which will expand the application areas.

There are image-based methods that can generate talking faces from speech using a single frame of the target identity [4, 19, 20, 21, 22, 23]. Some of these works first predict sparse intermediate points of the face [4, 19] followed by mapping these sparse points to images, and others predict the images directly from speech features [20, 21, 22], where [23] directly predicts the images from raw

waveforms. Facial expression generation and speech comprehension are not the focus of these works. Only [22, 23] consider natural movements, such as head movements and eye blinking.

## 1.3 Related Areas

In this section, we present the related research areas for generating an emotionally expressive talking face from speech. These problems have been studied extensively by the research community.

### 1.3.1 Speech Enhancement (SE)

In real-world applications, the speech signal sampled from the user usually is not clean. The signal is frequently corrupted by background noise, channel compression, and reverberation. These corruptions can be in different signal-to-noise ratio (SNR) levels and can include a wide variety of non-stationary noises. A speech signal corrupted in this way is hard to understand for humans, especially for the hearing impaired population, as well as machines. Almost all automatic speech processing systems such as automatic speech recognition (ASR), automatic speech emotion recognition (ASER), and automatic speaker recognition/verification (ASID/ASV) have degraded performance when they receive corrupted speech as input.

Speech enhancement is the problem of removing/reducing the corruption in a speech signal, and it has received much attention and been well-studied in the speech processing research community. Early works in this area focused on compensating static-noise such as white, pink, blue, purple, and brown noises using statistical models. However, most of the noise encountered in the real world is non-stationary. To eliminate non-stationary noises, the temporal structure of the speech signal must be considered. Most recently, deep learning based methods have significantly improved the SE performance over classical methods.

The biggest challenge in SE is to deal with unseen noises in the wild, for unseen speakers, and in unknown SNRs. Although deep learning based methods provide decent results, there is still room for improvement.

### 1.3.2 Speech Animation (SA)

As computational resources rapidly increase, complex graphical animations for video games, animated movies, and augmented/virtual reality applications have emerged. The animation of a character often requires animating the mouth/face, as those characters tend to speak. The best quality mouth/face animations are obtained using motion capture systems, where an actor/actress wears gear designed specifically for this purpose. However, these systems are costly and require manual labor.

Given a speech signal, automatically generating talking face graphics in accordance with speech is called speech animation. The challenges include accent, language, culture, gender, age and emotion variabilities of the speakers. Usually, these systems must be speaker-independent to be useful.

The works in SA include generation of 2D and 3D meshes, active appearance model parameters, and directly generating the image/video. 2D/3D models are more appropriate for computer graphics, while directly generating images/videos is more realistic and is suitable for boosting speech comprehension.

### 1.3.3 Automatic Speech Emotion Recognition (ASER)

As mentioned in the previous sections, emotion is essential for communication. The affective state of the person must be determined to interact with that person appropriately. Being able to estimate emotions from the speech is beneficial for the field of behavioral psychology for analyzing the development of children, social interactions, and couple relations. It is also useful for businesses that interact with customers via telephones to try to estimate the customer/employee satisfaction, such as call centers. One of the requirements for realistic AI systems is being able to estimate the user's emotions, as emotions can profoundly influence the context of communication.

Automatically determining the affective state or mood of a person from only an acoustic speech signal is called ASER, and it has been studied over two decades. ASER has not matured as much as the other speech processing systems to be able to apply it to our daily lives. However, as the research on ASER has been making rapid progress recently, the techniques may be mature in the near future for commercial applications.

The lack of well annotated data is the biggest issue in ASER. Usually, researchers rely on psychologists to annotate the speech samples. Emotions are subjective, and determining emotions objectively requires experts. Besides, annotators do not agree with each other all the time. Even if they agree, the annotating process is manual and takes a lot of time. Therefore, no single benchmark dataset is accepted by the community. There are a few datasets, and researchers usually benchmark their results on some/all of these datasets to show the performance of their approach.

## 1.4 Contributions

This thesis aims to address the issues in the areas of speech enhancement (SE), speech animation (SA), and automatic speech emotion recognition (ASER), as steps to developing a system that can produce an emotionally expressive talking face video solely from speech. The specific contributions of this thesis include the following:

- I propose two deep neural network (DNN) architectures for SE, and I compare the performance of the proposed networks with existing work. I evaluate the resulting SE networks using the objective measures of perceptual evaluation of speech quality (PESQ) and short-time objective intelligibility (STOI). Second, I analyze the performance of automatic speech verification (ASV) systems when SE methods are used as front-end processing to remove the non-stationary background noise. I compare the resulting equal error rate (EER) using my DNN based SE approaches, as well as existing SE approaches, with real customer data and the freely available *RedDots* dataset. The results show that my DNN based SE approaches provide benefits for speaker verification performance. This work is described in [24] and in Chapter 2, Section 2.1.
- I develop a speech super-resolution system that leverages generative adversarial networks to obtain state-of-the-art results. The network generates the missing high-frequency spectrograms and contains only convolutional layers with 1D kernels. The results are evaluated against recent DNN based methods in terms of log-spectral distance (LSD), segmental SNR, and PESQ metrics in addition to perceptual listening tests. Both objective and subjective tests show that the

proposed systems reach better performance compared to the baselines. The proposed system is light-weight and can run in edge devices. This work is described in [25] and [26] and in Chapter 2, Section 2.2.

- I propose a system that can generate landmark points of a talking face from an acoustic speech signal in real time. The system uses a long short-term memory (LSTM) network and is trained on frontal videos of 27 different speakers with automatically extracted face landmarks. After training, it can produce talking face landmarks from the acoustic speech of unseen speakers and utterances. I evaluate this system using the mean-squared error (MSE) loss of landmarks of lips between predicted and ground-truth landmarks as well as their first- and second-order temporal differences. I further evaluate this system by conducting subjective tests, where the evaluators try to distinguish the real and fake videos of talking face landmarks. Both tests show promising results. This work is described in [5] and in Chapter 3, Section 3.1.
- I improve the face landmark generation work by introducing noise-resilient training that can increase the robustness against unseen non-stationary noises. In this new system, the network generates 3D face landmarks instead of 2D landmarks. The network architecture is changed by replacing the LSTM layers with 1D convolutional layers that directly operates on the raw waveforms. I also propose another variant of the network, that accepts the previous frame's face landmarks in order to generate temporally smooth sequences. I evaluate the proposed networks using the root-mean-squared error (RMSE) loss of face landmarks between predicted and ground-truth landmarks as well as their first- and second-order temporal differences using unseen data. I further evaluate this system by conducting subjective tests, where the evaluators try to distinguish the real and fake videos of talking face landmarks. Both objective and subjective results show that the proposed networks achieve better results compared to the baseline systems. This work is described in Chapter 3, Section 3.2.
- I propose an end-to-end image-based talking face generation method that works with arbitrary length speech input. The system utilizes LSTM-Convolutional layers in addition to the attention mechanism to produce realistic talking faces. The speech encoder takes the raw waveform and

calculates the short-term features. The final speech features are obtained by adding the context information from past and future short-term features and skipping every fifth frame to match the video frame per seconds. The generated results are further improved by employing generative adversarial networks. This work is described in Chapter 4.

- I present the first large-scale comparison in a speech-based emotion classification task between 138 Amazon Mechanical Turk workers (Turkers) and an SVM based automatic computer system. I show that the computer system outperforms the naive Turkers in almost all cases. I conclude that the computer system can increase the classification accuracy by rejecting to classify utterances for which it is not confident, while the Turkers did not show a significantly higher classification accuracy on their confident utterances versus unconfident ones. This work is described in [27] and in Chapter 5, Section 5.2.
- I introduce a web-based interactive speech emotion classification system, WISE. WISE has a web-based interface that allows users to upload speech data and automatically classify the emotions within this speech using pre-trained models. The system adapts to the user's choices over time to increase prediction accuracy. This is the first system of its kind to my knowledge. I evaluate WISE by simulating the user interactions with the system using the LDC dataset, which has known, ground-truth labels. I evaluate the benefit of the user feedback enabled by WISE in situations where manually classifying emotions in a large dataset is costly, yet trained models alone will not be able to accurately classify the data. This work is described in [28] and in Chapter 5, Section 5.3.
- I systematically investigate four kinds of unsupervised feature learning methods for improving the performance of a speaker-independent ASER system. I specifically explore the denoising autoencoder (DAE), variational autoencoder (VAE), adversarial autoencoder (AAE) and adversarial variational Bayes (AVB) in the context of ASER. I show that all methods improve the performance regarding unweighted accuracy rating (UAR) and F1-score over methods that use hand-crafted features or that do not perform feature learning on external datasets. I also show that VAE, AAE and AVB methods, which control the distribution of the latent representation,



outperform DAE that does not control for such a distribution. This work is described in [29] and in Chapter 5, Section 5.4.

## 1.5 Thesis Structure

This thesis has the following structure: I describe my speech enhancement (SE) and speech super-resolution (SSR) work in Chapter 2. Next, I describe my speech animation (SA) research and describe my talking face landmarks from the speech network in Chapter 3 and image-based speech animation work in 4. I present my contributions regarding automatic speech emotion recognition (ASER) in Chapter 5. Finally, I summarize the conclusions of my thesis in Chapter 7 and discuss future work.

## Chapter-2

# Speech Enhancement

## 2.1 Front-End Speech Enhancement for Commercial Speaker Verification Systems

### 2.1.1 Introduction

Automatic speaker verification (ASV) systems are vital for security applications in areas such as financial services, law enforcement, and government security. A security breach occurs when an ASV system makes a false authorization for an imposter, which may lead to economic, personal or national security consequences. Noise, reverberation and channel distortion are factors that significantly impair the performance of ASV systems and make the ASV system particularly vulnerable to imposter attacks or missed verification.

Therefore, speech enhancement (SE), which aims to reduce noise in the speech signal, is an important pre-processing module in commercial ASV systems. These systems in general use traditional SE techniques [1, 30, 2], which have been shown to be effective against stationary noise. However, as most noise types encountered in real-world applications are non-stationary, traditional SE techniques do not perform well in these cases.

Deep neural networks (DNNs) have been successfully applied to SE systems to model non-stationary noise [31, 32, 33, 34, 35, 36, 37, 38, 39, 40]. However, these techniques have typically been tested in laboratory settings using an artificially created speech corpus (e.g., TIMIT Acoustic-Phonetic Continuous Speech Corpus sentences [41]), where the utterances are spoken in a very different way,

i.e., not natural, compared to real-world speech utterances. To be able to assess the feasibility of SE systems in commercial applications, these methods need to be evaluated with real-world utterances in addition to artificial tests.

In this work, we propose two DNN-based speech enhancement approaches. We apply them as a front-end noise removal module for a state-of-the-art speaker verification system and test the combined systems. In addition to evaluating the proposed systems using utterances collected and mixed in laboratory settings, we also use utterances that are collected by a commercial ASV system from real customers, as well as the freely available Reddats dataset to evaluate the proposed systems. We show that both systems yield superior results compared to traditional methods, in terms of both objective speech quality and intelligibility measures and speaker verification performance.

## 2.1.2 Related Work

In this section, we review existing work on speech enhancement and its application to speaker verification systems.

### Speech Enhancement: Classical Methods

Early notable works on speech enhancement modeled the noise statistically, typically using the first 4-5 frames of the noisy speech signal, assuming those are noise only. These methods, such as spectral subtraction (SS) [1], minimum mean square error spectral amplitude estimator (MMSE) [30] and minimum mean square error log-spectral amplitude estimator (Log-MMSE) [2], produce disturbing musical artifacts, which are portions of spectral power appearing in random frequency regions, in the predicted signal. Since these techniques use the first frames to model the noise, they are not effective against time-varying noises.

### Speech Enhancement: Deep Learning Methods

In recent years, DNN based methods have been shown to significantly outperform classical methods. Various deep models have been proposed, but generally they can be classified into two categories:

*regression-based* and *masking-based*.

Regression-based methods attempt to learn the mapping from noisy speech to clean speech directly. Lu et al. [31] trained a deep auto-encoder (DAE) on Mel-scale power spectral patches of clean speech and used this to denoise noisy speech. Later, they extended the model by training the DAE with noisy-clean speech pairs [32] and by introducing ensemble models [33].

Similarly, Xu et al. [34, 35, 36, 37] used restricted Boltzmann machines (RBMs) to learn a mapping function from the log power spectra of noisy speech to those of clean speech. They extended this work by adding a statistical estimate of the noise from the first several frames to the network's input to achieve noise-aware training [35]. In [36], they further extended this work by introducing global variance equalization to tackle the over-smoothing issue that causes the removal of speech segments in the predicted speech, which leads to muffled speech.

Park et al. [3] proposed a redundant convolutional encoder-decoder (R-CED) network, which is a fully convolutional network, for mapping the noisy STFT magnitude to clean STFT magnitude. They applied 1D convolution along the frequency axis. The input to the network is eight frames including the current and the past seven frames, where the output is the current frame's clean version.

Masking-based methods, on the other hand, attempt to predict the time-frequency (T-F) filters or masks that are later applied to noisy speech spectra to recover the corresponding clean speech spectra. Methods in this category have shown significant improvements over regression-based methods [42, 43, 44, 38, 45, 46]. Various types of masks have been proposed. Binary masks such as the ideal binary mask (IBM) [47, 42] and the target binary mask (TBM) [48] set the mask value at a T-F unit to 1 when speech dominates and to 0 when noise dominates. Soft masks such as the ideal ratio mask (IRM) [43] and the Wiener-like mask [47, 44, 43] use a real value between 0 and 1 to reflect the relative dominance of speech in each T-F unit. An extension to soft masks is a complex soft mask such as the complex ideal ratio mask [46]. This mask uses complex numbers and is applied to the complex spectra of the noisy speech. Wang et al. [45] investigated some of the above-mentioned masks in a supervised simultaneous speech separation system.

Different types of DNNs have been proposed to predict these masks from noisy speech for SE. Chen et al. [39] trained a feed-forward DNN to predict the IRM from 64-band cochleagrams of the

noisy speech. The network was trained with 10,000 different types of noise to increase the robustness against unseen noises. Weninger et al. [40] used a long short-term memory (LSTM) network to predict *phase sensitive masks*, and tested the use of this speech enhancement system on the performance of a speech recognition system. Huang et al. [38] proposed a recurrent neural network to jointly output the clean speech, noise, and the IRM. The training objective function considers both the interference reduction and mask prediction.

### Automatic Speaker Verification

The Gaussian mixture model (GMM) - universal background model (UBM) ASV system described in [49, 50] utilizes GMMs to model the acoustic space, which is parametrized by the selected acoustic features. A GMM with a typically large number of mixtures is trained using a large pool of speakers. This model is usually called the UBM.

Dehak et al. [51] proposed a *total variability space* that represents the speaker and channel variability. The speaker's supervector can be represented in the total variability space by the following equation,

$$s = m + Tw \text{ ,} \quad (2.1)$$

where  $s$  is the speaker's supervector,  $m$  is the mean supervector of the GMM-UBM,  $T$  is the total variability matrix, and  $w$  is the latent variable where the maximum a posteriori (MAP) point estimate of  $w$  given the utterance is  $\phi$ , which is called the identity vector (i-vector). For the process of training the  $T$  matrix and extracting the i-vectors, please see [52] and [51], respectively.

Probabilistic linear discriminant analysis (PLDA) assumes that the i-vector  $\phi$  can be represented by the following equation,

$$\phi_{l,r} = \mu + Fh_l + Gv_{l,r} + \epsilon_{l,r} \text{ ,} \quad (2.2)$$

where  $F$  and  $G$  matrices represent the speaker and channel subspace,  $l$  and  $r$  represent the speaker and session indexes,  $h_l$  and  $v_{l,r}$  represent the speaker- and session-specific vectors,  $\mu$  represents the mean i-vector and  $\epsilon_{l,r} \sim \mathcal{N}(0, \Sigma)$  represents the residual noise. The PLDA parameters  $\theta_{PLDA} = \{\mu, F, G, \Sigma\}$

can be estimated by expectation maximization (EM). The probabilistic form of Eq. 2.2 is as follows

$$p(\phi_{l,r}) = \mathcal{N}(\phi_{l,r} | \mu, FF^T + GG^T + \Sigma) . \quad (2.3)$$

For detailed information on how to estimate PLDA parameters with EM, how to calculate multi-session PLDA scoring and how to apply length normalization, please refer to [53, 54], [55] and [56], respectively.

### SE Application to ASV Systems

Godin et al. [57] evaluated speaker identification (SID) methods and SID performance improvements using the early (classical) speech enhancement techniques described in Section 2.1.2 [1, 30, 2] to see if SE is useful in real noisy telephone conversations. They compared the equal error rate (EER) values between artificially generated noisy speech (i.e., adding noise to clean speech) and natural noisy speech, and found that they do not correlate well.

In recent years, deep-learning based speech enhancement methods have also been integrated into ASV and SID systems. Zhao et al. [58, 59] proposed a robust SID system under noisy and reverberant conditions where the IBM prediction was adopted for speech enhancement. They integrated SE and SID systems at the feature level.

Kolbæk et al. [60] proposed an LSTM-based SE front end for a text-dependent i-vector-based ASV system. This SE network includes two LSTM layers and a fully connected layer. For each audio frame (32 ms window with 16 ms hop size), the input to their network is a concatenation of the magnitude spectra of the current frame and its previous 15 and future 15 frames, totaling 31 frames of data. The output of the network is the T-F mask of the current frame. They trained and evaluated their system using six types of non-stationary noises and compared their results with classical SE methods. They showed that their method outperforms classical methods in an SNR range from -5 dB to 10 dB.

Although this was a good evaluation of SE systems as a denoising front-end, this evaluation had two limitations. First, all noise types that were used for evaluation were used for training; a more thorough analysis using unseen noise types would be required. Second, all noisy speech utterances

were created by artificially mixing clean speech utterances with noise; while this made it possible to create noisy speech with different SNRs, an additional evaluation with natural noisy speech would be required to show the SE front end’s performance with commercial ASV systems in real-world scenarios.

To the best of our knowledge, there has not been a thorough analysis of state-of-the-art speech enhancement approaches working with commercial text-independent ASV systems in real-world scenarios. As in [60], we treat the ASV system as a black box: we enhance the noisy speech and then feed it to the ASV system for speaker verification. In our experiments, we use natural noisy speech samples that were collected by Voice Biometrics Group (VBG) and utterances from the *RedDots* dataset and evaluate the verification error rate on these enhanced utterances. In addition, we conduct artificial tests by mixing additional noise to natural noisy speech utterances with different SNRs and evaluate the verification error rate.

### 2.1.3 Network Architecture

In this section, we propose two neural network architectures for speech enhancement as the front end of our ASV systems.

#### **Bidirectional LSTM Network**

The first architecture we propose has a total of five layers including the input layer, as shown in Figure 2.1. Each hidden BLSTM layer contains 1024 units. The input layer receives a sequence of  $L$  vectors, each of which corresponds to one time frame of the input noisy speech. Specifically, each vector is the concatenation of the log-amplitude spectrogram of the  $2c + 1$  neighboring frames centered around the current frame, where  $c$  is the short-term context window parameter. Including the neighboring frames provides subsequent layers with contextual information. The input then goes through three Bidirectional LSTM (BLSTM) [61] layers that model the temporal dependencies of the signal. The output layer consists of a BLSTM layer to reconstruct the speech mask.

We use dropout layers with a 0.2 dropout rate between the BLSTM hidden layers and add  $l_2$

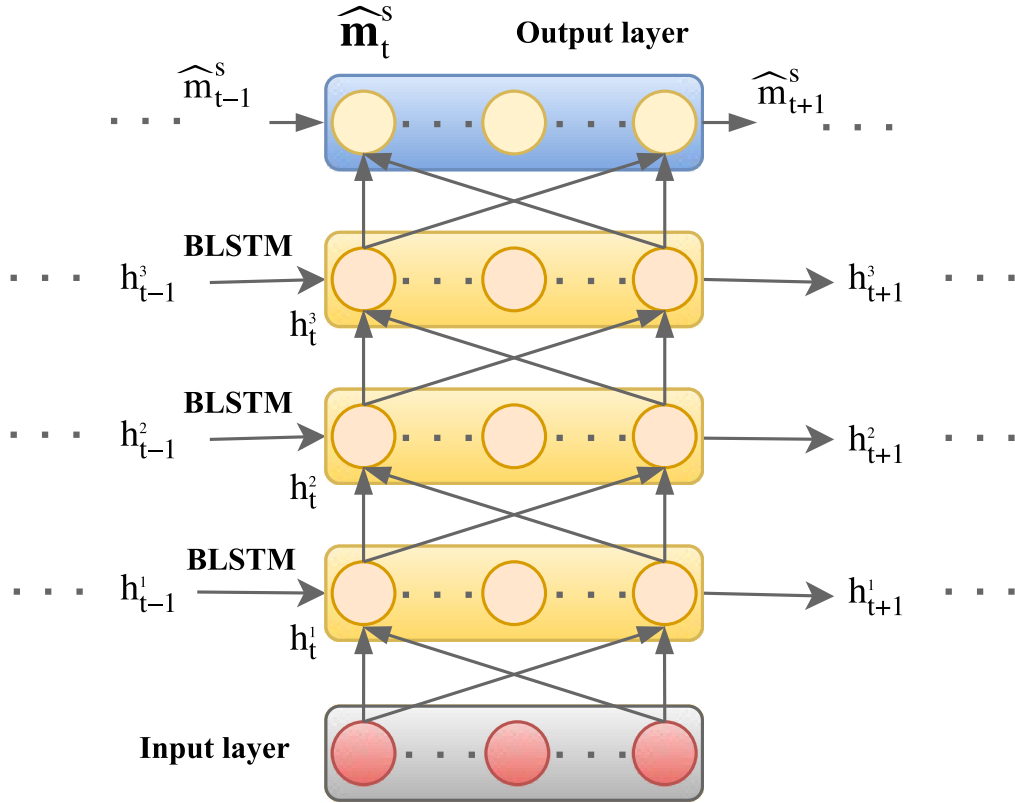


Figure 2.1: Proposed BLSTM network architecture for speech enhancement. Input vector  $\mathbf{v}_t$  is the concatenation of the normalized log-amplitude spectra of  $2c + 1$  frames centered around the  $t$ -th time frame, where  $c$  is the short-term context window parameter. Hidden layer outputs are denoted as  $\mathbf{h}_t^n$ , where  $n$  is the layer index.  $\widehat{\mathbf{m}}_t^s$  is the predicted mask for the speech.

regularization to the network weights during the optimization to overcome overfitting and to increase robustness against unseen noise types. The sigmoid activation function is used in the BLSTM hidden layers.

The BLSTM network is a fully recurrent network, i.e., it only contains BLSTM layers, even in the output. The main difference between the RNN-based method in [38] and our network is that we use BLSTM layers instead of basic recurrent layers. Compared to general RNNs, LSTM units are better at modeling long-term temporal dependencies of data, as it suffers less from the vanishing gradient issue [61]. Our network directly predicts the T-F masks rather than computing it in a deterministic layer as in [38, 40].



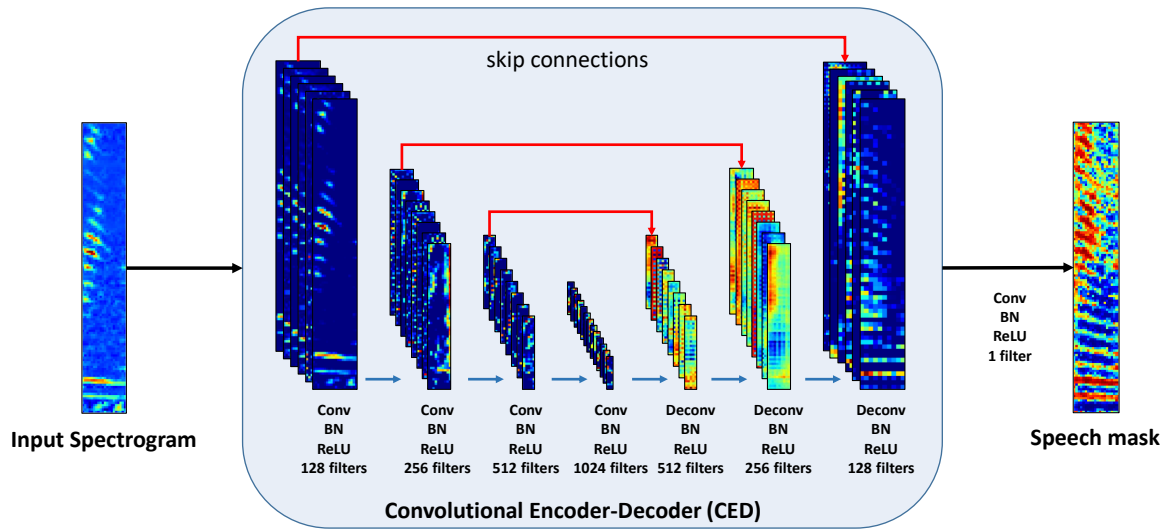


Figure 2.2: Proposed convolutional encoder-decoder (CED) network architecture for speech enhancement. The numbers of filters in the convolution and deconvolution layers are 128, 256, 512, 1024, 512, 256, and 128, respectively. The input is an L-frame magnitude spectrogram, where the output are estimated L-frame mask of speech and noise spectrograms. The red arrows represent the skip connections.

### Convolutional Encoder-Decoder Network

The second network architecture that we propose here is a convolutional encoder-decoder (CED) network, as shown in Figure 2.2. The input layer receives a short-time Fourier transform (STFT) magnitude spectrogram of the noisy speech. This input is then passed to four convolutional layers with a stride length of two forming an encoder, followed by three deconvolutional layers [62] with a stride length of two forming a decoder. This encoder-decoder design compresses and reconstructs the input, and preserves compact and important features. Three skip connections, as denoted by red arrows in the figure, are also added, to help preserve the fine details for better decoding. Finally, a mask for speech is estimated at the output layer. Each of the convolutional and deconvolutional layers also includes a batch normalization (BN) layer and an activation layer with rectified linear unit (ReLU), that are not shown in the figure. The numbers of filters used in all of the convolutional and deconvolutional layers are 128, 256, 512, 1024, 512, 256, 128, and 1, respectively. Filter sizes are

$7 \times 7$  for all layers, except for the output layers, where filter sizes are  $3 \times 3$ . We add  $l_2$  regularization to the network weights during the optimization to overcome over-fitting and to increase robustness against unseen noise types.

This architecture is inspired by [3] and [63]. The main difference between redundant convolutional encoder-decoder (*R-CED*) proposed in [3] and our approach is that we model both speech and noise where R-CED only models the speech. Another difference is that instead of using only 8 STFT frames to denoise a single frame, our network takes much more ( $L = 100$  in the experiments) frames and returns the same amount of mask frames. We divide each test utterance into non-overlapping segments that are  $L$  frames long and feed each segment into the CED network for enhancement. The rationale behind selecting this much larger number of frames to analyze is that it leads to modeling longer-term dependencies and yielding a better reconstruction. In addition, the network depth is also different, R-CED contains 15 layers and is deeper than CED, where each layer contains a convolution, BN and an activation layer, and the proposed CED has 7 layers, where each layer contains three layers, namely a convolution/deconvolution, a BN layer and an activation layer. The number of filters are symmetric in R-CED blocks which are 10, 12, 14, 15, 19, 21, 23, 25, 23, 21, 19, 15, 14, 12, 10, and 1, while the number of filters in the proposed CED are fixed.

## Objective Function

We consider the amplitude soft mask (ASM) in our experiments. ASM for the speech source is defined as

$$\mathbf{m}_t^s(f) = \frac{\mathbf{s}_t(f)}{\mathbf{s}_t(f) + \mathbf{n}_t(f)}, \quad (2.4)$$

where  $\mathbf{s}_t$  and  $\mathbf{n}_t$  are the clean speech and the noise magnitude spectra at time  $t$ , respectively.

To train the networks, we consider two loss functions, the mean-squared error (MSE) and binary cross-entropy (BCE). The MSE objective function minimizes the reconstruction error of the T-F mask of the speech source of the training data as

$$J_{MSE} = \sum_{t,f} \|\mathbf{m}_t^s(f) - \hat{\mathbf{m}}_t^s(f)\|^2, \quad (2.5)$$

where  $\mathbf{m}_t^s$  is the mask calculated from the clean speech and the noise, and  $\widehat{\mathbf{m}}_t^s$  is the mask that is predicted by the network.

The ground-truth ASM speech mask, whose values range from 0 to 1, can be considered as probabilities of T-F bins belonging to the speech source. The predicted speech mask, whose values also range from 0 to 1, thanks to the sigmoid transfer function at the output layer, can be viewed as the predicted probabilities of T-F bins belonging to the speech source. Therefore, BCE can be used to measure the mismatch between the two Bernoulli distributions as

$$\begin{aligned} J_{BCE} &= \sum_{t,f} H(\mathbf{m}_t^s(f), \widehat{\mathbf{m}}_t^s(f)) \\ &= - \sum_{t,f} \mathbf{m}_t^s(f) \log \widehat{\mathbf{m}}_t^s(f) + (1 - \mathbf{m}_t^s(f)) \log(1 - \widehat{\mathbf{m}}_t^s(f)). \end{aligned} \quad (2.6)$$

We compare the MSE and BCE objective functions and analyze their effects on speech enhancement performance in Section 2.1.4.

## 2.1.4 Experiments

We divide the experiment section into two parts. The first part evaluates the speech quality and intelligibility of the speech enhancement approaches on noisy speech utterances that are artificially mixed from clean speech and noise. The clean utterances are not naturally encountered by commercial ASV systems and the mixing process is artificial, however, they are needed for calculating the evaluation measures and are publicly available for results reproduction. The second part connects the proposed approaches with a speaker verification system and evaluates their verification error rates on real-world speech utterances.

For training, we create noisy speech sentences by mixing clean speech utterances from the Librispeech corpus [64] with 138 different types of non-stationary noise obtained from Sound Ideas [65], with SNRs at -6, -3, 0, 3, 6, and 9 dB, totaling about 80 hours of training data. The noise data includes non-stationary noise from various environments such as nature, city, domestic, office, traffic and industry, all of which are what commercial ASV systems may encounter. All files are downsampled

to 8 kHz to simulate the telephone frequency range, since many commercial ASV systems use this range. Our proposed networks described in Section 2.1.3, namely *BLSTM* and *CED*, are trained once and used in all of the experiments described in this section.

### Comparison Methods

As a comparison to our approaches, we trained the fully convolutional redundant CED (*R-CED*) network, described in [3] and in Sections 2.1.2 and 2.1.3, as our convolutional baseline .

We designed another DNN-based baseline identical to our BLSTM architecture, but instead of BLSTM layers it uses general recurrent layers, similar to the approach in [38]. The differences are that we directly predict the masks instead of using a deterministic layer to compute them, and we do not include signal interference terms in the objective function as described in [38]. We call this network recurrent neural network (*RNN*) for simplicity.

We also compare with traditional SE methods described in Section 2.1.2, namely SS and Log-MMSE methods. We use implementations provided in [66].

We implement all DNN-based methods (including the proposed ones) using Keras, a Python library for deep learning [67].

### Speech Quality and Intelligibility Evaluation

We mix 300 utterances of 85 unique speakers with 5 types of noise (babble, factory, speech-shaped noise (SSN), motorcycle and cafeteria) at SNRs of -6, 0, 6 and 9 dB. All of the 85 speakers and the 5 types of noise have not been used as part of the training data. Specifically, the babble and factory noises are obtained from [68], motorcycle noise is obtained from [69], the cafeteria noise is recorded by ourselves at the University of Rochester, and the SSN noise is created by filtering white noise with an FIR filter with frequency response that matched the long-term spectrum of speech utterances [70]. We provide the mentioned test noise samples on our website <sup>1</sup>. Figure 2.3 shows an example noisy spectrogram corrupted by motorcycle noise at 0 dB SNR along with its corresponding clean and enhanced versions. Among the 300 utterances, 120 are from the Librispeech corpus spoken

<sup>1</sup><http://www.ece.rochester.edu/projects/wcng/code.html>

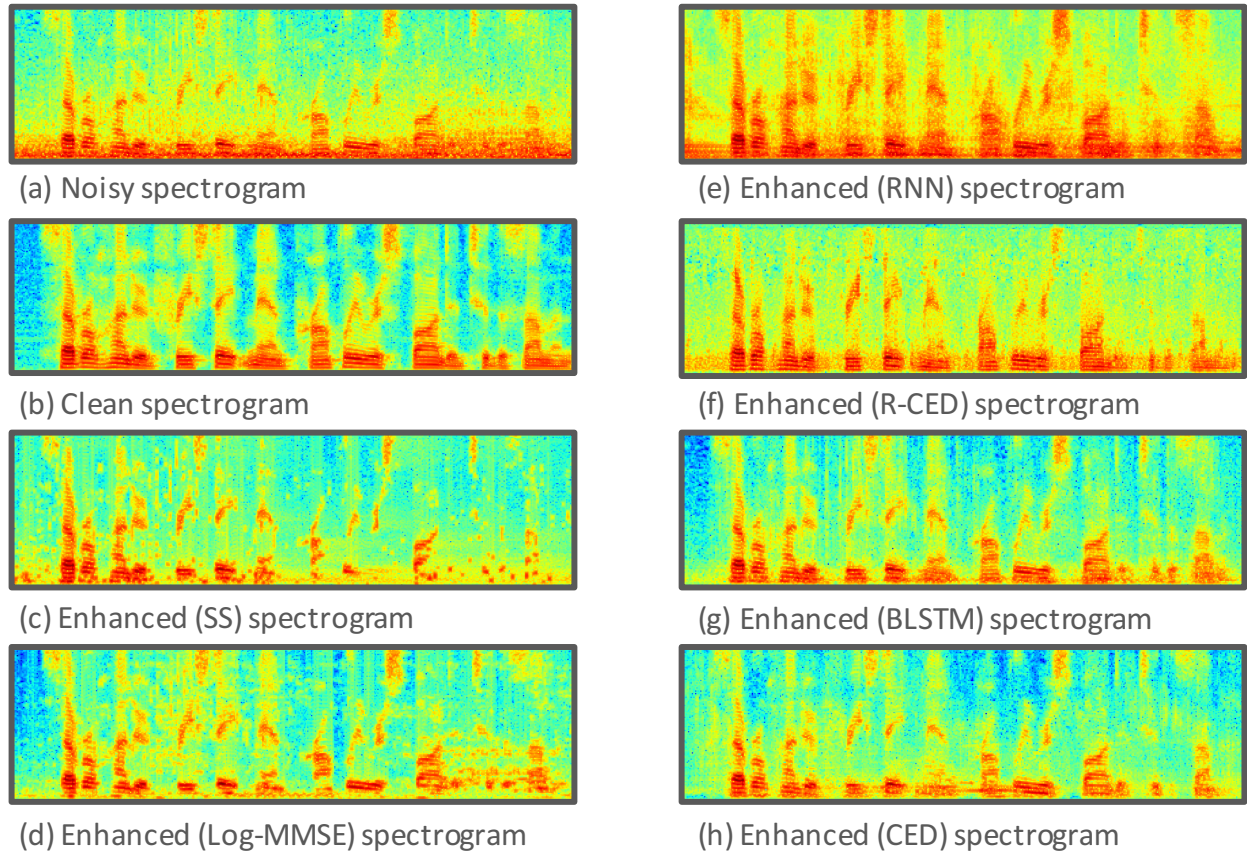


Figure 2.3: An example of speech enhancement results. Magnitude spectrograms of the noisy speech signal corrupted by motorcycle noise at 0 dB, the ground-truth clean speech, and enhanced speech of six speech enhancement methods, namely *SS*, *Log-MMSE*, *RNN*, *R-CED*, *BLSTM* and *CED*.

by 65 unique speakers, and 180 utterances are from the PTDB-TUG corpus [71] spoken by 20 unique speakers. In particular, the inclusion of the PTDB-TUG utterances is to further test the cross-corpora performance of the proposed approach. We use the perceptual evaluation of speech quality (PESQ) [72] and short-time objective intelligibility (STOI) [73] to evaluate our approaches. Both metrics are widely used in SE research. We do not conduct subjective listening tests, as our primary goal in this work is to analyze the effect of DNN-based SE systems on the performance of an ASV system.

For pre-processing, we perform STFT with a 32 ms Hanning window and an 8 ms hop size to obtain the log-amplitude spectrogram of the noisy speech to be input to all networks. We set FFT size to 256 in our experiments and we use the full frequency range of 0 to 4000 Hz. These parameters are kept the same for all of the speech enhancement experiments. We normalize the input to have zero mean and unit standard deviation. For the BLSTM network, we set the short-term context window

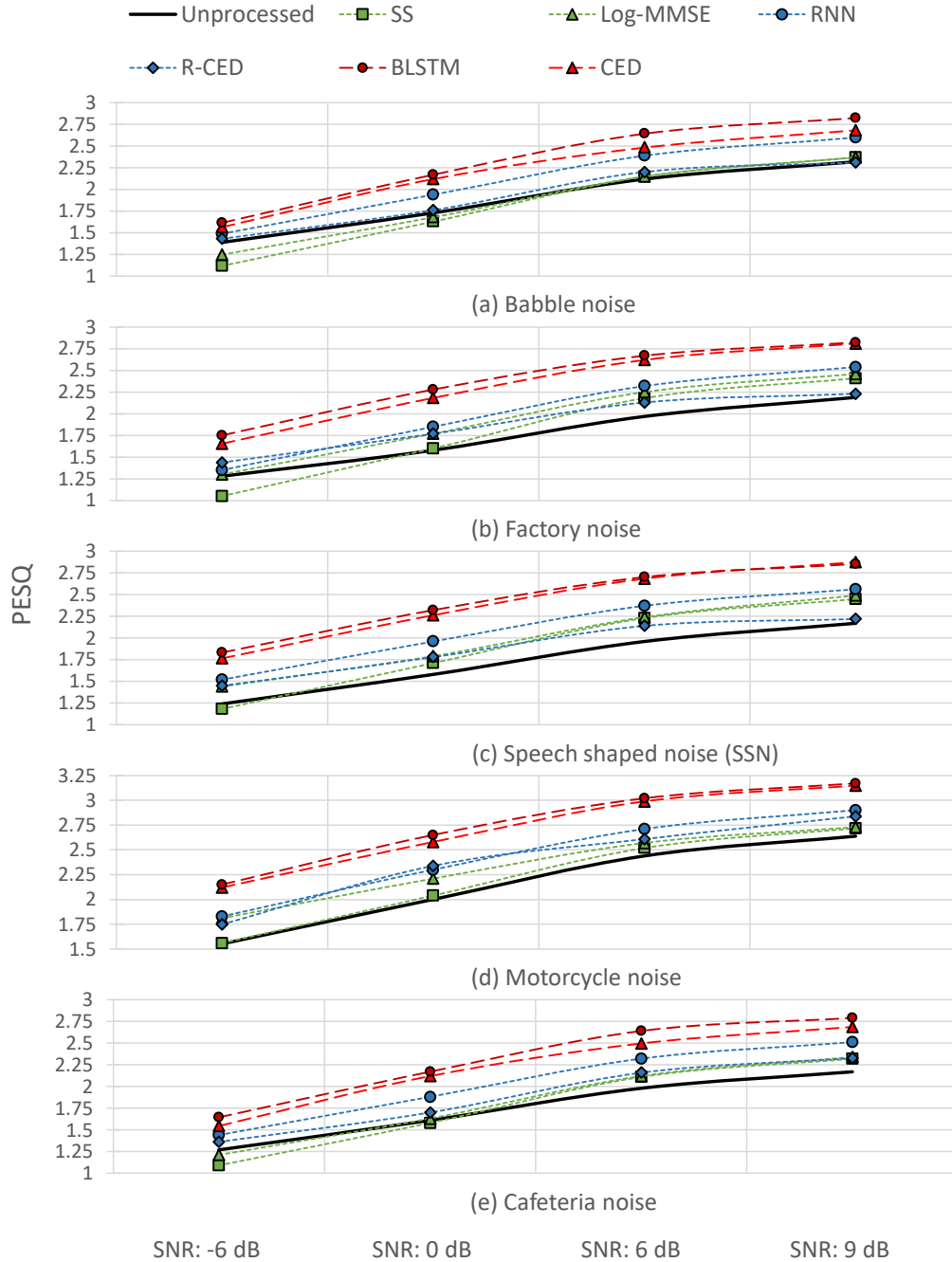


Figure 2.4: PESQ comparison between the proposed methods (*BLSTM* and *CED*) and baseline traditional methods (*SS* [1] and *Log-MMSE* [2]) and baseline DNN-based methods (*RNN* and *R-CED* [3]) for different noise types and SNRs.

parameter  $c$  to 5 frames in all experiments. Increasing this parameter yields faster convergence, but at the cost of computational complexity. We empirically set the time sequence length parameter  $L$  to 100 frames for both networks. Training the networks on the long input sequences makes the networks

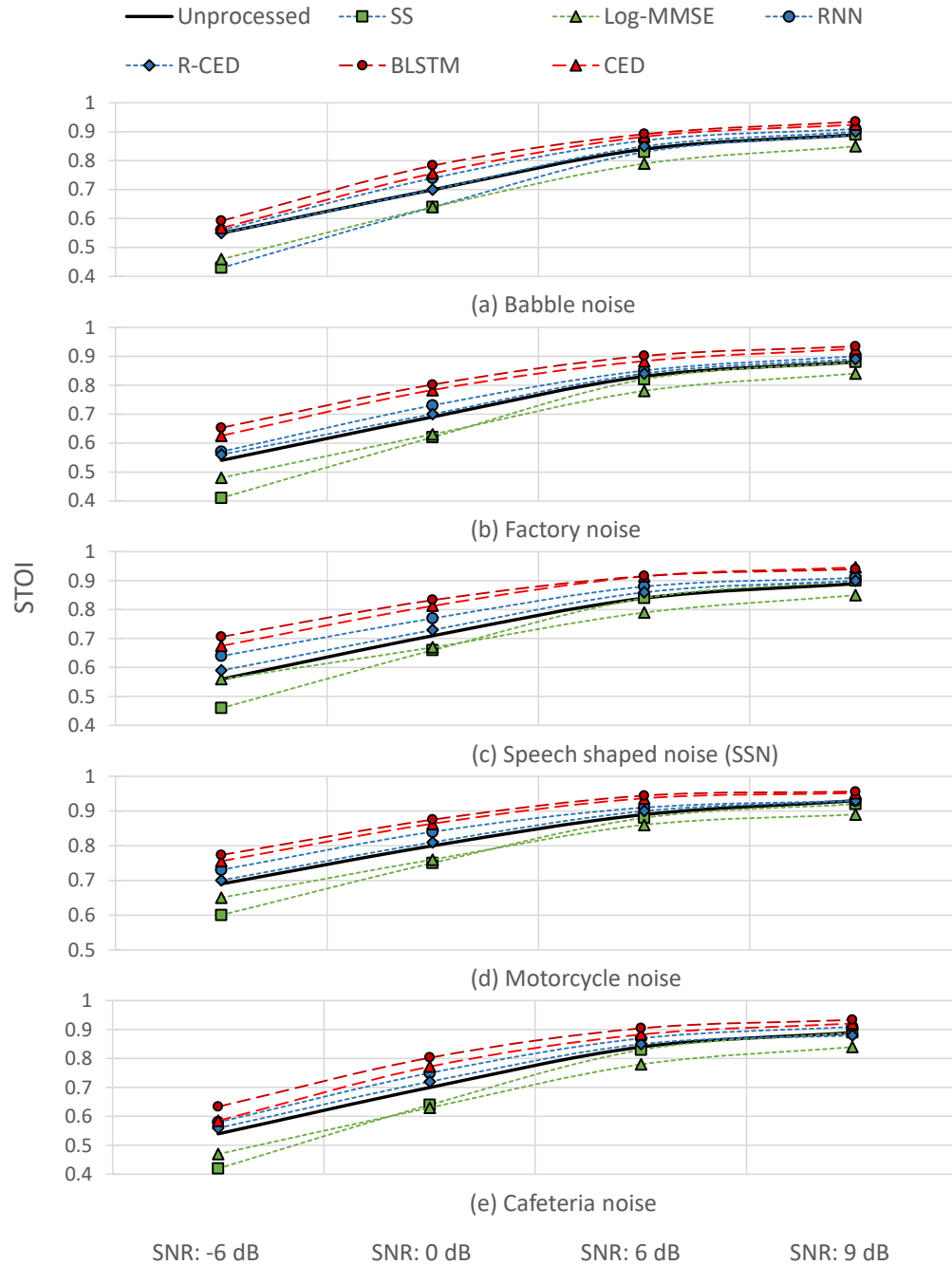


Figure 2.5: STOI comparison between the proposed methods (*BLSTM* and *CED*) and baseline traditional methods (*SS* [1] and *Log-MMSE* [2]) and baseline DNN based methods (*RNN* and *R-CED* [3]) for different noise types and SNRs.

more robust to non-stationary noise, which varies over time.

For training, the dropout rate is set to 0.2 for the BLSTM network, and the  $l_2$  regularization value is set to 0.000001 for both networks. The models are trained for 100 epochs, i.e., we iterate over

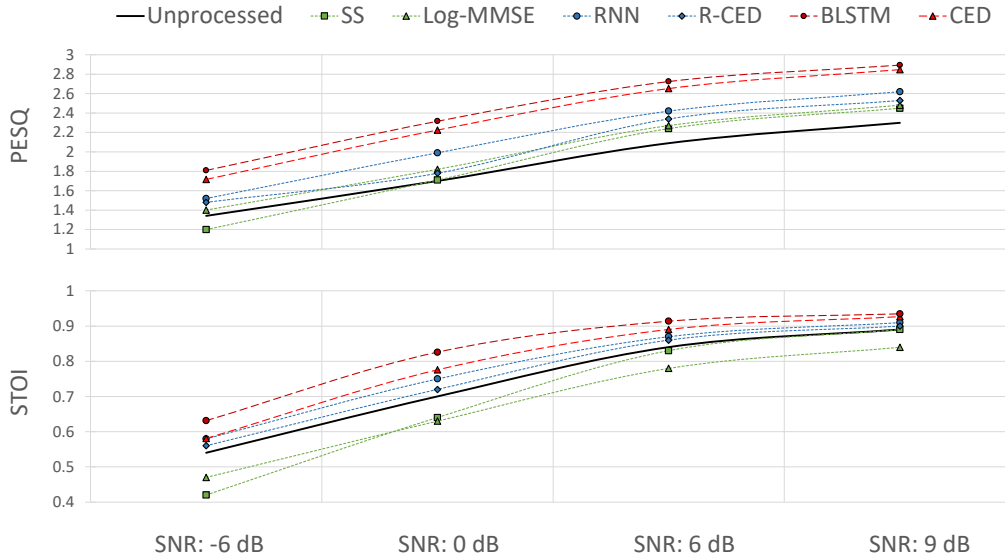


Figure 2.6: PESQ and STOI comparisons averaged over all noise types.

the training set for 100 times. For testing, the network reconstructs the masks of both speech and noise. We then apply the predicted speech mask to the noisy signal’s magnitude spectrogram and then reconstruct its time-domain signal using an inverse STFT with overlap-add from the resulting magnitude spectrogram with the noisy speech’s phase. We trained both networks using only the BCE objective function described in Eq. 2.6, as we found that BCE consistently outperforms MSE in our system analysis experiments in Section 2.1.4.

Figures 2.4-2.6 show the PESQ and STOI results for the unprocessed noisy speech and the enhanced speech using the traditional techniques of spectral subtraction (*SS*) and minimum mean square error log-spectral amplitude estimator (*Log-MMSE*) as well as the DNN-based *RNN*, *R-CED*, and the two proposed networks described in Section 2.1.3, namely *BLSTM* and *CED*.

The results show that the proposed techniques (*BLSTM* and *CED*) are superior than other techniques in terms of the PESQ and STOI metrics in completely unmatched noise types and speaker scenarios. *BLSTM* achieves the best improvement in terms of PESQ and STOI, while *CED* achieves the second best results. *SS* and *Log-MMSE* make the STOI values worse than for the unprocessed noisy speech. We believe that this is due to the musical artifacts introduced by the spectral subtraction operation: the amount of subtraction is determined by the estimated instantaneous SNR, but the estimation does not consider long-term temporal dependencies and leads to fluctuating and inappropriate





Figure 2.7: PESQ and STOI comparisons averaged over all noise types for different numbers of hidden units (64, 128, 256, 512 and 1024) per layer in the BLSTM network.

estimation. This issue also leads to degraded performance of the following ASV system, as shown in Section 2.1.4.

### Parameter Analysis of the Proposed Methods

In this section, we further analyze the effects of several key parameters of the proposed CED and BLSTM networks, including the number of hidden units and layers, the objective function, and the input features. In the following experiments, we use the same settings described in Section 2.1.4, i.e., the train and test speech and noise combinations are the same. We report the average results of five test noise types.

**The Number of Hidden Units** We analyze the effect of different numbers of hidden units in the BLSTM network on PESQ and STOI results. We investigate a three-layer BLSTM network with  $N$  units in each layer, where  $N$  is varied to take values of 64, 128, 256, 512 and 1024. PESQ and STOI results are shown in Figure 2.7. The results suggest that increasing the number of hidden units monotonically improves PESQ and STOI across all SNR conditions, yet the improvement seems to be close to saturation when  $N$  is 1024. Increasing  $N$  beyond 1024 is not feasible for us due to insufficient memory; we used an NVIDIA Tesla K80 GPU which has 12 GB memory.



Figure 2.8: PESQ and STOI comparisons averaged over all noise types for different numbers of filters ( $M = 8, 16, 32, 64$  and  $128$ ) in the first convolutional layer in the CED network. The numbers of filters in the other convolutional and deconvolutional layers are powers-of-two times of  $M$ , following the same symmetric pattern shown in Fig. 2.2.

Next, we investigate the effect of different numbers of filters of the CED network on PESQ and STOI results. The CED network has a symmetric encoder-decoder structure, and the number of filters can be described as  $M, 2M, 4M, 8M, 4M, 2M, M$  for the hidden layers and 1 filter for the predicted speech mask. We vary  $M$  to have values of 8, 16, 32, 64 and 128 and show PESQ and STOI results in Figure 2.8. Again, we can see that increasing  $M$  generally improves PESQ and STOI across all SNR conditions, yet the improvement is very small when  $M$  is greater than 32. Increasing  $M$  above 128 is not feasible for us due to insufficient memory.

In practice, the trade-off between system performance and computational cost needs to be balanced. In our experiments, we chose  $N$  and  $M$  to be 1024 and 128, respectively, to achieve the best possible PESQ and STOI on our device.

**The Number of Hidden Layers** We investigate the effect of the number of layers in BLSTM and CED networks. For the BLSTM network, we let each hidden layer contain 1024 units and vary the number of hidden layers between 1 and 3. The PESQ and STOI results are shown in Figure 2.9. We can see that increasing the number of hidden layers improves both PESQ and STOI across all SNR

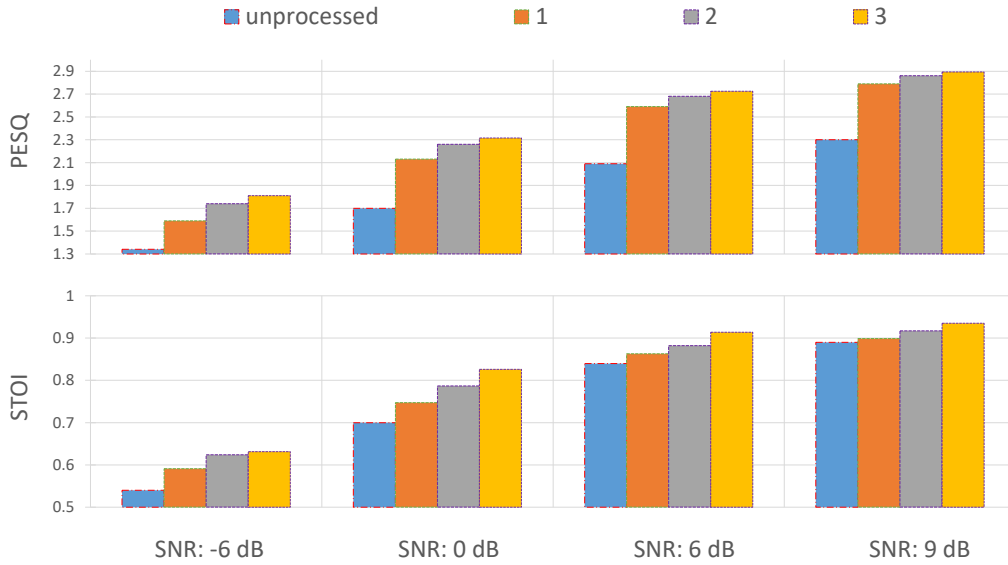


Figure 2.9: PESQ and STOI comparisons averaged over all noise types for different numbers of hidden layers (1, 2 and 3) in the BLSTM network.

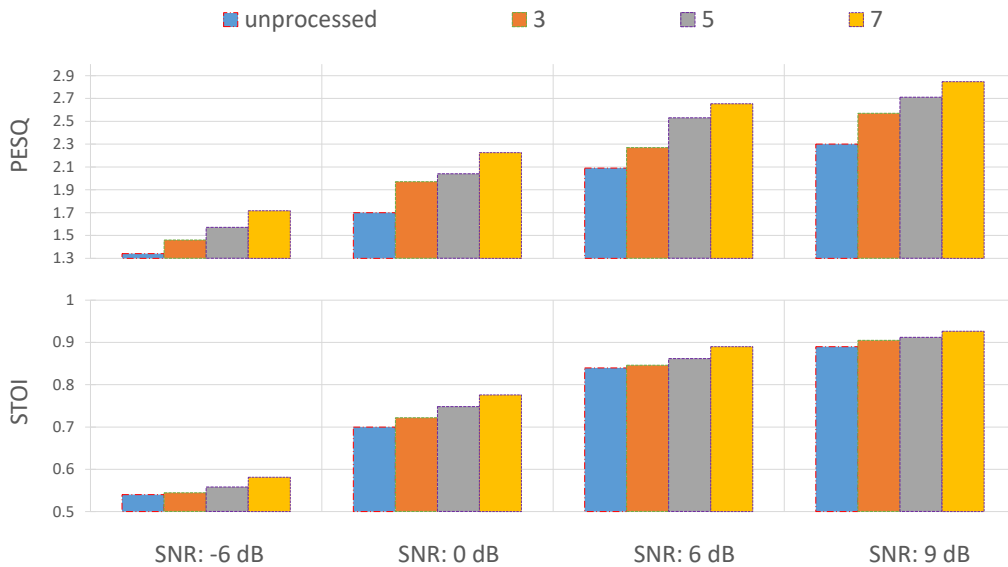


Figure 2.10: PESQ and STOI comparisons averaged over all noise types for different numbers of layers (3, 5 and 7) in the CED network.

conditions. Increasing the number of layers above three is not feasible due to insufficient memory.

The CED network has two parts, the encoder and the decoder. In Figure 2.2, there are a total of 7 layers shown. We vary this number to 3, 5 and 7 and compare their PESQ and STOI performance. The number of filters of the hidden layers follows the same power of 2 ratio as described in the previous subsection, and we set  $M$  to 128. Also note that the number of skip connections also varies to be 1, 2



Figure 2.11: PESQ and STOI comparisons averaged over all noise types for mean-squared error (MSE) and binary cross-entropy (BCE) loss functions in BLSTM and CED networks.

and 3 for networks with 3, 5 and 7 layers, respectively. Results are shown in Figure 2.10. Again, we see that more layers leads to better PESQ and STOI performance across all SNR conditions. However, the number of parameters also increase dramatically, by approximately 11 times from 3 layers to 7 layers.

In our experiments, we set the number of hidden layers to 3 and 7 for the BLSTM and the CED networks, respectively, in order to achieve the best possible PESQ and STOI on our device. Considering the hidden layer size parameters in the previous subsection, the BLSTM and the CED networks have 54,782,992 and 17,669,889 trainable parameters, respectively.

**The Objective Function** This section compares the mean-squared error (MSE) objective function from Eq. 2.5 and the binary cross-entropy (BCE) objective function from Eq. 2.6 for CED and BLSTM networks. The results are shown in Figure 2.11. We can see that the BCE objective function achieves slight but consistent improvement over the MSE objective function on both metrics and networks and across all SNR conditions. Therefore, we use the BCE objective function in all the remaining experiments.

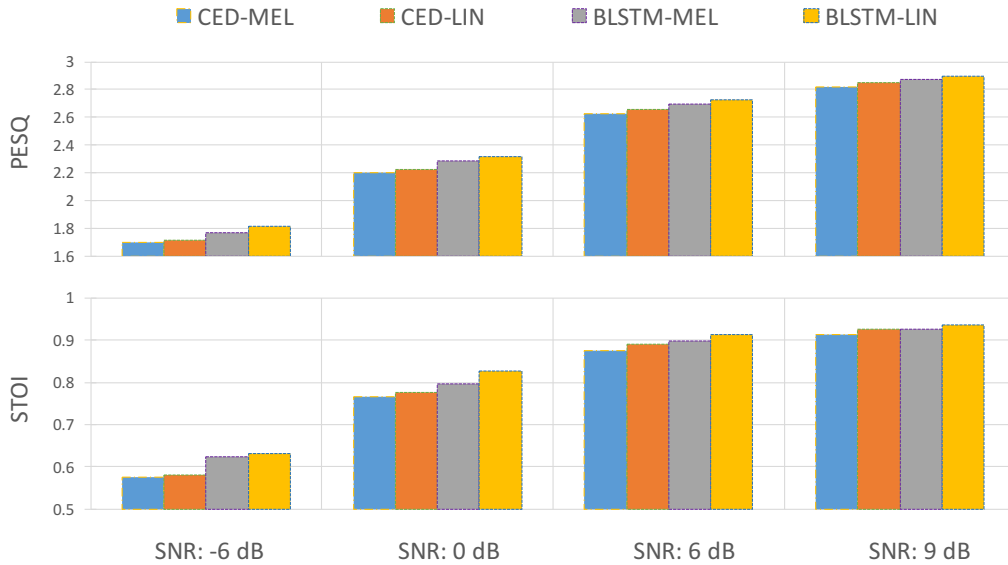


Figure 2.12: PESQ and STOI comparisons averaged over all noise types between log-mel spectrogram (MEL) and log-linear spectrogram (LIN) inputs for BLSTM and CED networks.

**The Input Feature** Next, we compare the log-amplitude linear-frequency (log-linear) spectrogram with the log-amplitude mel-frequency (log-mel) spectrogram as the input feature to the networks. The main difference between these two inputs is the frequency resolution. Compared to the linear-frequency scale, mel-frequency scale has a better correspondence with human auditory systems. It has a higher frequency resolution at low frequencies but a lower frequency resolution at high frequencies. The PESQ and STOI results are shown in Figure 2.12. From the results, we can see that there is a slight difference between the two types of input. The log-amplitude linear frequency spectrogram yields slightly better PESQ and STOI results, therefore, we selected it as our input in other experiments.

### Application in Automatic Speaker Verification

In this section, first we describe the ASV system used for the experiments, and then we use the different speech enhancement methods as a pre-processor for the described ASV system and compare their effects in decreasing the verification error rate.

The i-vector approach is the state of the art in speaker verification and is commonly used in current commercial systems. Therefore, we evaluate our SE system on an i-vector-based text-independent ASV system with probabilistic linear discriminant analysis (PLDA) scoring, which is implemented

based on [74], an open source Python library for speaker and language recognition. We choose this open-source ASV implementation for result reproduction purposes. For all ASV experiments, we use 13 Mel Frequency Cepstrum Coefficients (MFCCs) with their delta and double-delta features, resulting in a 39-dimension vector. The rank of the T matrix, and therefore the dimension of the i-vectors, is set to 100. We found that using low dimensional i-vectors provide better EER results when the utterances are short in duration. We apply length normalization described in [56]. The dimensionalities of the subspaces  $F$  and  $G$  in PLDA training are set to  $100 \times 50$  and 100, respectively.

We use the widely used metric, equal error rate (EER), to evaluate the ASV performance. EER is defined as the intersection point where false rejection rate and false acceptance rate are equal. Lower EER means better ASV performance.

**Datasets** We run our experiments on two datasets: *VBG RANDNUM* and *RedDots*. All of the utterances in both datasets are sampled at 8 kHz, and are natural noisy utterances with a high SNR.

*VBG RANDNUM* is a dataset from the Voice Biometrics Group (VBG)'s production system. It contains 1300 English utterances from 100 speakers, where each speaker has 3 enrollment utterances, and 10 verification utterances. Please note that in our experiments we use multi-session scoring described in [55]. Each utterance contains four random digits and its average length is 6.3 seconds. We estimated the SNR of *VBG RANDNUM* samples using the tool described in [75] with a window size of 8 ms and 50% overlap, and show the SNR distribution in Figure 2.13. We use the enrollment and verification samples of 50 speakers to train the ASV system, namely, the UBM, T matrix and PLDA parameters. These samples already contain natural noise, but we also added artificial noise between 10-25 dB SNR level to 100 randomly chosen samples to obtain a multi-condition training set. We use the remaining 50 speakers for evaluation, where there are in total  $50$  (target speakers)  $\times 10$  (verification utterances)  $\times 50$  (potential speakers) = 25,000 trials in the evaluation. Since this dataset contains constrained speech, we follow the general guidelines described in [50] and keep the number of components used for the UBM small (128 components). Some examples from the *VBG RANDNUM* corpus and their enhanced versions are available for the research community<sup>2</sup>.

---

<sup>2</sup>Free download at <http://www.ece.rochester.edu/projects/wcng/code.html>

This *VBG RANDNUM* dataset is representative of VBG’s RandomPIN™ offering, which is currently deployed (commercially) in 8 countries, using 36 different languages. VBG is currently processing over 6 million RandomPIN™ verification requests annually, and this number is growing rapidly.

To build voice-prints for RandomPIN™, users are prompted to repeat a series of six separate static numeric digit phrases, each five digits long. Note that each RandomPIN™ user is prompted with these same enrollment phrases. To verify the speaker, the VBG system generates a random 4- or 5-digit phrase for the user to repeat.

VBG uses text-independent technology to perform speaker verification. As a pre-processor, VBG uses automatic speech recognition to make sure all content is spoken as requested. A variety of audio quality assessment tests are also performed to ensure the audio is of sufficient quality to perform biometric voice processing. Should samples “fail” content or quality pre-checks as part of a verification request, the system will automatically generate a new random PIN and re-prompt the user.

Using constrained data (digits only) helps the client to create reliable voice-prints in a limited amount of time efficiently. As the majority of VBG’s customers are interactive voice response (IVR) users (stand-alone or as an entry to a call center conversation), telephone connect time (i.e., “call handle time”) becomes a sufficient economic consideration. Thus, shorter and more compact uses of voice biometrics are advantageous. Moreover, when RandomPIN™ is combined with other security factors, such as knowledge-based authentication (KBA), an extremely reliable match can be provided to VBG clients - without the lengthy data collection requirement of free speech or passive voice biometric applications (which VBG also supports commercially).

The second dataset, namely *RedDots* [76], is a collection of short utterances in English from native and non-native speakers reading text prompts to mobile devices. The sessions are collected over a long period (aimed to be over a year), where each speaker records a session per week. The dataset contains 13 female and 49 male speakers from different regions worldwide, a total of 21 countries, which results in vast inter-speaker variations. Since the data collection is carried out from a mobile device, the user can choose to record an utterance in any place, indoor or outdoor. Therefore utterances contain various types of noise with various SNRs. We estimated the SNR of the *RedDots*

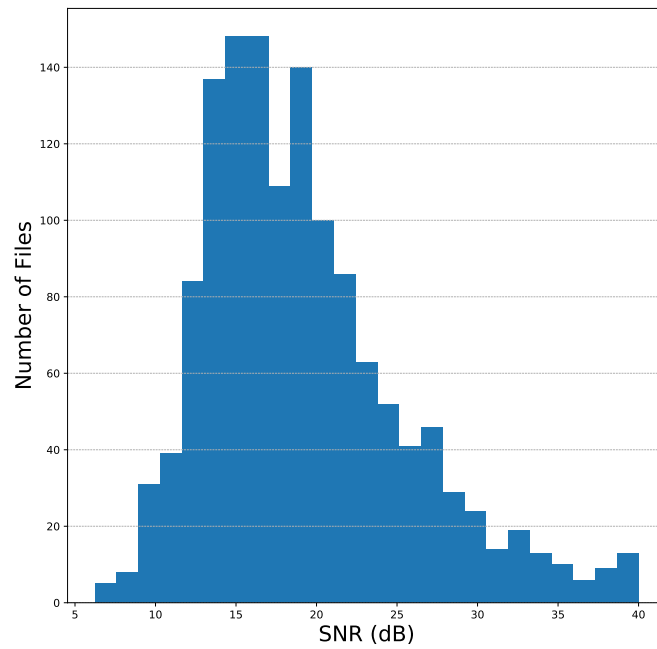


Figure 2.13: Histogram of SNR estimation of *VBG RANDNUM* files.

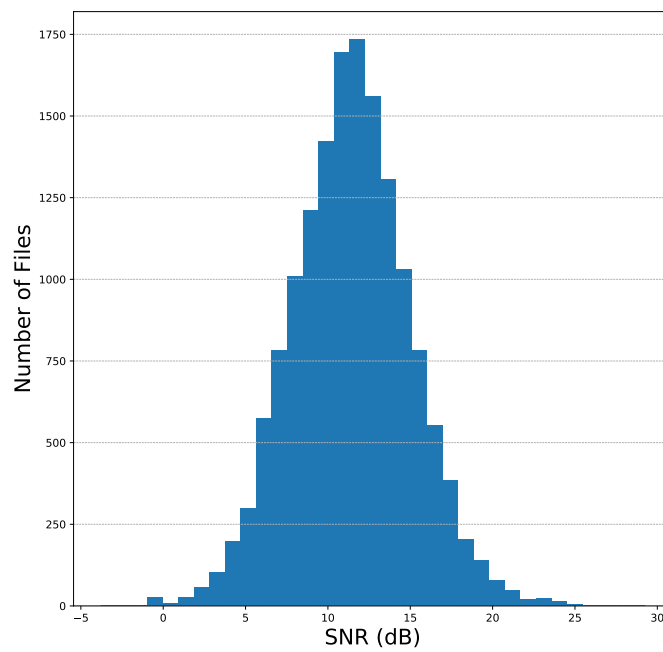


Figure 2.14: Histogram of SNR estimation of *RedDots* files.



samples in the same way that we estimated the SNR of the *VBG RANDNUM* samples. Figure 2.14 shows the SNR distribution for the *RedDots* samples.

We conduct our experiments in a text-independent fashion. Therefore, we use *RedDots part 04: text-independent* test set. There are a total of 136,698 target trials and 5,098,950 imposter trials for males, and 26,928 target and 184,368 imposter trials for females in this test set. Since the number of female samples are relatively limited in this dataset, we only use male trials in our experiments, different from the gender-independent case in the experiments with the *VBG RANDNUM* dataset.

To conduct a more comprehensive evaluation in different noise conditions, we also mix *RedDots* test utterances with five types of noise at SNRs of -6, 0, 6 and 9 dB to create more noisy utterances and report their ASV results. To construct the UBM and i-vector models (i.e., the  $T$  matrix), we use two other datasets, NIST SRE06 [77] and the NIST SRE08 [78]. We randomly draw 650 male speakers from these datasets' training set. We also added artificial noise between 10-25 dB SNR level to 150 randomly chosen samples to obtain a multi-condition training set. Since the test samples are unconstrained speech, we set the number of mixtures in the UBM to 2048 in our experiments, as suggested in [50]. Finally, we used the remaining male data in the *RedDots* dataset that is not included in the trials to train PLDA parameters.

**Evaluations** Figure 2.15 and Figure 2.16 show the EER results for speech that is unprocessed as well as speech that is enhanced with *SS*, *Log-MMSE*, *RNN*, *R-CED*, *CED* and *BLSTM* for *VBG RANDNUM* and *RedDots* datasets, respectively.

For constrained speech data, Figure 2.15 (*VBG RANDNUM*) shows that *BLSTM* significantly decreases the EER compared to other techniques, from the unprocessed EER (%) result of 6.59 to 5.21. This is followed by *CED* with an EER (%) value of 5.78. The gap between *BLSTM* and *CED* EER results are significant, although their PESQ and STOI values shown in Figures 2.4-2.6 are close. While the reason for this mismatch is unclear, this result suggests that speech quality and intelligibility measures for speech enhancement preprocessing modules only provide qualitative predictions of the final speaker verification error rates. *RNN* yields slightly better results compared to *R-CED*, which is consistent with the PESQ and STOI results. An important observation from these results is that there

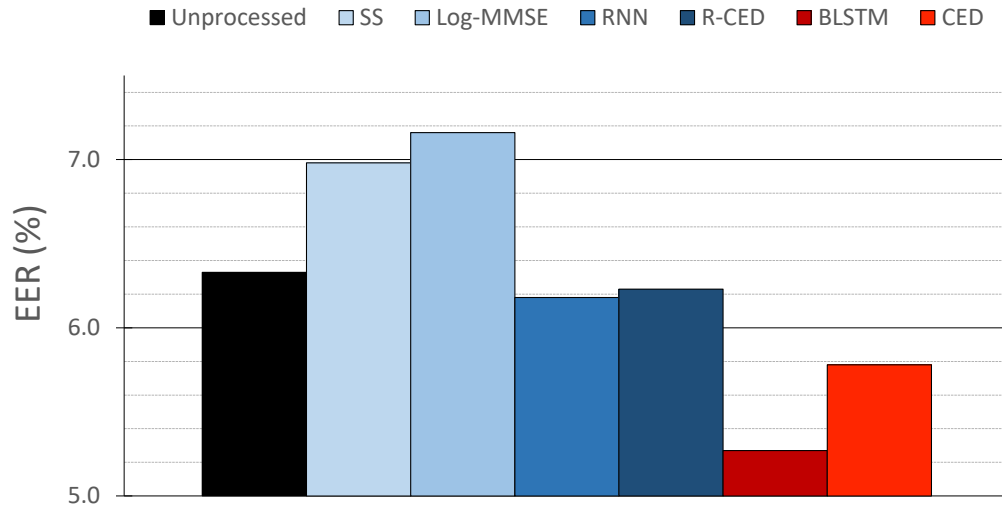


Figure 2.15: *VBG RANDNUM* EER results. Note that the y-axis starts from 5.0%.

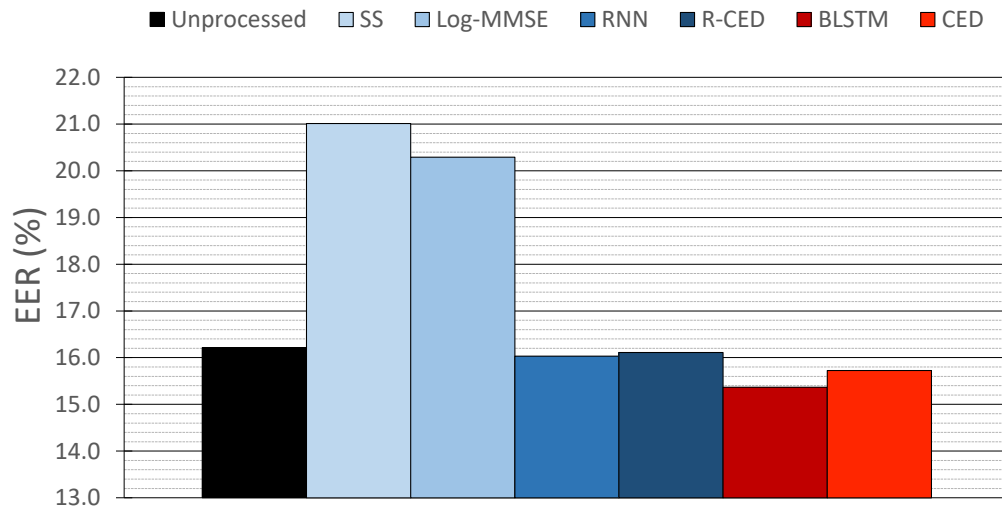


Figure 2.16: EER results for *RedDots* Dataset. Note that the y-axis starts from 13.0%.

is a benefit of using DNN-based approaches as a front-end SE module since all DNN-based methods yield EER improvements on naturally noisy data. *SS* and *Log-MMSE*, however, significantly increases EER, showing that they cannot deal with non-stationary noise conditions well.

The same trends can be observed for unconstrained speech data (*RedDots*) results shown in Figure 2.16, although the improvement on EER of the DNN-based methods are slighter compared to the *VBG RANDNUM* results. *SS* and *Log-MMSE*, again, do not perform well in this dataset.

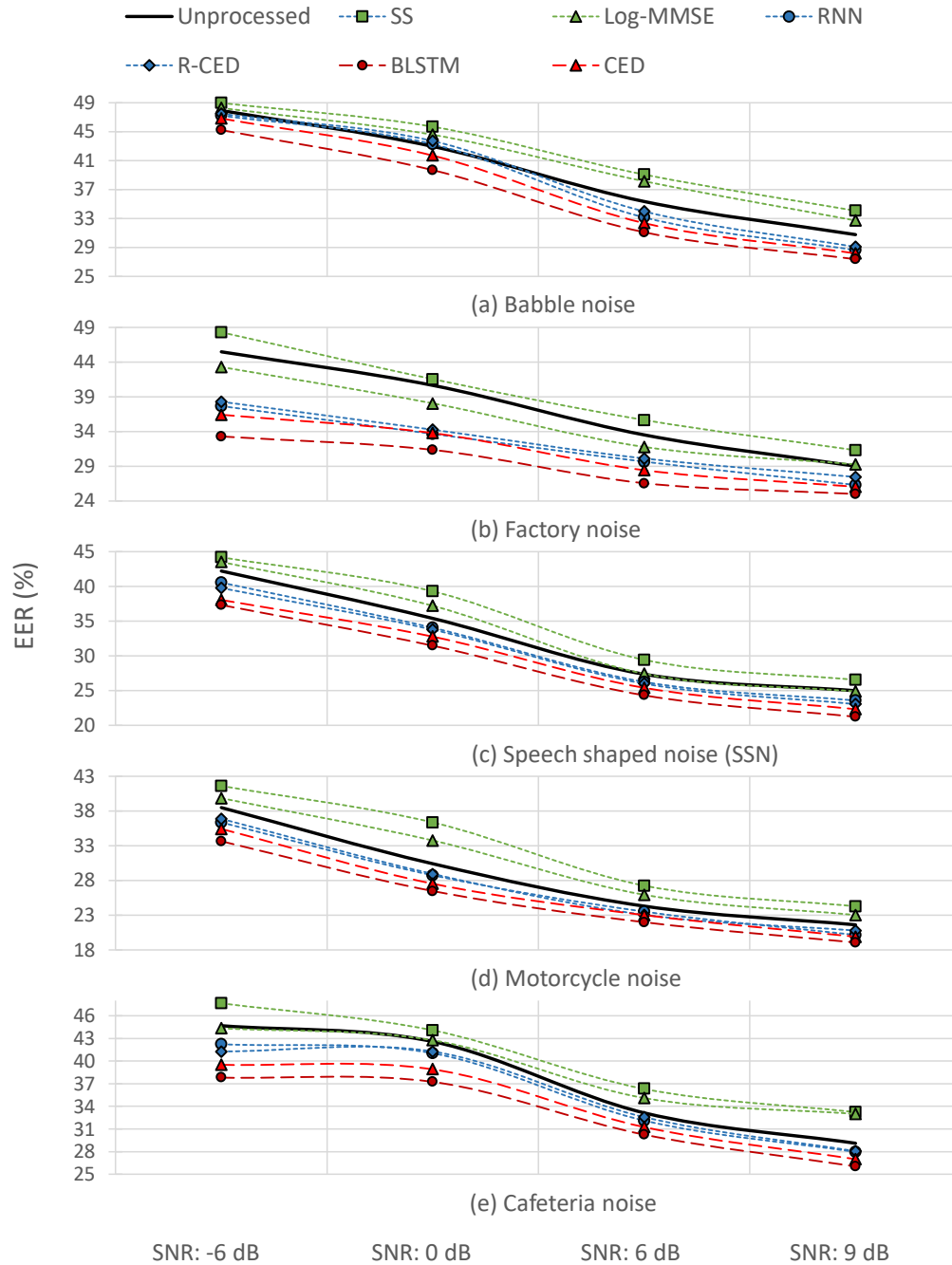


Figure 2.17: *RedDots* dataset EER results for different noise types and SNRs.

Figure 2.17 shows the artificial test results, i.e., the EER results when additional noise is introduced at an SNR of -6, 0, 6 and 9 dB to the *RedDots* dataset. For all noise cases, the SS method increases the EER. The Log-MMSE method yields EER improvements in low SNRs for factory and cafeteria noise types, however, it does not provide EER improvements for all the other noise types

and SNRs. The DNN-based methods yield EER improvements in most cases. The BLSTM network performs the best for all noise types.

### 2.1.5 Conclusions

In this work, two DNN-based speech enhancement methods (BLSTM and CED) are introduced, and their effect as a preprocessor for an automatic speaker verification (ASV) system is investigated. Compared to two classical and two DNN-based speech enhancement baselines, the proposed methods significantly improve the PESQ and STOI of the enhanced speech on different kinds of non-stationary noise that are unseen in the training data. Moreover, they decrease the verification error rate on natural utterances encountered by the verification system and on utterances artificially mixed with additional noise. We show that all DNN-based methods investigated in this work yield performance improvements when they are used as a front-end noise removal module on natural noisy data collected from real customers, while the classical methods degrade the performance in the same conditions.

## 2.2 Adversarial Training for Speech Super-Resolution

### 2.2.1 Introduction

Deep neural networks (DNNs) have been outperforming traditional methods in various classification and regression tasks, and speech processing is not an exception. For speech recognition, enhancement, emotion recognition, and speaker identification/verification, state-of-the-art methods are based on DNNs.

An interesting problem in speech processing is to expand the bandwidth of speech signals by generating the missing high frequencies (i.e., increasing the waveform resolution). This problem is named *artificial speech bandwidth expansion* or Speech Super-Resolution (SSR) in the literature. In this work, we tackle this problem and refer it SSR.

SSR is beneficial for speech communication over low-bandwidth channels. An SSR module can be integrated into receiver-end devices to enhance the resolution of transmitted low-resolution signals.

One study shows that users prefer a wider frequency range in communication [79]. Other studies show that the narrowband communication is challenging for the hearing impaired population [80], and artificially expanding the bandwidth up to 8 kHz leads to improved speech recognition rates for Cochlear Implant (CI) users [81]. Furthermore, speech synthesis systems can also benefit from employing a computationally light-weight SSR module after synthesizing low-resolution speech. This is because the computational cost of speech synthesis drastically increases as the sampling rate increases, preventing a real-time high-resolution synthesis on edge computing devices. Also, speech synthesis systems, once trained, are not straightforward to change the sampling rate on the fly.

In this work, we propose a novel neural network framework that leverages adversarial training for SSR, and utilize a recent regularization method that stabilizes the adversarial training. We employ a sequence-to-sequence convolutional autoencoder network that accepts Log Power Spectrogram (LPS) as input and yields the corresponding high-frequency range LPS. We use 1D kernels in the convolutional layers that operate along the time axis of the spectrogram. The training process contains two major steps. First, we train our network using only a reconstruction loss for a few epochs as the initialization. Then, we switch to the adversarial loss in addition to the weighted reconstruction loss.

We train our network on the Centre for Speech Technology Research (CSTR) Voice Cloning Toolkit (VCTK) Corpus [82] and evaluate it on an entirely disjoint dataset to show the robustness against unseen speakers and recording conditions, namely the Wall Street Journal (WSJ0) corpus [83]. We compare with [84, 85] baselines. The objective and subjective evaluations show that the resulting enhanced time domain signals yield better results than the baseline methods. We further analyze our network by changing the network parameters, namely the number of layers and filters in the autoencoder, and the reconstruction loss weight parameter, and report the objective scores. Besides, we discuss the stability of GAN training for different regularization methods and compare phase estimation methods. Furthermore, we compare the computational complexity of our method and the baselines. We also propose a method to train the network against the noise, and we analyze it against the unseen non-stationary noise types. In addition, we conducted a listening test to verify the intelligibility of the generated samples. Some examples of synthesized super-resolution speech are

publicly available<sup>3</sup>.

In summary, our contributions in this work are as follows:

- We apply the generative adversarial network framework to speech super-resolution and synthesize the high-resolution speech spectrogram directly with the network.
- We use a regularization method [86] to address the failure modes encountered during GAN training, and effectively stabilize it.
- We obtain a computationally light-weight generator compared to the baselines due to the usage of 1D kernels in the convolutional layers.

## 2.2.2 Related Work

### Artificial Speech Bandwidth Expansion

Speech Super-Resolution (SSR) is studied widely by the research community under the name of *artificial speech bandwidth expansion* [87, 88, 89, 84]. In [87], Park et al. used Linear Predictive Coding (LPC) coefficients, pitch, and power that were extracted from the narrowband signal, and modeled the mapping between narrowband and wideband parameters using a Gaussian Mixture Model (GMM). Chennoukh et al. [90] proposed a method that extends the bandwidth using Line Spectral Frequencies (LFS), applied on LPC coefficients. Seo et al. [89] proposed a GMM model for maximum a posterior estimation of the wideband spectrum from the narrowband. This method also considers sentence-level temporal dynamics to synthesize wideband speech. Jax et al. [91] proposed a method to estimate the gain and the shape of the spectral envelope of the wideband using a Hidden Markov Model (HMM). Song et al. [92] showed that the Baum-Welch re-estimation algorithm outperforms the method proposed by Jax et al. [91]. They also showed that the GMM-based methods are a special case of the HMM-based methods, while their performances are comparable. Abel et al. [93] proposed to use DNNs for high band spectral envelope estimation, and compared with GMM and HMM-based baselines. They showed that DNNs outperform the baselines.

---

<sup>3</sup><http://www.ece.rochester.edu/projects/air/projects/SSRGAN.html>

While some works focused on predicting the wide-band spectral envelope, others focused on directly estimating the missing data points [94, 95, 84, 85]. In [94], the authors used a latent component analysis and Expectation-Maximization (EM) algorithm to estimate missing frequencies, similar to Non-negative Matrix Factorization (NMF). Sun et al. [95] cast the bandwidth extension problem as a convex optimization problem and employed NMF to estimate the missing frequencies. In one of the notable works, Li et al. [84] proposed a DNN to predict the log-power spectrum of the wideband. They used 32 ms window size and 16 ms hop size when extracting LPS features from the input narrowband. The hidden layers were pre-trained using the Restricted Boltzmann Machine (RBM). Their network accepts nine frames of narrowband LPS and predicts a single frame of wideband LPS. Since phase information is still missing, they flip the phase of the low-frequency band as that of the high-frequency band to reconstruct the time domain signal. They trained and evaluated their method on the Wall Street Journal (WSJ0) Corpus. They showed that their method yields better results compared to the GMM baseline in both objective and subjective evaluations.

Kuleshov et al. [85] proposed an end-to-end super-resolution method that takes the raw waveform as input and outputs the super-resolution waveform. They employed 1D convolution layers and formed an auto-encoder with concatenating skip connections, which are similar to skip connections but instead of adding the feature maps together, they are concatenated. Before being fed to the network, the low-resolution waveform is upsampled to match the sampling rate of the target super-resolution signal. This upsampled input is also added to the network output. A Mean-Squared Error (MSE) loss function is used for training. Compared with neural methods working with time-frequency representations, one significant advantage of this time domain approach is that no special module is needed to estimate the signals' phase. However, it is computationally very expensive.

### **Generative Adversarial Networks (GANs)**

Generative Adversarial Networks (GANs) [96] have been employed to generate highly realistic images, videos and speech signals. In essence, GANs contain two neural networks, a generator, and a discriminator. The generator tries to generate fake but realistic data, while the discriminator tries to distinguish between the real and fake data. When the training converges, the generator is able

to generate data that lie on the real data manifold, and the discriminator cannot tell the fake from real data. There are variants of GANs, which improve the generation capability or add controls over the generated distributions. Deep Convolutional GAN (DCGAN) [97] can generate realistic images, where both the generator and discriminator architectures are based on convolutional neural networks. The conditional GANs [98] are another family of GANs where the generator and discriminator accepts a condition input and enables control over the generated distribution.

Although GANs are powerful, they suffer from instabilities during training [99], which lead GANs not to converge and make them yield poor results. Therefore, researchers steered towards finding better training methods for GANs [100, 101, 102, 86, 99]. Wasserstein GAN (WGAN) [100] is one of the regularized GAN family members that employs the Wasserstein divergence instead of the Jensen-Shannon divergence and maintains the Lipschitz constraint by clipping the weights. In an improved version of WGAN [101], instead of weight-clipping, Gulrajani et al. proposed a Gradient-Penalty (GP) to satisfy the Lipschitz constraint. In the proposed method, the data point between a real and generated distributions is drawn, and the norm of the gradient for this data point is penalized for not having a unit norm. For WGAN and WGAN-GP, the critic (discriminator) is usually updated for a few iterations before alternating to updating the generator, which makes the training computationally intense. Another regularization technique is to add instance noise, which is typically chosen as Gaussian noise, to the input of the discriminator [102]. Mescheder et al. [99] show that instance noise is indeed useful for GAN training, and leads GANs to converge. Roth et al. [86] derived a zero-centered GP regularizer that is inspired from the instance noise. Mescheder et al. [99] proposed two similar but simplified versions of Roth et al.'s regularizer, one of them only penalizes the generated data distribution, while the other one only penalizes the real data distribution. In this work, we choose to penalize both the real and generated distribution; therefore we use the regularizer proposed by Roth et al. [86].

GANs have been successfully applied to image and video super-resolution. Ledig et al. [103] confirmed that reconstruction loss based single image super-resolution systems yield blurry results. By using an adversarial training loss, they showed that their Super-Resolution Generative Adversarial Network (SRGAN) yields sharper, superior results that lie on the data manifold. GANs also benefit Video Super-Resolution (VSR). Lucas et al. [104] showed that their GAN based VSR system



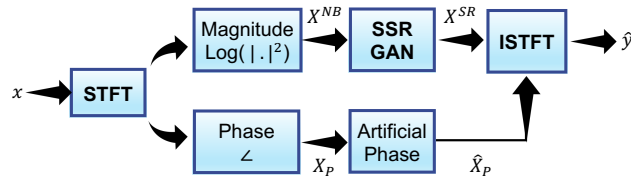


Figure 2.18: Overview of the proposed SSR system during test time. The Log-Power Spectra (LPS)  $X^{NB}$  and the phase spectrogram  $X_P$  are calculated from the input narrowband waveform  $x$  through Short-Time Fourier Transform (STFT).  $X^{NB}$  is fed to the speech super-resolution generative adversarial network (SSR-GAN) to obtain the estimated high-frequency range LPS  $\hat{X}^{WB}$ , which is then concatenated with the original narrowband LPS. The phase of the high-frequency range is artificially produced by flipping and repeating the narrowband phase  $X_P$  and adding a negative sign. For fractional super-resolution factors, the last repeat is truncated to match the frequency range. Finally, the estimated wideband LPS and artificial phase are used to reconstruct the time-domain signal  $\hat{y}$  by Inverse STFT (ISTFT) and overlap-add.

outperforms the current state-of-the-art VSR systems. These studies inspired us to investigate the application of GANs to SSR, where we work with spectrograms that are similar to images or video frames.

It is noted that Li et al. [105] has proposed a GAN-based SSR approach recently. They employed a fully connected neural network (generator) with two hidden layers to predict the Line Spectral Frequencies (LSF) and speech energy of the high band (HB) from LSF, delta LSF and speech energy of the low band signal. They used a fully connected discriminator to distinguish fake parameters from real parameters. They then used the EVRC-WB framework [106] and a synthesis filterbank to synthesis high-resolution speech signals from the predicted speech parameters. Although our approach is similar to [105] in the sense that they are both applications of GANs in SSR, one of the key differences is that we directly generate the speech spectrograms, while [105] generates speech parameters (LSF + energy) and synthesize speech from those parameters with another synthesis framework. Another novelty of our work is that we use a recently proposed regularizer [86] to stabilize GAN training. Furthermore, our generator and discriminator architectures contain convolutional layers, while [105] uses only fully connected layers.

### 2.2.3 Proposed SSR System

#### System Overview

We propose a neural network approach with adversarial training to tackle the Speech Super-Resolution (SSR) problem. Before we introduce the network architecture and training processes, we think it is helpful to first explain how the whole SSR system runs during test time, treating the network as a black box. This process is shown in Figure 2.18. Let  $x$  be the time domain waveform of the narrowband speech that we want to increase the time resolution. First, the Short-Time Fourier Transform (STFT) is applied to  $x$  with parameter settings described in Section 2.2.4. The Log-Power Spectrogram (LPS)  $X^{NB}$  and the phase spectrogram  $X_P$  are computed from  $X$ , and  $X^{NB}$  is fed to the proposed generator network, or namely the Speech-Super Resolution Generative Adversarial Network (*SSR-GAN*) to estimate the high-frequency range LPS,  $\hat{X}^{WB}$ . The original narrowband and the predicted high-frequency range are concatenated to obtain the estimated wideband LPS  $X^{SR}$ . In order to avoid discontinuities at the concatenation [84], we also predict the highest  $C$  frequency bins of the narrowband spectrogram, where  $C$  is called the *offset* parameter. During concatenation, the top  $C$  frequency bins are removed from the narrowband spectrogram.

Since we do not estimate the phase of the high frequencies, we follow Li et al. [84] to create an artificial phase by flipping the narrowband phase and reverting the sign. For the 2x super-resolution version, we concatenate this flipped phase with the narrowband phase to obtain an artificial phase  $\hat{X}_P$  of the entire wideband signal. For the 4x super-resolution version, we repeat the flipped phase three times. For fractional super-resolution factors, the last repeat is truncated to match the frequency range. Our method could be improved by predicting the phase from the magnitude spectrogram; however, this is a challenging problem itself [107].

Finally, inverse STFT is applied to the complex spectrogram calculated from the estimated wideband LPS  $X^{SR}$  and artificial phase  $\hat{X}_P$ , and the time domain signal  $\hat{y}$  is reconstructed using the overlap-add method.

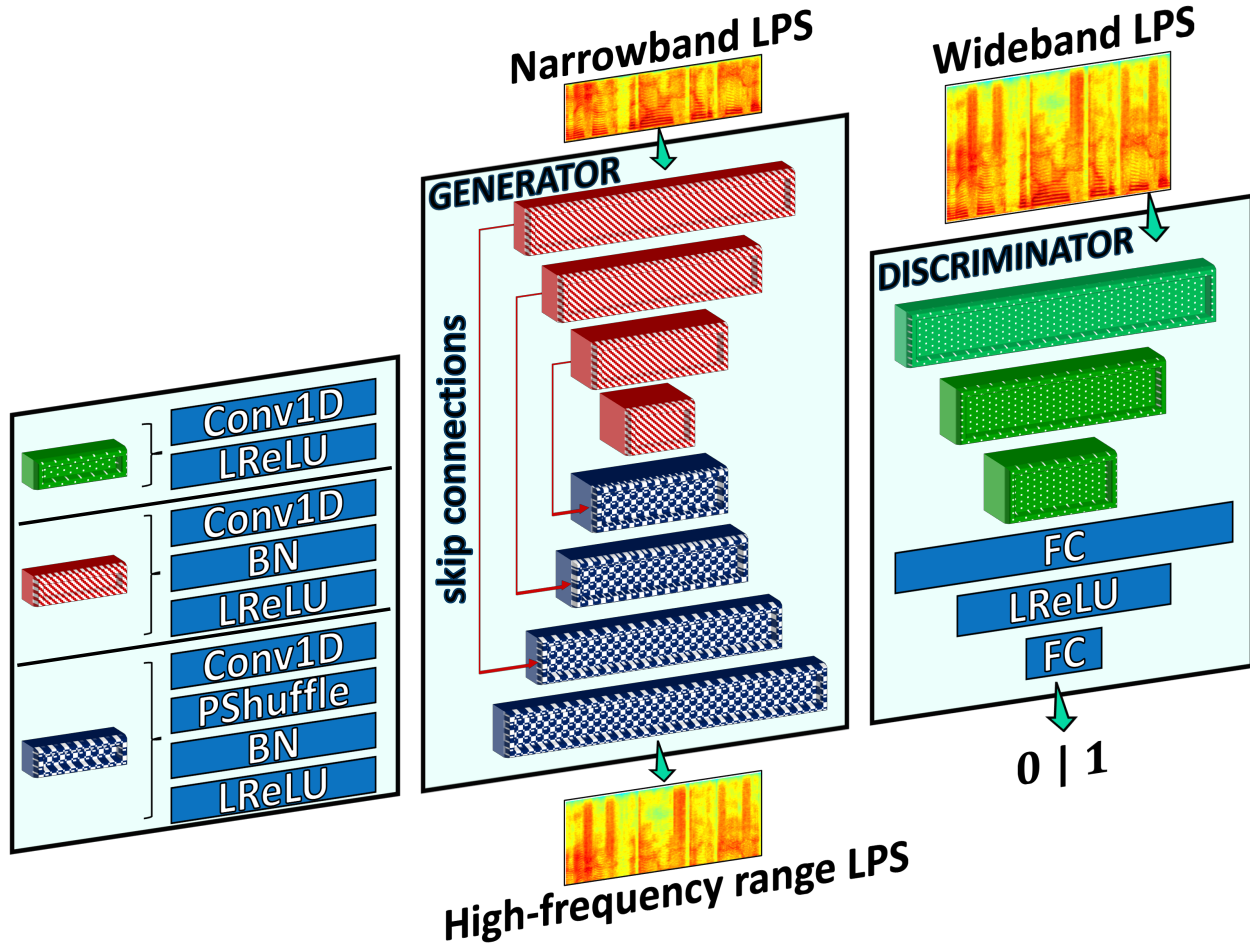


Figure 2.19: The proposed network architectures for the generator (middle) and the discriminator (right). Each rectangular block is a convolutional layer with structures color coded and detailed on the left subfigure. The generator is an autoencoder with concatenating skip connections, predicting the high-frequency range of the input narrowband magnitude spectrogram. It is then concatenated with the original low-frequency range to generate the full wideband magnitude spectrogram. The input to the discriminator is the full wideband spectrogram of either a real sample or a generated sample. We do not use batch normalization in the discriminator. Notations: *BN* - batch normalization layer, *FC* - fully connected layer, *LReLU* - LeakyReLU activation, and *PShuffle* - pixel shuffle or sub-pixel layer, *LPS* - log-power spectrogram.

## Network Architecture

In this section, we explain the generator and discriminator architectures. The generator is fully convolutional, while the discriminator contains convolutional layers followed by two Fully Connected (FC) layers. The architectures are shown in Figure 2.19.

For the generator network, we employ a common bottleneck autoencoder architecture described

in [85]. The generator is a sequence-to-sequence model that accepts the narrowband LPS with  $T$  time steps and outputs the high-frequency range LPS with  $T$  time steps.

In the generator network, we use a Batch Normalization (BN) layer after each convolutional layer and before the activation. BN allows the network to converge faster and allows higher learning rates to be used for training. We use sub-pixel (or pixel shuffle) layers introduced in [108], which is proved useful for image and video super-resolution. The main idea behind the sub-pixel layers is to compute more feature maps on the convolution layer and resize them into an upsampled data. Readers are referred to see [108] for more details about sub-pixel layers. We use leaky rectified linear units (LeakyReLU) as the activation with a slope of 0.2, except for the output layer, where we use linear activation.

The discriminator network accepts the concatenated narrowband and high-frequency range LPSs as input, where the high-frequency range LPS could be generated by the generator network or coming directly from the data distribution. Including the narrowband to the discriminator's input is essentially conditioning the input high-frequency range LPS on the narrowband LPS, similar to conditional GANs [109]. The discriminator contains three convolutional layers as shown in Figure 2.19. Different from the generator, we do not employ BN layers in the discriminator. Using BN in the discriminator leads to instabilities during training, especially if the discriminator loss is regularized [86, 99]. The convolutional layers are followed by two FC layers. We use LeakyReLU activation with a slope of 0.2 in all layers, except for the output layer, where we use a linear activation function. The details of both network architectures are shown in Table 2.1.

## Loss Functions

In this section, we describe the training objectives of the generator and the discriminator. First, we train our network using a reconstruction loss as initialization for several epochs. This process lets the generator to produce the “mean” results, which are overly smooth. Then, we switch to using both the reconstruction loss and an adversarial loss (GAN loss). Using GAN loss produces sharper and more detailed LPSs. We use a parameter to weight these two losses in the generator's objective function. In the following, we explain the details for each loss function.

Table 2.1: Detailed parameters of the proposed network architecture. The number of channels and hidden units, filter sizes, strides, activations and output shapes are shown for each layer in the generator and discriminator networks.  $K$  and  $N$  are the narrowband and the high-frequency range LPS dimensions along the frequency axis, respectively.  $K$  is 129 and 65 for 2x and 4x super-resolution scales, respectively.  $N$  is 141 and 199 for 2x and 4x super-resolution scales, respectively.

Net	Layer	Activation	Filter No.	Filter Size	Stride	BN	Sub-Pix	Output Shape
Generator	Input	-	-	-	-	-	-	$32 \times K$
	Conv	LeakyReLU	256	(7, 1)	(2, 1)	Yes	No	$16 \times 256$
	Conv	LeakyReLU	512	(5, 1)	(2, 1)	Yes	No	$8 \times 512$
	Conv	LeakyReLU	512	(3, 1)	(2, 1)	Yes	No	$4 \times 512$
	Conv	LeakyReLU	1024	(3, 1)	(2, 1)	Yes	No	$2 \times 1024$
	Conv	LeakyReLU	512	(3, 1)	(1, 1)	Yes	Yes	$4 \times 512$
	Conv	LeakyReLU	512	(5, 1)	(1, 1)	Yes	Yes	$8 \times 512$
	Conv	LeakyReLU	256	(7, 1)	(1, 1)	Yes	Yes	$16 \times 256$
	Conv	LeakyReLU	$N$	(7, 1)	(1, 1)	Yes	Yes	$32 \times N$
	Conv	LeakyReLU	$N$	(9, 1)	(1, 1)	No	No	$32 \times N$
Discriminator	Input	-	-	-	-	-	-	$32 \times (K + N)$
	Conv	LeakyReLU	1024	(7, 1)	(2, 1)	No	No	$16 \times 1024$
	Conv	LeakyReLU	1024	(5, 1)	(2, 1)	No	No	$8 \times 1024$
	Conv	LeakyReLU	1024	(3, 1)	(2, 1)	No	No	$4 \times 1024$
	Flatten							4096
	FC	LeakyReLU	2048			No		2048
	FC	Sigmoid	1			No		1

**Reconstruction Loss** There are a few candidates for the reconstruction loss. The most common distance functions are L1-norm and L2-norm, or namely, Mean Absolute Error (MAE) and Mean Squared Error (MSE). Our initial testing showed that using Log-Spectral Distance (LSD) (or Log-Spectral Distortion) function as our training objective yield slightly better results for the SSR task. The LSD measures the distance between two spectrograms in decibels, and it is mathematically defined as follows:

$$l_{LSD} = \frac{1}{L} \sum_{l=1}^L \sqrt{\frac{1}{K} \sum_{k=1}^K [X^{HR}(l, k) - X^{SR}(l, k)]^2}, \quad (2.7)$$

where  $X^{HR}$  and  $X^{SR}$  are the ground truth and estimated LPS, respectively  $K$  is the number of frequency bins. LSD is used widely for evaluating SSR methods objectively. In this work, we use it as both the reconstruction loss and an objective evaluation metric. LSD is essentially the average L2 distance of LPS across time frames.

**Adversarial Loss** The original generative adversarial network (GAN) is a two player, zero-sum (minimax) game between a generator and a discriminator. The generator’s job is to generate realistic data that can fool the discriminator into classifying it as real data, while the discriminator’s job is to distinguish the real and fake data apart. When this game reaches a Nash equilibrium, the generator is

able to produce realistic data that the discriminator cannot tell from real data. In the SSR context in this work, this two-player game can be defined as follows:

$$\begin{aligned} \min_{\theta} \max_{\psi} \mathbb{E}_{\mathbb{P}}[\log D_{\psi}(X^{HR})] + \mathbb{E}_{\mathbb{Q}}[\log(1 - D_{\psi}(G_{\theta}(X^{NB})))] \\ \mathbb{P} : X^{HR} \sim p(X^{HR}) \\ \mathbb{Q} : X^{NB} \sim p(X^{NB}) \end{aligned} \quad (2.8)$$

where  $X^{HR}$  is the high resolution data (real data),  $X^{NB}$  is the narrowband data.  $G_{\theta}(\cdot)$  is the generator and  $D_{\psi}(\cdot)$  is the discriminator, where  $\theta$  and  $\psi$  represent their trainable parameters.  $\mathbb{P}$  is the distribution of real data and  $\mathbb{Q}$  is the distribution of the narrowband data. This formulation assumes the generator contains the concatenation of narrowband LPS and high-frequency LPS. This equation can be simplified as follows:

$$\min_{\theta} \max_{\psi} \mathbb{E}_{\mathbb{P}}[\log \varphi_R] + \mathbb{E}_{\mathbb{Q}}[\log(1 - \varphi_F)], \quad (2.9)$$

where  $\varphi_R$  and  $\varphi_F$  are the discriminator output for real and fake data, respectively.

In practice, unregularized GANs are usually unstable during training, depending on the task at hand, and do not always converge [99]. To stabilize the GAN training, we add a penalty on the weighted gradient-norms of the discriminator as described in [86]. The regularization term is described as:

$$\Omega = \mathbb{E}_{\mathbb{P}}[(1 - \varphi_R)^2 \|\nabla \phi_R\|^2] + \mathbb{E}_{\mathbb{Q}}[\varphi_F^2 \|\nabla \phi_F\|^2], \quad (2.10)$$

where  $\phi = \sigma^{-1}(\varphi)$ , and  $\sigma$  is the sigmoid activation used in generating the output of the discriminator. Note that the gradients are computed on  $\phi$ , before the sigmoid activation, which yields more robust gradients [86]. We add this term into the traditional GAN loss and obtain the loss for the discriminator as follows:

$$l_{DIS} = \mathbb{E}_{\mathbb{P}}[\log \varphi_R] + \mathbb{E}_{\mathbb{Q}}[\log(1 - \varphi_F)] - \frac{\gamma}{2} \Omega, \quad (2.11)$$

where  $\gamma$  is the weight for the regularization term.

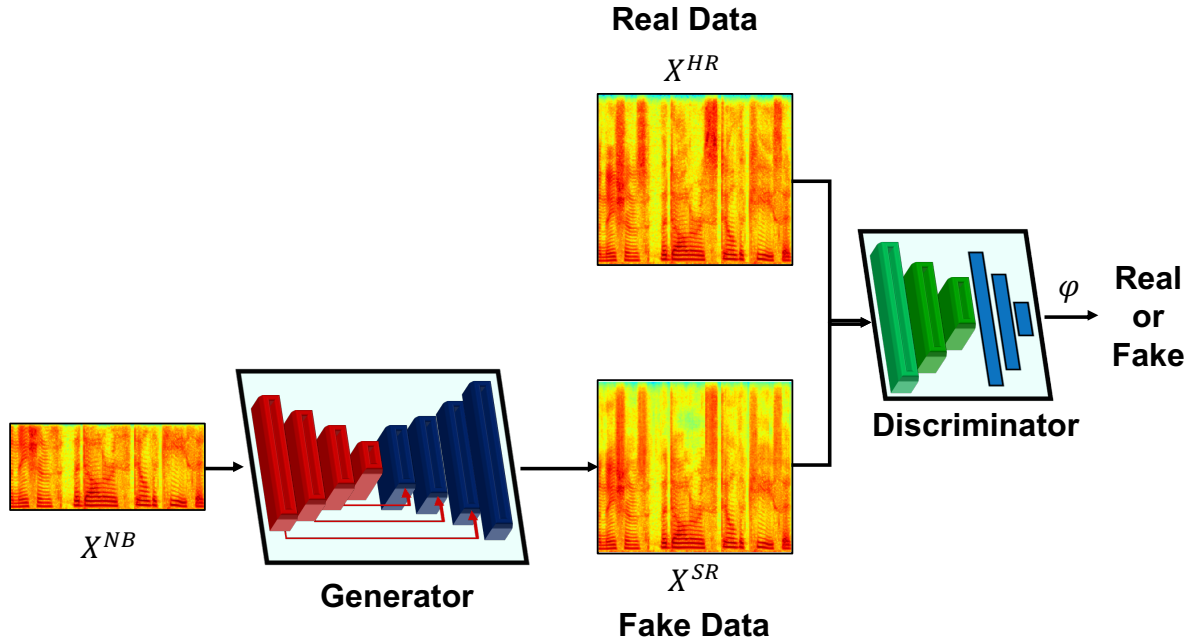


Figure 2.20: The adversarial training procedure for the proposed method. The generator contains the concatenation of narrowband LPS and high-frequency LPS.

The generator loss is defined as the weighted sum of the reconstruction loss and the adversarial loss. We minimize the following function:

$$l_{GEN} = \mathbb{E}_{\mathbb{Q}}[-\log(D_{\psi}(G_{\theta}(X^{NB})))] + \lambda l_{LSD}, \quad (2.12)$$

where  $l_{LSD}$  is the loss function described in Equation (2.7) and  $\lambda$  is the weighting factor for the LSD loss.

## 2.2.4 Experiments

In this section, first, we describe the data used in this study and how we prepared the data for network training. Next, we describe the objective metrics used for evaluating our method. Then, we show the results of our experiments and analyze our network architecture by changing parameters. Next, we investigate our network's resilience to background noise, propose a training method to make the network robust against noise. Finally, we conclude this section by describing and presenting the

results of a subjective evaluation of our method.

## Datasets

The CSTR Voice Cloning Toolkit Corpus (VCTK), which is originally designed for training Text-to-Speech (TTS) synthesis systems, was used to train our network. There are a total of 109 English speakers with different accents. The recordings are 16-bit WAV files with 48 kHz sampling rate and contain clear speech. Each speaker utters 400 sentences, where the sentences are either taken from newspaper articles, the International Dialects of English Archive’s Rainbow passages or an elicitation passage that aims to identify the speaker’s accent. We split this dataset into training and validation sets, where we randomly chose six speakers for the validation set and use the rest for the training set.

We employed another dataset for evaluation that has different speakers and different recording conditions than the VCTK corpus, in order to evaluate the generalization capability of our network. This is the Wall Street Journal (WSJ0) corpus, where the speakers read the Wall Street news articles plus spontaneous dictations. The sampling rate of the recordings is 16 kHz. The recordings contain natural background noise. We randomly selected 5000 samples (around 12 hours) within this dataset for the objective evaluations.

We applied a low-pass filter and downsampled the high-resolution signals to obtain their parallel low-resolution signals for training and testing.

## Objective Metrics

To evaluate our method and compare it with the baselines, we employed LSD defined by Equation (2.7), Segmental Signal-to-Noise Ratio (SegSNR) [110], and Perceptual Evaluation of Speech Quality (PESQ) [72] objective metrics, which are widely used for SSR and speech enhancement literature. LSD measures the similarity between two spectrograms in decibels and defined in Equation (2.7), where a lower value is better. SegSNR is the signal-to-noise (SNR) ratio, averaged over segments of



audio samples. It is defined as:

$$\text{SegSNR} = \frac{1}{L} \sum_{l=1}^L 10 \log \frac{\sum_{n=1}^N [x(l, n)]^2}{\sum_{n=1}^N [x(l, n) - \hat{x}(l, n)]^2}, \quad (2.13)$$

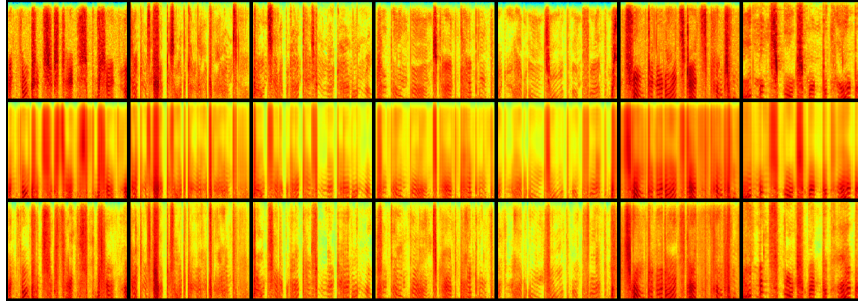
where  $L$  is the number of segments, and  $N$  is the number of data points in the utterance. A higher value of SegSNR is better.

PESQ measures speech quality and it is standardized by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T). It is widely used in industry to assess the quality of telephony speech and in research fields such as speech enhancement. PESQ ranges from -0.5 to 4.5, where higher values correspond to better speech quality.

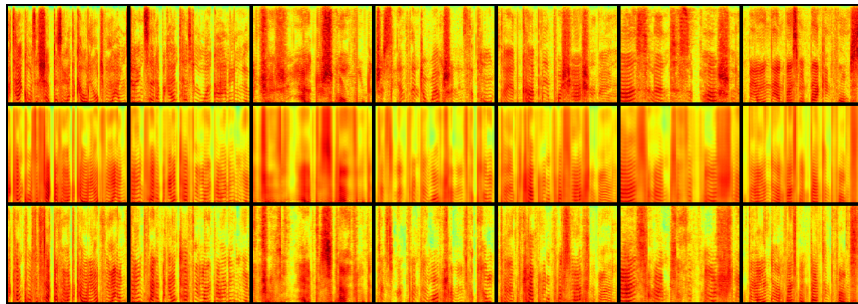
## Baseline Methods

We chose two state-of-the-art methods described in Section 2.2.2 as comparison baselines. The first baseline is an FFT-based method [84], which we name as *BL1* through the rest of the paper. The neural network architecture of *BL1* is a DNN with three hidden layers with 2048 hidden units per hidden layer. The network accepts nine STFT frames, including four past and four future frames, and generates a single STFT frame. The objective function of this network is MSE. We implemented *BL1* as described in the original paper, except that we used VCTK corpus for training in order to fairly compare all methods. Since this work only considers 2x SSR, we did not implement 4x SSR version of this work.

The second baseline is a waveform-based method [85], which we name as *BL2* through the rest of the paper. Similar to ours, this network is a convolutional autoencoder, although our network is applied to spectrograms instead of waveforms. Another difference is that their network has an additive residual connection between the input and output of the network. The number of filters of the convolutional encoder layers is 128, 256, 512, and 512, and is 512 for the bottleneck layer. The decoder has twice the number of filters in the encoder layers but in reverse order. The size of filters of the convolutional encoder layers is 65, 33, 17, and 9, and is 9 for the bottleneck layer. The size of filters in the decoder layers are the same as the encoder but in reversed order. Their network is



(a) 2x SSR results



(b) 4x SSR results

Figure 2.21: Spectrogram examples for 2x and 4x, shown in (a) and (b), respectively. The samples are randomly selected from the WSJ0 corpus (unseen speakers). The first row in each Figure shows the ground truth high-frequency range spectrograms. The second and third rows show the generated high-frequency range spectrograms of the proposed network trained with only the LSD loss (second rows) and with both LSD and GAN losses (third rows).

trained with the MSE objective function. For implementation, we used the code provided by the authors directly to generate results for both 2x and 4x SSR, using the hyperparameters described in their paper. To ensure fairness, we used the exact same data we used for our method during training and testing the baselines.

### Pre-Processing

For our method, we applied the band-limited sinc interpolation method described in [111] to the high-resolution signal and obtained the downsampled signal. We computed the short-time Fourier transform (STFT) on both low and high-resolution signals, with 32 ms window size and 8 ms hop size. We applied the log and power operations to these spectrograms to obtain log-power spectra (LPS). We chopped up the utterances into  $T$  timesteps and form our dataset with narrowband and

high-frequency range LPS pairs.

Similarly, for *BL1*, we followed the same steps. However, we followed their original implementation, and instead of 8 ms hop size, we used 16 ms hop size.

For the pre-processing for *BL2*, we used the author’s code, which is available online. The low-resolution signals were created by applying an order 8 Chebyshev type I low-pass filter and downsampling the high-resolution signals. The low-resolution signals were upsampled to match the size of the high-resolution signals using cubic upscaling as the input to their neural network. The samples were chopped into patches with the length of 6000 in the high-resolution space (0.375 seconds), which is the same for 2x and 4x scales.

### Implementation Details of Proposed Method

We implemented our system in Tensorflow [112]. We used mini-batches during training, and we set the mini-batch size to 64. We trained our network using only the LSD loss for 50 epochs, and then switched to LSD plus GAN loss for 100 epochs. We decided the number of epochs empirically. We still use LSD loss during GAN training, which keeps the output around the mean distribution as discussed in [103]. The number of time-steps  $T$  of our input and output spectrograms is 32. We used a learning rate of  $10^{-4}$  when training the network using only LSD loss, and we used a learning rate of  $10^{-5}$  for both the generator and discriminator when training the network using LSD plus GAN losses. We chose lower learning rate during GAN training to further stabilize it. The  $\lambda$  value is set to 0.5. We used Adam optimizer [113] to train our generator and RMSProp optimizer [114] to train the discriminator. The  $K$  variable shown in Table 2.1 is 129 for 2x experiments and 65 for 4x experiments. The frequency offset value is calculated according to the following formula:

$$C = \text{floor}\left(\frac{K}{10}\right) + 1, \quad (2.14)$$

where  $K$  is the number of frequency bins in the input spectrogram. The  $N$  variable shown in Table 2.1 is 141 and 199 for 2x and 4x super-resolution scales, respectively. The  $\gamma$  variable shown in Equation (2.11), which weighs the regularization term for the discriminator, is set to 2. Please note that we did

Table 2.2: The objective evaluation results for 2x and 4x SSR experiments. The bolded values show the best results. Our method (*SSR-GAN*) outperforms the baselines for all metrics. *LSD HF* shows the LSD value calculated only for the high-frequency range, where *LSD Full* shows the LSD value calculated for the whole spectrogram.

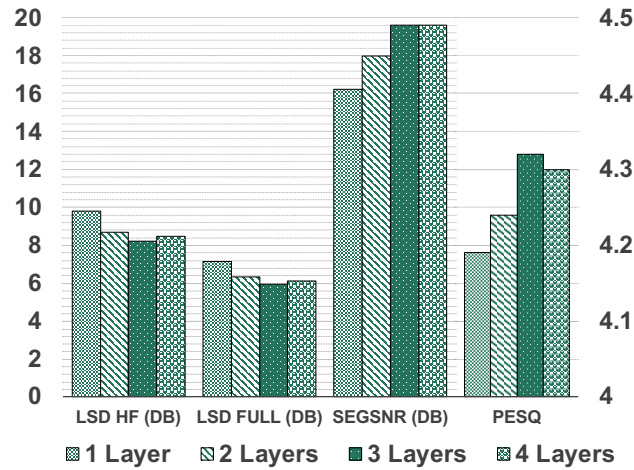
Scale	Method	LSD HF (dB)	LSD Full (dB)	SegSNR (dB)	PESQ
2x	BL1 [84]	9.32	7.06	15.73	4.21
	BL2 [85]	10.56	7.64	14.96	4.19
	SSR-LSD	8.60	6.09	17.58	4.25
	<b>SSR-GAN</b>	<b>8.20</b>	<b>5.95</b>	<b>19.64</b>	<b>4.32</b>
4x	BL2 [85]	16.20	14.96	8.24	2.89
	SSR-LSD	14.10	12.42	11.78	3.26
	<b>SSR-GAN</b>	<b>12.90</b>	<b>10.24</b>	<b>13.01</b>	<b>3.40</b>

not use decaying on this parameter as in the original work [86]. We normalized the input and output LPSs to have zero mean and unit variance. We calculated these statistics from the training data and applied them during inference. We reverted the normalization when we calculate the LSD loss during training since calculating LSD on normalized data does not make sense perceptually.

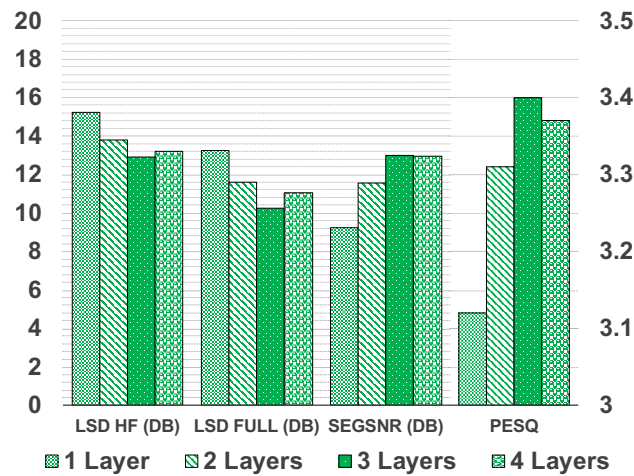
## 2.2.5 Results

Objective evaluation results are shown in Table 2.2. The table shows the high-frequency LSD values (*LSD HF*), full-range frequency LSD values (*LSD Full*), SegSNR values and PESQ values for the baseline methods, our neural network trained with only the LSD loss (denoted as *SSR-LSD*) and that with the full loss *SSR-GAN*. *SSR-GAN* method outperforms the baselines in both 2x and 4x SSR tasks with a good margin in terms of all of the three objective evaluation metrics. The improvement of our method, compared to BL2, is more pronounced in the 4x setting.

Figure 2.21 (a) and (b) show the example spectrograms, where the first row is the ground truth high-frequency range spectrogram, the second row is the high-frequency range spectrograms obtained from the *SSR-LSD*, and the third row shows *SSR-GAN* results, for 2x and 4x, respectively. Note that the LPSs on the second rows are overly smooth. After the GAN training, the resulting LPSs are sharper, containing fine details and usually, more energy. Generating more energy, in addition to generating fine details, leads to slightly better objective measures as seen in Table 2.2. Nevertheless,



(a) 2x



(b) 4x

Figure 2.22: Objective evaluation results are presented for changing the number of layers in the encoder and decoder of the generator network. The results for 2x and 4x scales are shown in (a) and (b), respectively. The four sets of bars show *LSD HF*, *LSD Full*, *SegSNR*, and *PESQ* values, respectively.

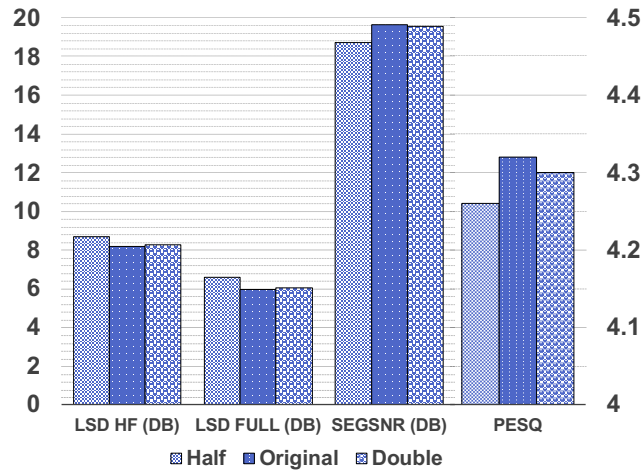
the difference between the objective results for *SSR-LSD* and *SSR-GAN* are somewhat close compared to the baselines, especially for LSD metrics. We believe that the benefit of adversarial training is more evident for the subjective evaluations, which we discuss in Section 2.2.6.

### Architecture and Parameter Analysis

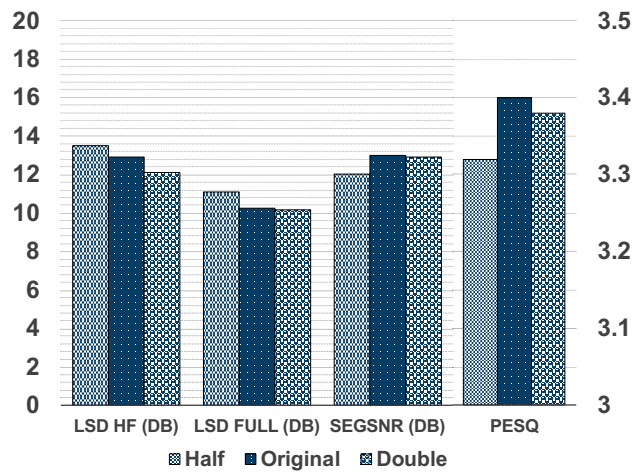
In this section, we analyze our network by changing the number of hidden layers and the number of filters to see how they influence the objective evaluation results.

**Number of Hidden Layers** Our proposed generator contains three encoder layers, followed by a bottleneck layer, three decoder layers, an upsampling layer, and an output layer. Note that the encoder and decoder layers are symmetric. We varied the number of layers in the encoder and decoder and reported the objective evaluation results in Figure 2.22. The results show that the network with three layers generally achieves the best performance across all of the objective metrics, although the differences between the three layers and four layers are rather small for the 2x scale. The network with one or two layers, however, achieves significantly worse performance. We believe that the networks with one or two layers perform worse due to underfitting, i.e., the capacity of these networks is not sufficient to learn patterns in the training corpus. As for the four-layer configuration, the performance slightly drops compared to three layers, which suggests that the increased capacity leads to overfitting. Considering the computational cost and slight performance differences between three layers and four layers, the three-layer configuration is preferred in our experiments.

**Number of Filters** Next, we investigated the effect of varying the number of filters on our generator network. We investigated two other configurations in addition to the original configuration shown in Table 2.1. The first configuration is called *Half*, where the number of filters of the original configuration is halved. The second one is called *Double* and has twice the number of filters of the original configuration. The results are shown in Figure 2.23. The results show that the configuration *Half* performs worse than the original in terms of objective measures, although the difference is not significant. This is a good option for systems with limited resources, where the number of filters can be halved in order to reduce the computational costs. Again, we suspect that the *Half* configuration suffers from underfitting due to the reduced capacity. For the *Double* configuration, increased computational complexity does not translate much into the performance gain compared to the original. Interestingly, for 4x scale, *Double* yields slightly better results for *LSD HF* and *LSD FULL* metrics, but overall, yields slightly lower speech quality. We believe that this is due to overfitting.



(a) 2x

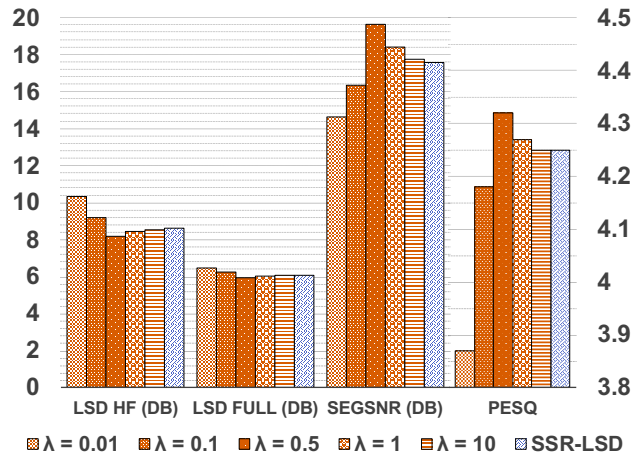


(b) 4x

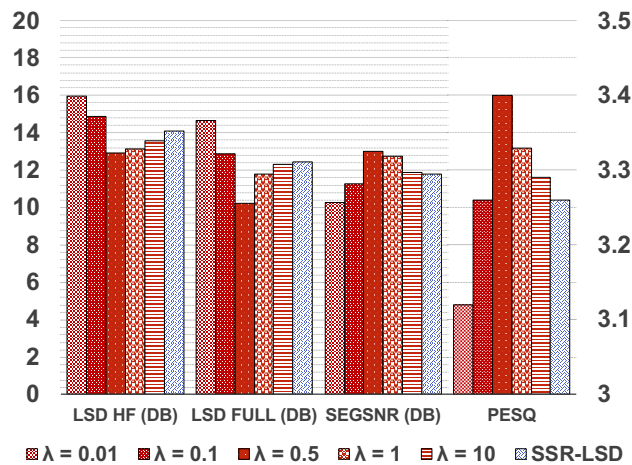
Figure 2.23: Objective evaluation results are presented for changing the number of filters of the generator network. The results for 2x and 4x scales are shown in (a) and (b), respectively. *Half* and *Double* means that the number of filters shown in Table 2.1 has been halved and doubled, respectively. The four sets of bars show *LSD HF*, *LSD Full*, *SegSR*, and *PESQ* values, respectively.

### Loss Weight Parameter ( $\lambda$ )

We analyzed the impact of changing the loss weight parameter  $\lambda$ . Increasing the value of  $\lambda$  increases the weight of the reconstruction loss. In this experiment, we used the following  $\lambda$  values: 0.01, 0.1, 0.5 (default), 1, and 10. The results for 2x and 4x scale experiments are shown in Figure 2.24. As the  $\lambda$  value increases the objective results get closer to the *SSR-LSD* results. On the other hand, decreasing  $\lambda$  from the default value of 0.5 leads to a degradation in generation quality. Since GAN



(a) 2x



(b) 4x

Figure 2.24: Objective evaluation results are presented for different loss weight parameters ( $\lambda$ ) and for *SSR-LSD* for comparison. The results for 2x and 4x scales are shown in (a) and (b), respectively. The four sets of bars show *LSD HF*, *LSD Full*, *SegSNR*, and *PESQ* values, respectively.

loss becomes dominant, the generator produces speech-like spectrogram shapes that are unintelligible. In conclusion, we chose  $\lambda = 0.5$  since it seemed a good balance between generating sharp and intelligible results.

## 2.2.6 Noise Analysis

In real-world applications, the incoming speech signal has a high chance of containing background noise. Therefore, we further analyze our method against unseen time-varying noise types in this



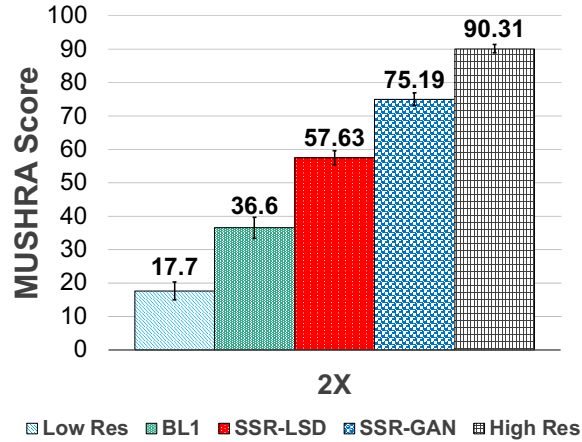
Table 2.3: Objective evaluation results for noise analysis.

Scale	Noise Type	Method	LSD HF (dB)
2x	Babble	SSR-GAN	14.63
		<b>NR-SR-GAN</b>	<b>10.23</b>
	Factory	SSR-GAN	13.47
		<b>NR-SR-GAN</b>	<b>9.97</b>
	Motorcycle	SSR-GAN	14.24
		<b>NR-SR-GAN</b>	<b>10.08</b>
4x	Babble	SSR-GAN	17.35
		<b>NR-SR-GAN</b>	<b>14.12</b>
	Factory	SSR-GAN	16.78
		<b>NR-SR-GAN</b>	<b>13.56</b>
	Motorcycle	SSR-GAN	17.16
		<b>NR-SR-GAN</b>	<b>13.84</b>

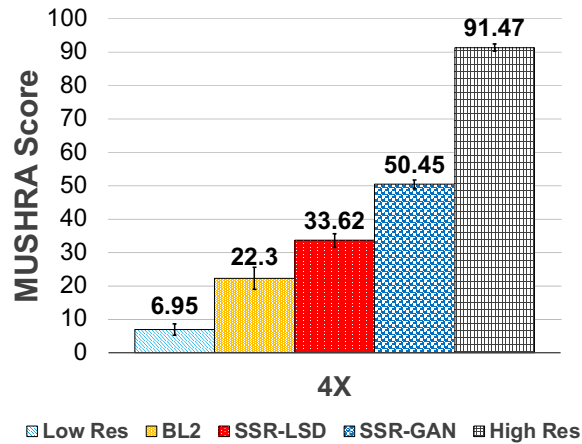
section. We trained our network against noise, by creating a dataset, where the narrowband signal is mixed with noise types in -6, -3, 0, 3, 6 and 9 dB SNR. We call this version of our network noise resilient *SSR-GAN* (*NR-SSR-GAN*). The network tries to predict the clean high-frequency range LPS from corrupted narrowband LPS. We employed the noise data from [115] for training. For evaluation, we used unseen noise types that were not present during training. Specifically, we used babble and factory noises described in [68] and a motorcycle noise described in [69]. We report the high-frequency range LSD results for samples that are mixed with 0 dB signal-to-noise ratio (SNR) testing noises using our base network model (*SSR-GAN*) and *NR-SSR-GAN* in Table 2.3. The results suggest that noise resilient version of *SSR-GAN* can yield better scores against all three test noise types than the original *SSR-GAN*. The most challenging noise type is babble noise, followed by motorcycle noise and lastly, the factory noise.

## Subjective Evaluations

**Perception Test** We conducted subjective evaluations to test if our method is successful regarding human perception. In our evaluations, we used a MUSHRA (Multiple Stimuli with Hidden Reference and Anchor) test [116]. We compiled two test sets, one for 2x scale and one for 4x scale, where each of them contains 10 different tuples of signals with 5 signals in each tuple. These 5 signals included the narrowband signal (anchor), ground-truth high-resolution signal (reference), predicted super-resolution signals of our methods (*SSR-LSD* and *SSR-GAN*), *BL1* for 2x scale, and *BL2* for 4x



(a) 2x



(b) 4x

Figure 2.25: The subjective evaluation results (MUSHRA test) for 2x and 4x scales are shown in (a) and (b), respectively. The error bars show the 95% confidence intervals.

scale. We wanted to limit the test time for each subject within 30 minutes; therefore we only used samples generated from one baseline method for each experiment. Before starting the experiments, each volunteer was trained by listening to 10 pairs of low and ground-truth high-resolution samples that were not contained in the testing tuples. After training, the testing utterances were presented to the volunteers in tuples, and within a tuple, the samples were presented randomly. The volunteers assigned a score between 0 and 100 for each utterance, where 0 corresponds to the low-resolution signal, and 100 corresponds to the high-resolution signal. We recruited 20 volunteers, where each of them evaluated 100 utterances (50 per 2x and 4x scales). During the test, the evaluators could

listen to each utterance as many times as they wanted, and could listen to the reference signal (high-resolution signal) anytime. In MUSHRA experiments, the utterance tuples for which the evaluator failed to identify the hidden reference signal should be excluded. In our experiments, all evaluators successfully identified the hidden reference signal for all tuples.

The 2x scale experimental results are shown in Figure 2.25 (a). The ground-truth high-resolution speech has an average score of 90.31, which is followed by the *SSR-GAN* with an average score of 75.19%. The *SSR-LSD* achieves a score of 57.63%. The low-resolution signal and *BLI* has low scores, which are 17.7% and 36.6%, respectively. A paired t-test shows that the *SSR-GAN* score are statistically better compared to those of *SSR-LSD* and *BLI* at the significance level of 0.01 ( $p = 10e-43$ ).

Figure 2.25(b) shows MUSHRA test results for the 4x scale. The results show that the 4x experiments are more challenging compared to 2x experiments. The gap between the high-resolution score and the *SSR-GAN* is around 41%. *SSR-GAN* can still outperform the baseline method and has slightly more than 50% score. A paired t-test shows that the *SSR-GAN* results are statistically better compared to the *SSR-LSD* and *BL2* results at the significance level of 0.01 ( $p = 10e-36$ ).

Although *SSR-GAN* only slightly outperforms *SSR-LSD* in objective evaluation, their subjective evaluation results show a wider gap and the evaluators clearly preferred *SSR-GAN* over *SSR-LSD*. This outcome confirms the benefit of using the GAN loss for the SSR task.

**Intelligibility Test** To rule out the possibility that the proposed *SSR-GAN* approach generates high-quality speech like sounds that are actually incomprehensible, we further conducted a listening test to check the intelligibility of the generated high-resolution speech. We employed the TIMIT dataset [117] for this test since it is distinct from our training dataset and the transcriptions of the sentences are available. As a baseline, we included the low-resolution samples into this test. We randomly selected 10 utterances with the low-resolution and selected 10 different utterances generated by *SSR-GAN* per 2x and 4x scales, totaling 40 sentences. We employed 20 volunteers among University of Rochester Graduate students, each of which evaluated all 40 sentences. During the experiments, the evaluators were presented each sample twice and were asked to transcribe the words.

Table 2.4: The intelligibility test results. The mean and standard deviation (std) of word error rate (WER) is shown for the 2x and 4x scale experiments using *SSR-GAN*.

Scale	Method	WER mean (%)	WER std (%)
2x	low-res	1.64	1.36
	SSR-GAN	1.48	1.28
4x	low-res	4.27	2.86
	SSR-GAN	3.82	2.12

Table 2.4 shows the mean and standard deviation of the word error rate (WER) between the ground-truth and evaluators’ transcription. The error rates for the 2x scale experiment are 1.48% and 1.64% for *SSR-GAN* and low-resolution signal (8 kHz sampling rate), and for the 4x scale experiment, they are 3.82% and 4.27% for *SSR-GAN* and low-resolution signal (4 kHz sampling rate). The 2x scale experiments have a lower error rate compared to 4x scale experiments since 8 kHz speech signals are more comprehensible than 4 kHz speech signals. Since *SSR-GAN* error rates are slightly lower than the low-resolution signal error rates, it can be concluded that the proposed SSR method does not impair the speech intelligibility.

### Stability of GAN Training

In this study, we have considered different types of GANs and regularization techniques for stabilizing their training processing for SSR. We started from exploring the vanilla GAN [96]. After training it for a few epochs, it became unstable and produced nonsensical results. We observed similar issues for the WGAN [100] and the least-squares GAN [118]. Next, we explored GANs with regularization. WGAN-GP [101] and a GAN with instance noise regularization [102] produced more meaningful (spectrograms that looked like speech) yet not intelligible results. Finally, the regularization method suggested by Roth et al. [86] stabilized the GAN training, and led to the results obtained in this work. The regularizer [86] introduces a term that penalizes the weighted gradient-norm of the discriminator, leading to overcome the phenomenon called mode collapsing effectively. Furthermore, it is a simple modification over the traditional GAN implementation and is computationally efficient compared to other regularization schemes.

Table 2.5: Computational complexity in terms of floating point operations per second (FLOPS), FLOPS per generating 1 second of speech and number of parameters for the baselines (BL1 and BL2) and the proposed SSR-GAN method.

Scale	Method	Number of Parameters	Computational Complexity (FLOPS)	FLOPS per 1 second of speech
2x	BL1 [84]	11.2 M	45.1 M	2.9 B
	BL2 [85]	56.4 M	76.2 B	202.7 B
	SSR-GAN	14.6 M	154.0 M	<b>616.0 M</b>
4x	BL2 [85]	56.4 M	76.2 B	202.7 B
	SSR-GAN	<b>16.0 M</b>	<b>190.5 M</b>	<b>762.0 M</b>

### 2.2.7 Computational Complexity

We compare the computational performance of our method with the baselines using two metrics: floating point operations per second (FLOPS) and the number of trainable parameters. To obtain the FLOPS for each network, we employed Tensorflow’s profiler.

Table 2.5 shows these values for 2x and 4x configurations of our method and the baselines. Please note that for *BL2*, the scale does not influence the computational complexity, since the input is always up-sampled to the target resolution. From values in the 2x scale, it can be observed that the fastest network during run time is *BL1*, followed by our method. It is important to highlight that *BL1* generates a single frame, while *BL2* and our method generate multiple frames. Therefore, we calculated the FLOPS value for generating 1 second of speech for each of these methods and concluded that our method has the lowest complexity.

### Phase Estimation

In this work, we simply flipped the phase of the low-resolution signal as the phase of the high-frequency range of the SSR output. To improve our results, we considered Griffin-Lim algorithm [119] to estimate the phase of the high-range frequencies. However, the results contained artifacts, namely musical noise, and compared to flipped-phase we used in our experiments, they were not satisfactory. We think it is beneficial to share this finding with the research community. In addition, some example samples reconstructed with Griffin-Lim algorithm are shared in the link we provided.

Future research directions to improve our results include estimating the phase using a deep learning approach or directly estimating the raw waveform.

## 2.2.8 Conclusions

We introduced a novel method for speech super-resolution using adversarial training and sequence-to-sequence modeling. To stabilize the GAN training, we employed a regularization method that penalizes the discriminator’s gradient norms. Our generator architecture is a bottleneck encoder-decoder, while our discriminator architecture contains a convolutional decoder followed by fully connected layers. We used 1D kernels in the convolutional layers to reduce the computational complexity. The proposed method was evaluated for 2x (8 kHz to 16 kHz) and 4x (4 kHz to 16 kHz) scale super-resolution. We showed that our method outperforms the two state-of-the-art baseline methods in terms of objective metrics. We also conducted a subjective intelligibility evaluation, which showed that our method can score closely to the ground-truth high-resolution signal for the 2x scale, and can perform decently for the 4x scale. In additional experiments, we introduced a training method to increase the system’s resilience against non-stationary, unseen noise types for real-world applications. Future directions include the estimation of phase information for better super-resolution quality.

## **Chapter-3**

# **Generating Talking Faces From Speech: Shape-Based Methods**

## **3.1 Generating Talking Face Landmarks From Speech**

### **3.1.1 Introduction**

Speech is a natural form of communication, and understanding speech is essential in daily life. The auditory system, however, is not the only sensory system involved in understanding speech. The visual cues from a talker's face and articulators (lips, teeth, tongue) are also important for speech comprehension. Trained professionals are able to understand what is being said by purely looking at lip movements (lip reading) [120]. For ordinary people and the hearing impaired population, the presence of visual signals of speech has been shown to significantly improve speech comprehension, even if the visual signals are synthetic [11]. The benefits of adding the visual speech signals are more pronounced when the acoustic signal is degraded, due to background noise, communication channel distortion, and reverberation.

In many scenarios such as telephony, however, speech communication is still acoustical. The absence of the visual modality can be due to the lack of cameras, the limited bandwidth of communication channels, or privacy concerns. One way to improve speech comprehension in these scenarios is to synthesize a talking face from the acoustic speech in real time at the receiver's side. A key challenge of this approach is to make sure that the generated visual signals, especially the lip movements, well coordinate with the acoustic signals, as otherwise more confusions will be introduced.

In this work, we propose to use a long short-term memory (LSTM) network to generate landmarks of a talking face from acoustic speech. This network is trained on frontal videos of 27 different speakers of the Grid audio-visual corpus [121], with the face landmarks extracted using the Dlib toolkit [122]. The network takes the first- and second-order temporal differences of the log-mel spectra as the input, and outputs the x and y coordinates of 68 landmark points. To help the network capture the audio-visual coordination instead of the variation of face shapes across different people, we transform all training landmarks to those of a mean face across all talkers in the training set. After training, the network is able to generate face landmarks from an unseen utterance of an unseen talker. Objective evaluations of the generation quality are conducted on the LDC Audiovisual Database of Spoken American English dataset [123], which will be referred as the LDC dataset in the remainder of this chapter. Subjective evaluation is also conducted to ask evaluators to distinguish speech videos with ground-truth and generated landmarks. Both the objective and subjective evaluations achieve promising results. The code and pre-trained talking face models are released to the community<sup>1</sup>

### 3.1.2 Related Work

Generating a talking head automatically has been a great interest in the research community. Some researchers focused on text-driven generation [124, 125, 126, 127]. These methods map phonemes to talking face images. Compared to text, voice signals are surface-level signals that are more difficult to parse. Besides, voices of the same text show large variations across speakers, accents, emotions, and the recording environments. On the other hand, speech signals provide richer cues for generating natural talking faces. For text, any plausible face image sequence is sufficient to establish natural communication. For speech, it must be a plausible sequence that matches the speech audio. Therefore, text-driven generation and speech-driven generation are different problems and may require different approaches.

There exist a few approaches to speech-driven talking face generation. Early work in this field mostly used Hidden Markov Models (HMM) to model the correspondence between speech and facial movements [128, 129, 130, 131, 132, 133, 134]. One of the notable early work, Voice Puppetry [128],

---

<sup>1</sup><http://www.ece.rochester.edu/projects/air/projects/talkingface.html>



proposed an HMM-based talking face generation that is driven by only speech signal. In another work, Cosker et al. [130, 131] proposed a hierarchical model that animates sub-areas of the face independently from speech and merges them into a full talking face video. Xie et al. [132] proposed coupled HMMs (cHMMs) to model audio-visual asynchrony. Choi et al. [129] and Terissi et. al [133] used HMM inversion (HMMI) to estimate the visual parameters from speech. Zhang et al. [134] used a DNN to map speech features into HMM states, which further maps to generated faces.

In recent years, a few DNN-based approaches have also been proposed. Suwajanakorn et al. [4] designed an LSTM network to generate photo-realistic talking face videos of a target identity directly from speech. Their system requires several hours of face videos of the specific target identity, which greatly limits its application in many practical scenarios. Chung et al. [20] proposed a convolutional neural network (CNN) system to generate a photo-realistic talking face video from speech and a single face image of the target identity. Compared to [4], the reduction from several hours of face videos to a single face image for learning the target identity is a great advance.

While end-to-end speech-to-face-video generation is very useful in many scenarios, the main limitation of this approach is the lack of freedom for further manipulation of the generated face video. For example, within a generated video, one may want to vary the gestures, facial expressions, and lighting conditions, all of which can be relatively independent of the content of the speech. These end-to-end systems cannot accommodate such manipulations unless they can take these factors as additional inputs. However, that would significantly increase the amount and diversity of data required for training the systems.

A modular design that separates the generation of key parameters and the fine details of generated face images is more flexible for such manipulations. Ideally, the key parameters should just respond to the speech content, while the fine details should incorporate all other non-speech-content related factors. Pham et al. [16] adopted a modular design: the system first maps speech features to 3D deformable shape and rotation parameters using an LSTM network, and then generates a 3D animated face in real-time from the predicted parameters. In [17], they further improved this approach by replacing speech features with raw waveforms as the input and replacing the LSTM network with a convolutional architecture. However, compared to face landmarks used in our proposed approach,

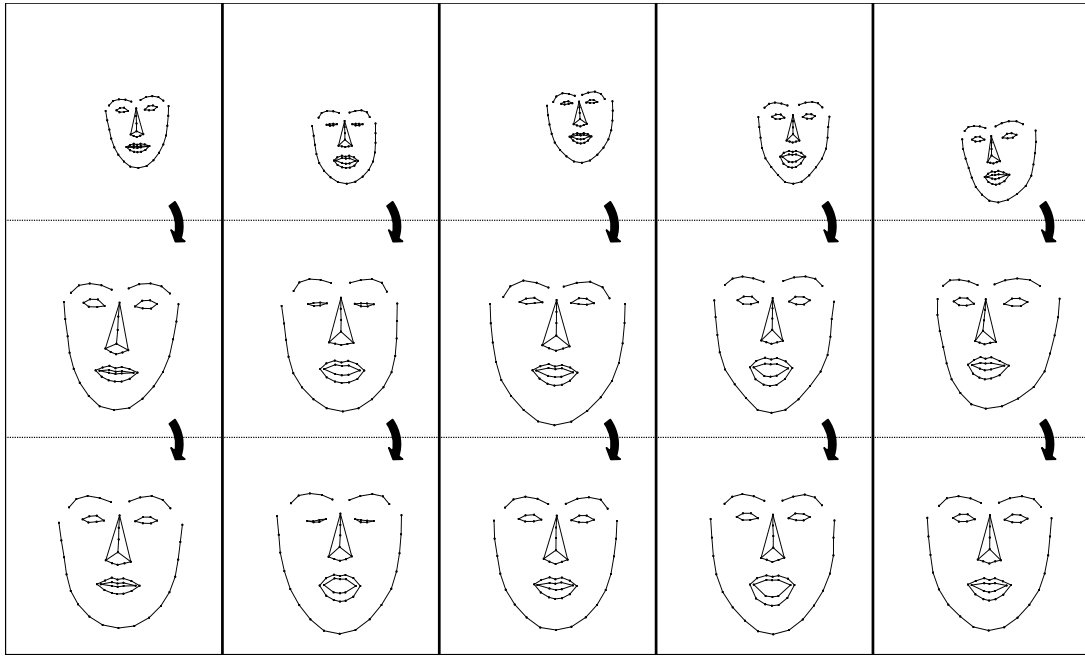


Figure 3.1: Examples of extracted face landmarks from the training talking face videos. Certain landmarks are connected to make the shape of the face easier to recognize. The first row shows unprocessed landmarks of five unique talkers. The second row shows their landmarks after outer-eye-corner alignment. The third row shows their landmarks after alignment and the removal of identity information.

these shape and rotation parameters are less intuitive, and the mapping from these parameters to a certain gesture or facial expression is less clear. In addition, the landmarks generated by our system are for a normalized mean face instead of a certain target identity. This also helps remove factors that are not directly related to the voice.

### 3.1.3 Proposed Method

In this section, we describe our method to generate talking face landmarks. First, we extract face landmarks and align them across different speakers and transform their shapes into the mean shape to remove the identity information. We extract the first and second order temporal difference of the log-mel spectrogram and use them as the input to our system. Finally, we train an LSTM network to generate the face landmarks from the speech features.

## Training Data & Feature Extraction

We employ the audio-visual GRID dataset [121] to train our system. There are in total 16 female and 18 male native English speakers, each of which has 1000 utterances that are 3 seconds long. The sentences are structured to contain a command, a color, a preposition, a letter, a digit, and an adverb, for example, “*set blue at C5 please*”.

The videos are provided in two resolutions, low (360x288) and high (720x576). In this work, we use the high-resolution videos. The videos use a frame rate of 25 frames per second (FPS), resulting in 75 frames for each video. The speech audio signal is extracted from the video with a sampling rate of 44.1 kHz.

We extract 68 face landmark points (x and y coordinates) using the DLIB library [122] from each frame for each video in the dataset. Examples are shown in the first row of Figure 3.1. We calculate 64 bin log-mel spectra of the speech signal covering the entire frequency range using a 40 ms hanning window without any overlap to match the video frame rate. We then calculate the first- and second-order temporal differences of the log-mel spectra and use them as the input (128-d feature sequence) to our network. We experimented using log-mel spectrogram with and without its first- and second-order derivatives as input to our network. The generated mouth for many speech utterances in these two setups, however, were almost always open even in silent segments, and the lip movements were less prominent than the current system. The first- and second-order temporal differences of the log-mel spectrogram may show less variations on the same syllable uttered by different speakers, and the mismatch problem is less pronounced.

## Face Landmark Alignment

Since the talking face may appear in different regions with different sizes in different videos, we need to align them to reduce the complexity of training data. To do so, we follow the procedure described in [135] to simply pin the two outer corners of the eyes in the first frame of each video to two fixed locations, (180, 200) and (420, 200) in the image coordinate system, through an 6 DOF affine transformation. We then transform all of the landmarks in all video frames with the same

transformation. Note that we do not align each video frame using their own affine transformation separately because we find that the eye-corner-based alignment is sensitive to eye blinks, which often results in zoom in/out effects of the transformed face shape. Also note that our approach assumes that the head does not move significantly within a video, as otherwise, the same affine transformation would not be able to align faces in different frames. The second row of Figure 3.1 shows several examples of the aligned face landmarks.

### Removing Identity Information from Landmarks

After alignment, faces of different speakers are of a similar size and general location; however, their shapes are still different as well as their mouth locations. This identity-related variation may pose challenges to the network for capturing the relation between speech and lip movement, especially when the amount and diversity of training data are small. Therefore, we propose to remove the identity information from the landmarks before training the network.

To do so, we apply the following steps. First, we calculate the mean face shape by averaging all aligned landmark locations across the entire training set. Second, for each face landmark sequence, we calculate the affine transform between the mean shape and the first frame of the sequence. Third, we calculate the difference between the current frame and the first frame and multiply with the scaling coefficients obtained from the second step with the result obtained in the third step. Finally, we add the mean shape to results obtained in fourth step to obtain the face landmark sequence that has no identity. The third row of Figure 3.1 shows several examples of landmarks with the identity removed.

### LSTM Network

Our proposed network, as shown in Figure 3.2, uses four long short-term memory (LSTM) [136] layers with a sigmoid activation function. At each time step, the input to the network is the first and second order temporal differences of the log-mel spectra of the current and the previous  $N$  frames. This provides short-term contextual information. The output is the predicted the  $x$  and  $y$  coordinates of face landmarks of the current frame (if no delay is added) or a previous frame (if a delay is added as described below). The reason for adding delay is because lips often move before the sound is

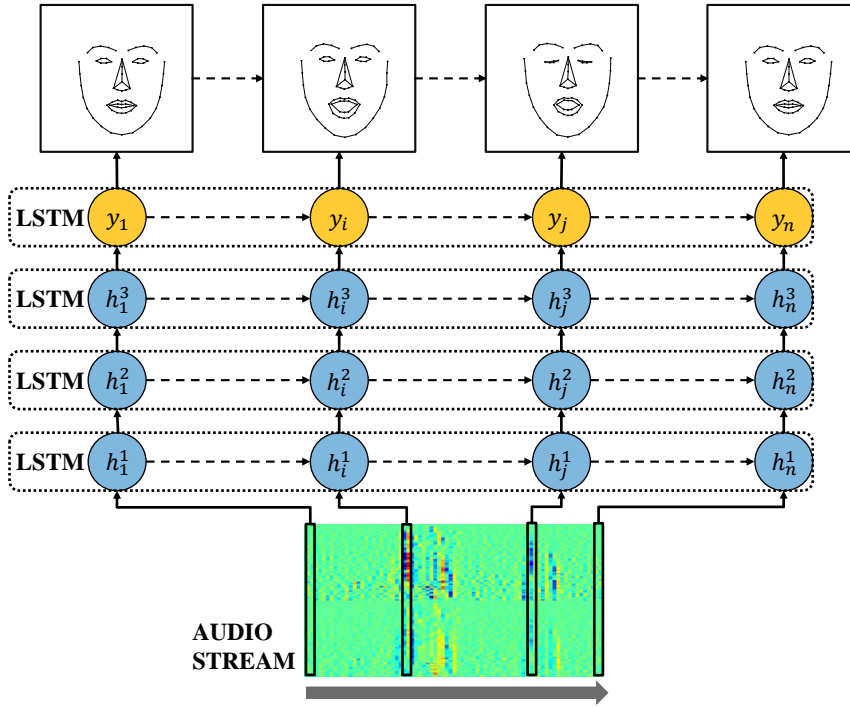


Figure 3.2: The LSTM network architecture for generating landmarks of a talking face from the first and second order temporal differences of the log-mel spectrogram.  $h_t^l$  are the hidden layers, where  $t$  is the time step and  $l$  is the hidden layer index.  $y_t$  are the output face landmarks for the time step  $t$ .

produced. With a little delay, the network is able to “hear into the future” and can better prepare for those lip movements. The generated lip movements tend to be smoother. The amount of delay we introduce is between 1 (40 ms) and 5 frames (200 ms). This turns out to be enough for good generation results and is still tolerable in real-time speech communication.

During training, we use dropout between each layer and between recurrent connections, with a rate of 0.2. We use Adam optimizer to train our network. The training sequences are all 75 frames long. We set the batch size to 128 sequences and the learning rate to 0.001. Our network minimizes the following mean squared error (MSE) objective function  $J_{MSE}$ ,

$$J_{MSE} = \frac{1}{N} \sum_t^N \|\mathbf{s}_t - \hat{\mathbf{s}}_t\|^2, \quad (3.1)$$

where  $\mathbf{s}$  and  $\hat{\mathbf{s}}$  are the x and y coordinates of ground-truth (GT) and predicted (PD) face landmarks sequences, respectively.  $N$  is the number of samples.

Finally, the predicted landmarks are further processed in order to fix the eye corner points to fixed

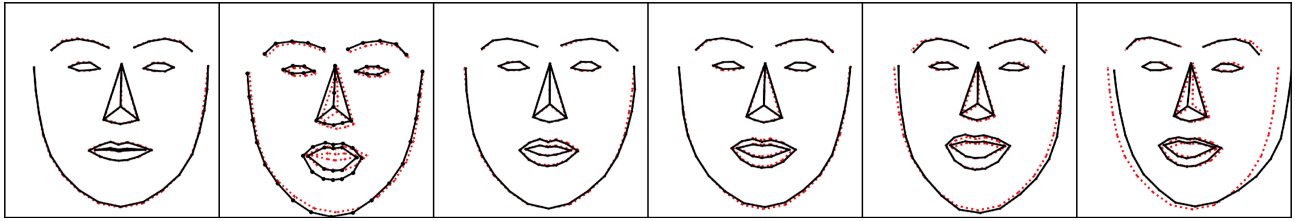


Figure 3.3: Pair-wise comparison between ground-truth landmarks (black solid lines) and generated landmarks (red dotted lines) on unseen talkers and sentences. The second image shows a failure case for “oh” sound.

points as described in Section 3.1.3, which produces more stable talking face landmarks.

Due to causality constraints, the bidirectional LSTM network is not considered in our experiments. We have also experimented with fully connected architecture instead of LSTM. However, the resulting face landmarks often show sudden jumps between frames, which looks unnatural. This is due to not having temporal connections in the architecture.

### 3.1.4 Experiments

We conduct our objective and subjective evaluations on a totally different audio-visual dataset, the LDC dataset [123]. It contains 10 female and 4 male speakers, where each speaker provides 94 samples, totaling to 1316 utterances. The duration of the videos is arbitrary, and the resolution of the samples are 720x480. Since the frame rate of the videos is higher than the Grid dataset used to train our system, we resampled the videos to the same frame rate of 25 FPS. The vocabulary of the LDC dataset is much larger than that of the Grid dataset. There are various words and sentences from TIMIT sentences [137], Northwestern University Auditory Test No. 6 [138], and Central Institute for the Deaf (CID) Everyday Sentences [139]. The audio stream is provided at 48 kHz sampling rate, which we down-sampled to 44.1 kHz. Figure 3.3 shows examples of ground-truth and generated face landmarks in the first and second row, respectively. Examples of generated videos are publicly accessible<sup>2</sup>.

<sup>2</sup><http://www.ece.rochester.edu/projects/air/projects/talkingface.html>

Table 3.1: Objective evaluation results for different system configurations. The models are named according to the amount of delay and contextual information. For example, “D40-C5” describes a model trained with 40 ms delay and 5 frames of context. The lower value means better results, where the ideal result is zero.

	RMSE	RMSE First Diff	RMSE Second Diff
D0-C3	0.0954	0.0045	0.0073
D0-C5	0.0945	0.0042	0.0071
D40-C3	0.0932	0.0039	0.0068
<b>D40-C5</b>	<b>0.0921</b>	<b>0.0032</b>	<b>0.0065</b>
D80-C3	0.0946	0.0044	0.0072
D80-C5	0.0944	0.0043	0.0069

### Objective Evaluation

We report the root-mean-squared error (RMSE) results between the ground-truth (GT) and predicted (PD) face landmarks according to Equation 3.1. The landmarks scale are between 0 and 1, therefore RMSE value of 0.01 approximately equivalent to 1% error. We also report the RMSE of the first and second order temporal differences of the GT and PD face landmarks to assess the movement. We report the results in Table 3.1. These results serve as a way of model selection. The best model according to these results is the model that has 40 ms delay and 5 frames of context information (D40-C5). We selected this model to conduct the subjective evaluations, which are described in the next section.

### Subjective Evaluation

We conducted subjective tests to determine if our system can generate realistic face landmarks. 17 naive volunteer evaluators who are graduate students at the University of Rochester participated in the test. The test presented 25 real landmark videos and 25 generated landmark videos in a randomized order to each evaluator and asked the evaluator to label whether each presented video was real or fake. Each video was presented twice in the randomized video sequence. The real landmark videos were created from randomly selected LDC videos. Landmarks were extracted and aligned, and the identity information was removed, according to Section 3.1.3. Fake videos were generated from the audio signals of another 25 randomly selected LDC videos. The GT landmarks were noisy; hence

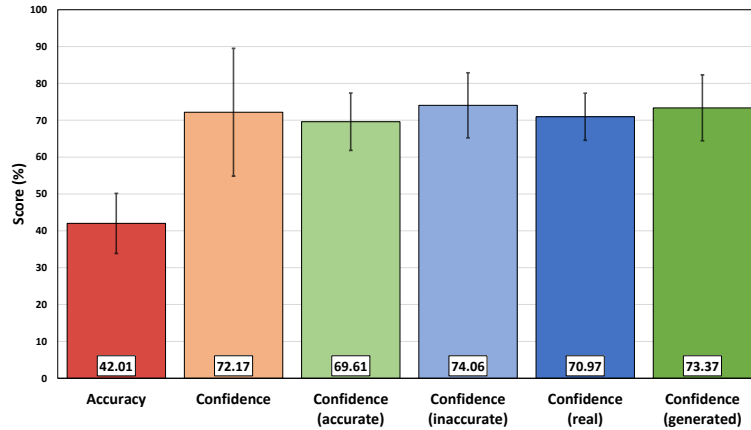


Figure 3.4: Subjective evaluation results. The mean accuracy score and its standard deviation are averaged over all subjects. The mean confidence scores and their standard deviations are averaged over all subjects and videos.

we also added Gaussian noise to the PD landmarks to make them look more like the GT landmarks. In addition to a binary decision, the evaluators were asked to report their confidence level of each decision, between 0 and 100 percent.

The mean accuracy score of the evaluators are shown in Figure 3.4, along with the overall mean confidence score and the mean confidence score for the correctly and incorrectly predicted samples. The results show that the evaluators struggled to distinguish real and generated samples, as the accuracy is 42.01% which is even below chance (50%). Another interesting observation of this test is that the mean confidence score for accurately determined samples is lower than that for inaccurately determined samples. This suggests that the evaluators had a higher classification accuracy when they were more cautious. Another outcome is that the mean confidence score on answers for generated samples is more than the confidence score on answers for the ground truth samples.

### 3.1.5 Conclusions

In this work, we present a method to generate talking face landmarks from speech. We extract face landmarks from the Grid corpus, align them across different speakers, and transform their shapes into the mean shape to remove the identity information. The LSTM network predicts the face landmarks from the first and second order temporal differences of the log-mel spectrogram from any arbitrary voice. The network can produce face landmarks that look natural for the given speech input. The



main limitation of this network is that it cannot produce “oh” and “oo” sounds correctly. We plan to balance the phonetic content of the dataset to enable the network to produce all phonemes correctly in our future work. We will evaluate the system against noise, and improve it to obtain a noise-resilient system in our future work. We report objective and subjective evaluation results that are promising. We release the code and example videos to the community.

## 3.2 Noise-resilient Training Method For Face Landmarks Generation From Speech

### 3.2.1 Introduction

Speech communication between humans is often not merely via the acoustic channel; visual cues can also play an important and even critical role. Extensive studies have shown that seeing lip movements besides hearing speech can significantly improve speech comprehension for both the general and hearing impaired population [8, 9, 10, 11], especially when background noise or compression effects corrupt the acoustic signal.

Therefore, having ways to generate talking faces from acoustic speech signals would significantly improve speech communication and comprehension in many scenarios and enable many applications. It improves access to abundantly available speech content on the web for the hearing impaired population. It is also useful in AR/VR professional training applications for pilots, drivers, machine operators, doctors, police officers, and soldiers, where the training scenarios are often noisy, and audio-only speech comprehension can be challenging. It is also useful for developing visual dubbing applications for movies.

To this end, researchers proposed end-to-end and module-based systems. End-to-end data-driven methods can learn the mapping between speech and visual cues; as a result, they can generate natural looking talking faces [4, 20, 17]. However, utilizing separate modules to generate the key parameters (articulation, mouth shapes) and fine details (texture, identity) has benefits. The key parameters, such as face landmarks, are driven by the speech content directly, and they play the skeleton role

in such systems. Another module can further process the generated face landmarks to impose photo-realistic textures and details of the face. This modular design provides more flexibility than end-to-end generation systems. For example, the face landmarks can be manipulated before being processed by the texture module to change the facial expression, emotion and the fine articulation of words.

Speech signals encountered in the wild often contain background noise that degrades the performance of automatic speech processing systems. It is vital that the talking face generation system is resilient to such background noise in practice. To our best knowledge, however, most of the existing systems do not consider background noise in their system design and evaluation.

In this work, we improve on our previous work [5] and present a new method that generates 3D face landmarks directly from the raw waveform. We propose a novel pre-processing method to normalize the identities of the face landmarks. In addition, we propose a neural network that processes the waveform with convolutional layers with 1D filters and predicts the active shape model (ASM) parameters of 3D face landmarks with a following fully connected (FC) layer. We train the network with pairs of speech audio and 3D face landmarks extracted from the GRID dataset [121]. To cope with background noise in speech input, we further propose a noise-resilient training method that uses speech enhancement ideas in feature learning. Objective evaluations show that our proposed method yields better results than two state-of-the-art baseline methods. Results also show significant improvement thanks to the noise-resilient training method in non-stationary noise conditions. Through subjective evaluations, we show that the generated 3D face landmarks demonstrate a convincing match with the speech audio signals. To promote scientific reproducibility, we release several generation examples, code of the proposed system, and pre-trained models<sup>3</sup>.

Compared to our preliminary work [5], we make the following contributions in this work: 1) We generate 3D face landmarks as opposed to 2D as our previous work. Including the 3rd dimension allows novel applications such as AR/VR, video games and movie dubbing. 2) Instead of Mel-Frequency Cepstral Coefficients (MFCC) and their temporal derivatives, we directly input the raw waveform to the network. 3) We propose a new network architecture that replaces Long Short-Term Memory (LSTM) layers with convolutional layers for improving the results on raw waveform inputs.

---

<sup>3</sup><http://www.ece.rochester.edu/projects/air/projects/3Dtalkingface.html>

4) We propose a noise-resilient training method to incorporate speech enhancement ideas at the feature level to increase the system’s robustness to non-stationary background noise. This noise-resilient training method can be applied to other speech processing tasks such as automatic speech recognition, emotion recognition, and speaker identification/verification.

### 3.2.2 Related Work

The multi-modal approaches can be divided into two categories, approaches that utilize multi-modal inputs to boost their performances, and approaches that convert one modality into another one. The research community showed that using visual modality in addition to the audio modality can significantly improve the performance of the traditional problems such as speech enhancement, source separation, speech recognition, emotion recognition, and voice activity detection [140, 141, 142, 143, 144, 145]. Conversion between the text, audio, and visual modalities has been extensively studied over the years [146, 147, 148]. In the following, we describe the works that generate visual signals from speech.

Generating talking faces from speech has drawn attention from researchers in recent years. There are *shape model-oriented* methods and *image-oriented* methods. Shape model-oriented methods usually employ a deformable face shape model, where the face shape is represented by sparse points in a 2D or 3D space. These models can be controlled by low dimensional parameters that are often obtained by principal component analysis (PCA) or other dimensionality reduction methods. Image-oriented models predict the RGB face or mouth image sequences directly from speech. Some of these methods use intermediate representations as constraints, such as the face or mouth landmarks.

Some works generate talking faces from the text [124, 125, 127, 149, 150, 126]. There is a key difference between text-driven and speech-driven talking faces. Speech signals show large variations across speakers, emotions, and accents for the same text, and the generated talking face must be in sync with the input speech. However, for text-driven faces, any plausible talking face is sufficient. These two tasks require different approaches. Therefore, in the following, we only review speech-driven talking face generation methods.

## Image-Oriented Methods

Suwajanakorn et al. [4] demonstrated an LSTM-based system on synthesizing videos of President Barack Obama from his speech. This system uses a two-stage approach. It first uses an LSTM to predict PCA coefficients of 18 mouth landmark points from 13 MFCCs plus the energy term. Then, according to the predicted PCA coefficients, few nearest candidate frames are selected from the dataset that contains the images of the target identity, and the weighted median is applied to synthesize the texture. Therefore, this method is a hybrid shape-image model since it predicts mouth landmarks first. Although the results are photo-realistic and impressive, the method requires a large amount of training data for the target identity. It can accept speech from a different person; however, it can only generate the face of the person in the training data. It is also computationally heavy, making it difficult to run on edge devices.

Chung et al. [20] proposed a method that accepts 12 MFCCs and a single frame target image to generate a talking face video. The system uses an audio encoder and an identity encoder to convert audio features and the target image to their respective embeddings. It then uses an image decoder to generate face images from these embeddings. Since the generated images are blurry, the system utilizes a separate deblurring module to sharpen the images. All modules are based on 2D convolutional neural networks. This method can run in real-time on a GPU. Similar to this work, Chen et al. [21] proposed a method that leverages an adversarial loss function in addition to a pixel-level reconstruction loss and a perceptual loss, to generate sequences of images from speech. The network accepts the speech and a target lip image as inputs and outputs 16 frames of lip images that are synchronized with the speech. The network contains an audio encoder, an identity encoder, and 3D convolutional residual blocks. Compared to Chung et al.'s method, the generated images are sharper, and a deblurring module is not needed.

A disadvantage of these systems is that facial expressions, animations, and for some systems, the identity information, are difficult to manipulate during generation. The shape model-oriented methods usually predict an intermediate representation that can be manipulated before rendering the details.

### Shape Model-Oriented Methods

Early works focused on Hidden Markov Models (HMMs) to map from speech to talking faces [128, 129, 130, 131, 132, 133, 134]. Voice puppetry [128] was one of the early works. It models 26 points of a face using HMM and drove them using linear predictive coding and relative spectral transform - perceptual linear prediction audio features. Choi et al. [129] used HMM inversion (HMMI) to estimate visual parameters from 12 MFCCs of speech, where the visual parameters include the left and right corners of the mouth and the heights of the upper and lower lips.

Cosker et al. [130, 131] employed a hierarchical model that models subareas of the face independently by an active appearance model (AAM) [151] and then merges them into a full face containing a total of 82 landmark points. Each sub-area is driven by 12 MFCCs of speech. Xie et al. [132] proposed a system that generates only the mouth region using coupled HMMs (cHMMs) to compensate audio-visual asynchrony. They used MFCCs and their first- and second-order derivatives as speech features and PCA coefficients of the mouth region as the visual parameters. Zhang et al. [134] also used PCA coefficients of the mouth region as the visual parameters, but estimated HMM states from speech features with a deep neural network (DNN).

Recent works use deep neural networks to map speech features to face landmarks. Pham et al. [16] proposed an LSTM network that predicts the 3D face model parameters from speech input features, namely MFCCs and the chromagram. The authors later improved their work by using spectrograms as input and employing convolutional and recurrent layers [17] in the network architecture. Karras et al. [18] employed a network that maps speech into 3D positions of 5022 landmark points. The network can generate realistic faces with emotions using only 5 minutes of training data. However, their system is designed for the generation of a single speaker.

Compared to these listed works, our approach includes a novel pre-processing method that normalizes the identities of the target data (face landmarks). This normalization improves the quality of results, leads to faster convergence for neural network training and can work using fairly simple network architectures. In addition, our approach uses a noise-resilient training mechanism to ensure its robustness in noisy conditions. To our best knowledge, this is the first consideration of back-

ground noise in the system design and evaluation of talking face generation from speech. Furthermore, compared to shape model-oriented methods described above, our system predicts landmarks by the multi-pie 68 point markup convention [152], which is used by most of the existing systems for facial landmark detection, face morphing, and face swapping applications. This allows our system to be seamlessly integrated into a pipeline for facial manipulation with such systems. A quantitative side-by-side comparison with the most closely related methods, however, is difficult. Karras et al. [18] and Pham et al. [16] are the two most similar methods to ours, but their systems are optimized for different face models making a side-by-side comparison difficult with ours. In particular, Karras et al. used facial motion capture to obtain a 3D mesh model of the face. Pham et al. used a 3D mesh model of the face built from a Kinect point cloud, and they developed a technique to map videos into this 3D mesh model. We do not have access to their code of these face models, and we believe that it is not fair to those methods to re-implement them but using our 68 point face model to compare with ours.

Considering the above-mentioned difficulties, we eventually chose to compare with the landmark generation part of the system proposed by Suwajanakorn et al. [4] and our prior preliminary system [5]. The system in [4] is a state-of-the-art image-oriented system that generates realistic face images of a single speaker. As an intermediate step, it also predicts PCA coefficients of mouth landmarks similar to our method. Therefore, we believe that it is a reasonable baseline for our method.

### 3.2.3 Method

In this section, we describe face landmark extraction, landmark pre-processing before training the neural network, the proposed neural network architecture, the proposed method to increase the system’s resilience against background noise, and how it works during the inference process.

#### Pre-processing

**Face Landmark Extraction** We first use the open source library DLIB [122] to extract 2D face landmarks ( $x$  and  $y$  coordinates), and then use the method described in [153] to estimate 3D face

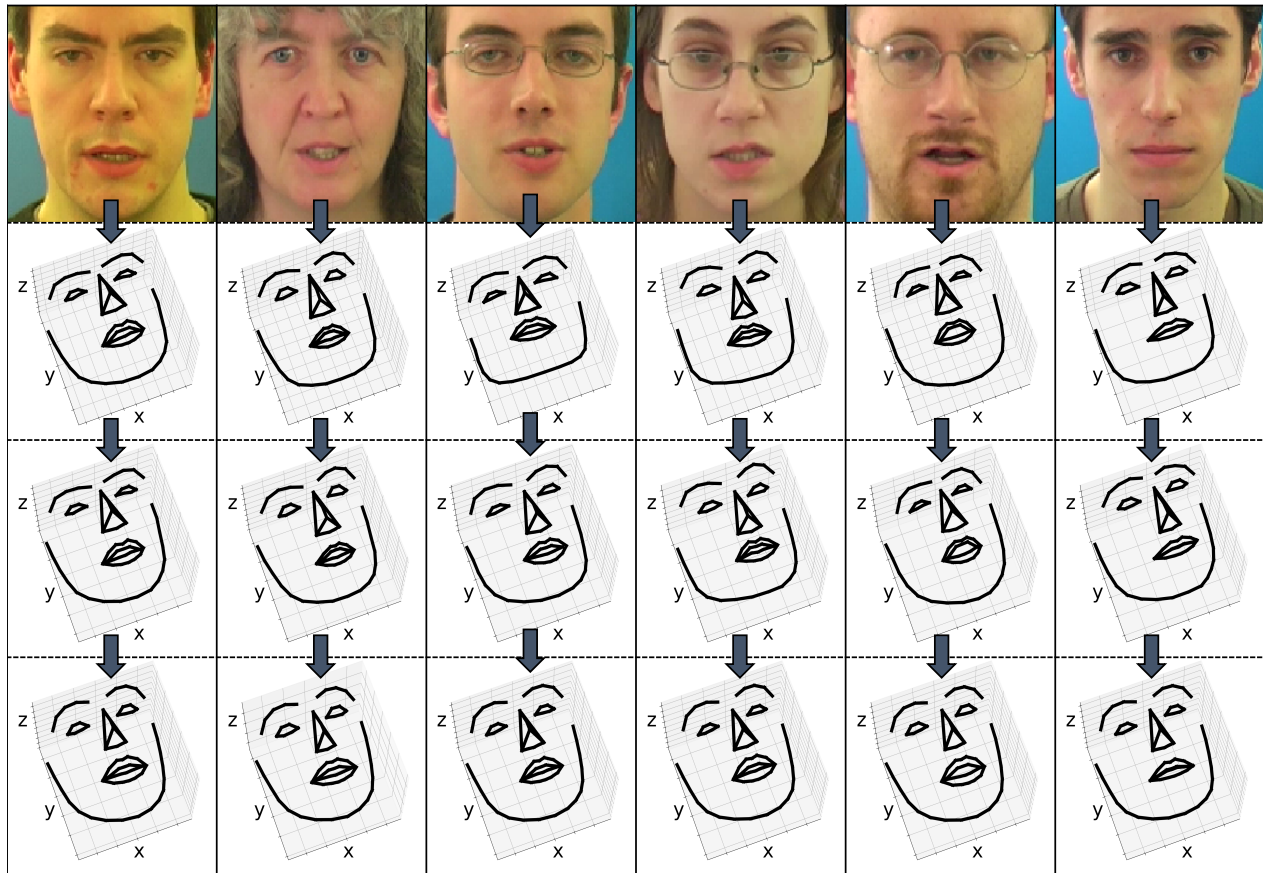


Figure 3.5: Data preparation steps for face landmarks illustrated on six different speakers, where each column corresponds to a speaker. We draw lines between certain landmarks to form face shapes. The first, second, and third rows show raw face landmarks extracted from video images, landmarks after Procrustes alignment, and landmarks after identity removal, respectively.

landmarks from these 2D landmarks and their corresponding video frames. We extract a total of 68 landmarks, following a standard in the mark-up convention described in [152]. Face shapes formed by connecting these landmarks are shown in the first row of Figure 3.5.

**Face Landmark Alignment** The extracted raw landmarks are in pixel coordinates and can be at different positions, scales and orientations. These variations make it difficult to train our neural network, as they are largely irrelevant to the input speech. To minimize these variations, we use Procrustes analysis [154] to align the 3D landmarks. This is a common practice for creating active shape models (ASMs) [155] and active appearance models (AAMs) [151, 156]. Face shapes after alignment are shown in the second row of Figure 3.5.

**Face Landmark Identity Removal** Different speakers have different face shapes, where mouth, nose, and eyes may not be well aligned across speakers even after the Procrustes analysis. These variations are also less correlated to the input speech. Therefore, we want to remove this *identity variation* from our 3D face landmarks. To achieve that, for each landmark sequence, we detect one reference frame that contains a closed mouth by thresholding the distance between the upper lip and lower lip coordinates. We then calculate the landmark coordinate deviations from this reference frame for each frame in the sequence, and impose these deviations onto a template face across all sequences of all identities. This template face is calculated as the average of aligned faces with a closed mouth across all identities. The 3D face landmarks can be represented as:

$$\mathbf{s} = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_N, y_N, z_N)^T, \quad (3.2)$$

where  $N$  is the number of vertices and  $T$  denotes vector transpose. The identity removal operation can be described as:

$$\mathbf{s}_{IR} = \mathbf{s} - \mathbf{s}_{CM} + \mathbf{s}_T, \quad (3.3)$$

where  $\mathbf{s}_{IR}$  represents the identity removed face shape,  $\mathbf{s}_{CM}$  is a face frame with mouth closed that is automatically selected from the video, and  $\mathbf{s}_T$  is the template (reference) shape. Face shapes after identity removal are shown in the third row of Figure 3.5.

**Active Shape Model (ASM)** ASMs [155] are deformable shape models that can represent the variations in the training set by a set of coefficients. These coefficients are the weights for eigenvectors that are obtained by PCA. By using the parameters obtained from PCA,  $\mathbf{s}$  in Equation 3.3 can be described as follows:

$$\mathbf{s} = \mathbf{s}_\mu + \mathbf{w}\mathbf{S}, \quad (3.4)$$

where  $\mathbf{s}_\mu$  is the mean shape vector,  $\mathbf{w} = [c_1, \dots, c_P]$  is a vector that contains the weights and  $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_P]$  is a matrix that contains the eigenvectors.  $P$  is the number of PCA components, which is smaller than their dimensionality ( $P < N$ ).



We create pairs of raw speech waveform and corresponding ASM weights  $w$  as the input-output pairs for neural network training.

**Data Augmentation** By removing the target identity from the 3D face landmarks, we already standardized the target data, which is described in Section 3.2.3. We do not further augment the target data.

For the speech input, we aim to develop a system that is robust against unseen speakers. However, speech recordings encountered in the wild contains articulation, pitch, timbre and talking speed variations. In addition, the recording conditions can be disjoint from that of the training set, which affects the performance of our network.

We resort to data augmentation to improve the robustness of the system. Augmentation is not performed before but rather during training iterations. For each sample in each training batch, we randomly choose whether we use the original training sample or an augmented sample. If it is the latter, two augmentation steps are applied in a sequence. We first pitch shift the sample by one or two semitones up or down. We then apply a gain factor to the amplitude of the sample between -12 dB and 6 dB with a 3 dB granularity. It is noted that this dynamic augmentation is random, but it saves memory compared to a preset augmentation beforehand.

### Network Architecture

The deep neural network (DNN) accepts a frame (280 ms) of the raw waveform as an input and outputs the ASM weights of that frame. There are four convolutional layers with 1D filter kernels operating on the raw waveform. The number of filters grows as the time dimension shrinks. We use strides for each convolutional layer, which halves the time-steps. Each convolutional layer is followed by LeakyReLU activation with a slope of 0.3 and a dropout layer that discards 20% of the units. The final layer is a fully connected layer that outputs the ASM weights. The network architecture is shown in Table 3.2 and Figure 3.6.

In order to have smooth transitions between generated talking faces across frames, we further added a temporal constraint to the network architecture. It accepts the previous frame's ASM weights

Table 3.2: Detailed parameters of the proposed network architecture. The number of filters and hidden units, filter sizes, strides, activations, and output shapes are shown for each layer. *ID\_CNN\_TC* is identical to *ID\_CNN*; further, it accepts condition input and concatenates it with the output of the fully connected (FC) layer that is shown in the last two rows of the table. This concatenated tensor is fed to another FC layer that outputs the final ASM weights.

Net	Layers	Number of Filters or Hidden Units	Filter Size	Strides	Activation	Output Shape
1D_CNN	Input	-	-	-	-	(2240, 1)
	Conv	64	(21, 1)	(2, 1)	LeakyReLU	(1110, 64)
	Conv	128	(21, 1)	(2, 1)	LeakyReLU	(545, 128)
	Conv	256	(21, 1)	(2, 1)	LeakyReLU	(263, 256)
	Conv	512	(21, 1)	(2, 1)	LeakyReLU	(122, 512)
	FC	6	-	-	LeakyReLU	(6)
1D_CNN_TC	Condition	-	-	-	-	(6)
	FC	6	-	-	LeakyReLU	(6)

as a condition in order to obtain smoother results over time. The condition is concatenated to the intermediate tensor immediately after the fully connected layer, and we add another fully connected layer as shown in shown in Table 3.2. We discuss the trade-off between these two models in Section 3.2.4. We denote our proposed method as *ID\_CNN* and the temporally constrained version as *ID\_CNN\_TC* throughout the rest of this paper.

The network minimizes the L1 loss between the predicted and ground-truth ASM weights, as follows:

$$J_{\ell_1}(\mathbf{w}, \hat{\mathbf{w}}) = \|\mathbf{w} - \hat{\mathbf{w}}\|_1, \quad (3.5)$$

where  $\hat{\mathbf{w}}$  is the ASM weight vector predicted by the network. Equation 3.5 shows the loss for a single sample. During training, the average of all training samples is minimized.

### Noise-Resilient Training

To make the system robust to noise, we propose a novel, yet simple method for noise-resilient training. The idea is to match the intermediate features obtained from the clean and noisy speech, as in theory, they contain the same speech information hence the extracted features are ideally be the same. This is shown in Figure 3.7. The clean features  $h$  is obtained by feeding the clean speech  $x$  to the network. The corrupted features  $\tilde{h}$  is obtained by feeding the corrupted speech  $\tilde{x}$  to the same

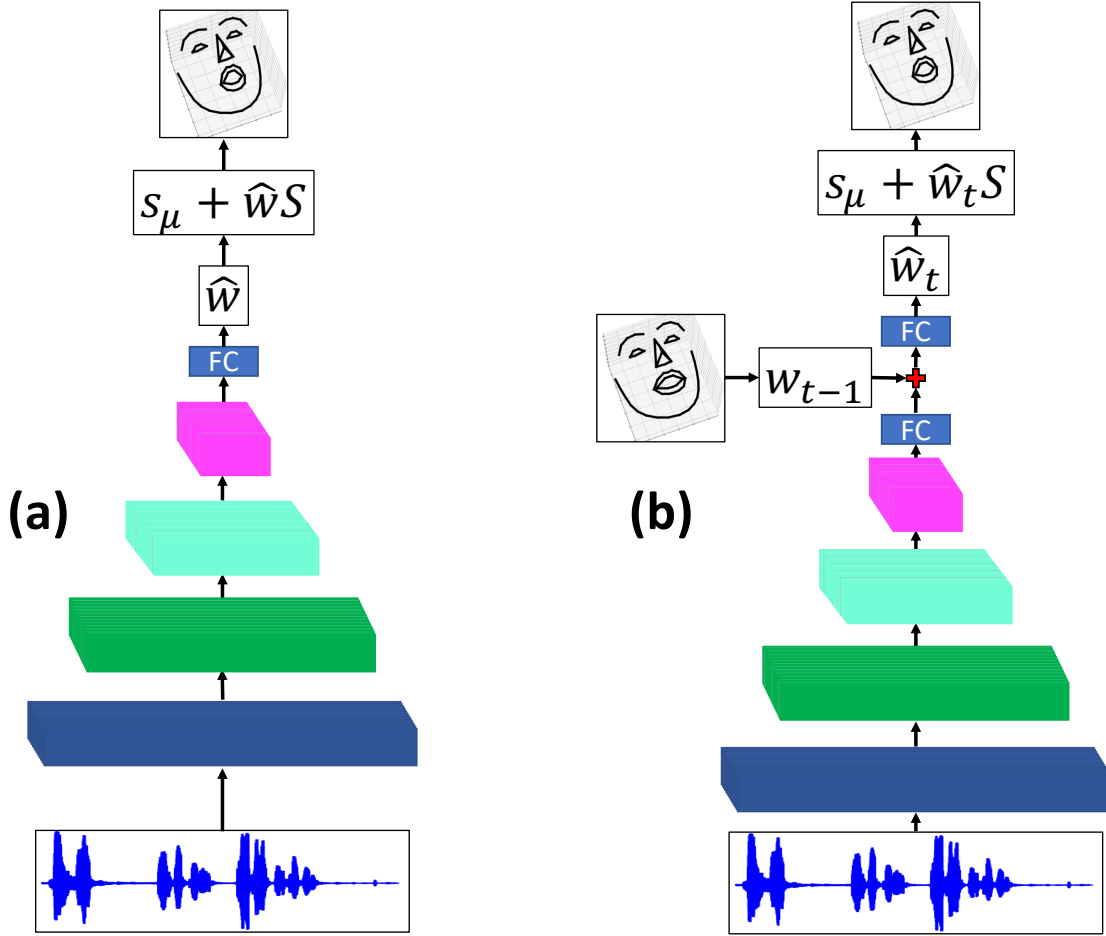


Figure 3.6: The network architecture for (a) *ID\_CNN* network and (b) *ID\_CNN\_TC* network. *ID\_CNN\_TC* is identical to *ID\_CNN*, except that it accepts the previous frame’s ASM weights as a condition to enforce temporal constraint. Raw waveform is fed to four convolutional layers, followed by a fully connected (FC) layer.

network. In addition to the ASM coefficient loss on both networks, we also add the weighted MSE between  $h$  and  $\tilde{h}$ :

$$J = J_{\ell_1}(\mathbf{w}, \hat{\mathbf{w}}) + J_{\ell_1}(\mathbf{w}, \hat{\tilde{\mathbf{w}}}) + \lambda \|\mathbf{h} - \tilde{\mathbf{h}}\|_2, \quad (3.6)$$

where  $\lambda$  is the weighting coefficient, and  $\hat{\tilde{\mathbf{w}}}$  is the ASM parameters generated from corrupted speech  $\tilde{x}$ .

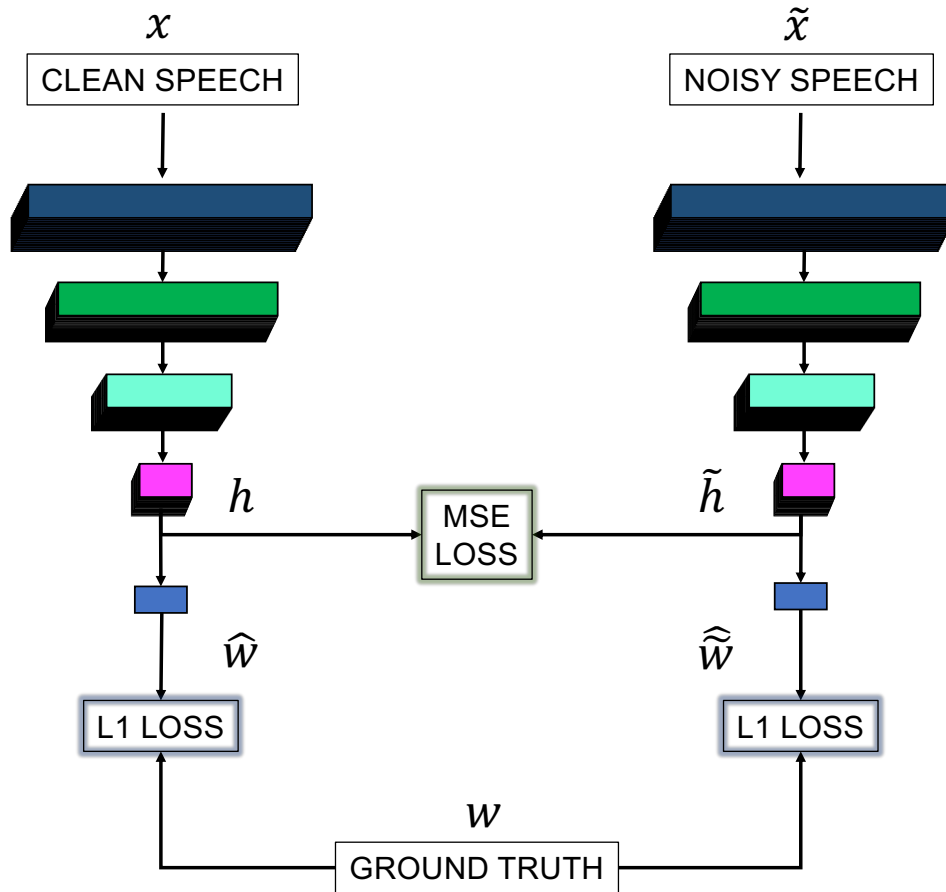


Figure 3.7: The noise-resilient training scheme. The networks on the left and right sides are the same, and their weights are shared. The clean and noisy speech goes through the left and the right networks, respectively, to reconstruct their face landmarks. A mean-squared error (MSE) constraint is applied to the latent representations to incorporate the supervised speech enhancement idea at the feature level.

### System Overview

During inference, our system utilizes a speech buffer that acts as first in first out (FIFO) queue. First, the speech buffer is initialized with zeros. When the system receives new speech data, it is pushed to the speech buffer, and the network predicts the next frame's weights. There is no pre-processing applied to the speech; the raw speech is directly fed to the neural network. The predicted weights are converted to 3D landmark points using Equation 3.4. The system overview is shown in Figure 3.8.

For the  $ID\_CNN\_TC$  network, the system utilizes another buffer, called the conditioning buffer,

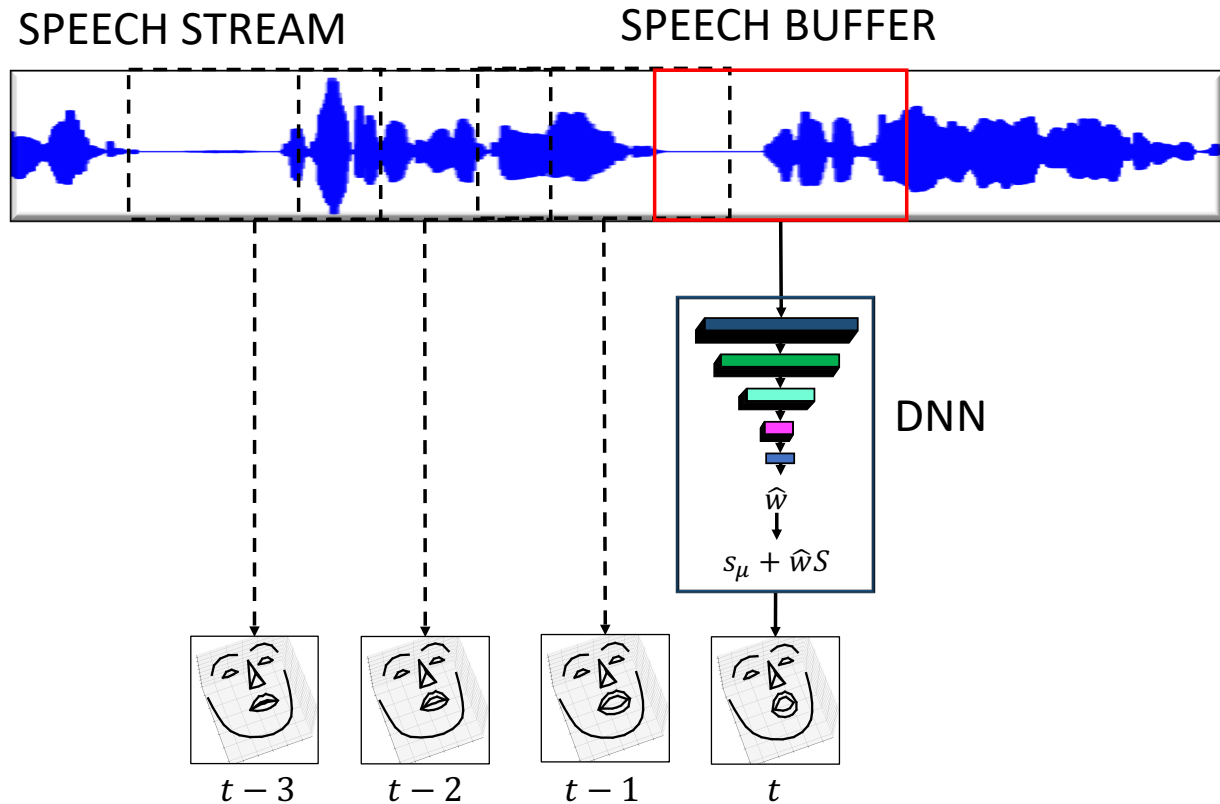


Figure 3.8: System overview. A talking face is generated every 40 ms (frame hop size) from 320 ms (frame length) of audio.  $t$  represents the time.

that stores the last frame. The conditioning buffer is initialized with the template face shape weights.

### 3.2.4 Experiments

#### Datasets

We start our experiments in a single-speaker setting. To this end, we follow Suwajanakorn et al. [4] and utilize President Obama’s weekly address videos, which are available online<sup>4</sup>. We downloaded 315 videos that have 3 minutes average duration, totaling to approximately 18 hours of content. The videos are provided in 30 frames per second (FPS) and we down-sampled the videos to 25 FPS. We split the dataset into training (70%), validation (15%), and testing (15%) sets.

<sup>4</sup><https://obamawhitehouse.archives.gov/briefing-room/weekly-address>

For multi-speaker experiments, we use a publicly available audio-visual dataset called GRID [121] to train our system. There are 34 native English speakers in this dataset, with 16 female and 18 male speakers, who are ranging from 18 to 49 years old. All of the speakers are from England except one from Scotland and one from Jamaica. Each speaker has 1000 recordings that are 3 seconds in duration. The recordings contain sentences that are identical for each speaker. The structure of the sentences is in the following form: *command (4) - color (4) - preposition (4) - letter (25) - digit (10) - adverb (4)*, where the numbers of choices are shown in parenthesis for each component. An example sentence can be given as “*set blue at C 5 please*”.

Recordings are provided both in audio and video format. In this study, we use the high-resolution videos included in the GRID dataset. These videos have a frame rate of 25 FPS and a resolution of  $720 \times 576$  pixels. Since each recording is 3 seconds in duration, each video has a total of 75 frames. The video files contain the corresponding audio that has a sampling rate of 44.1 kHz. We down-sample the audio to 8 kHz which is a typical sampling rate for speech signals in telecommunication.

We employ another dataset that is disjoint from the GRID dataset in order to evaluate our system against unseen speakers, namely Speech Test Video Corpus (STEVI) [157]. Specifically, we employ the *High-Probability speech perception in noise (SPIN) Sentences* and *Nonsense Sentences* listed in [157]. The videos are provided in 29.97 FPS and  $1920 \times 1080$  resolution. The audio stream has a sampling rate of 48 kHz. We down-sample the audio to 8 kHz and generate 3D talking faces. Since our system is trained to generate 25 FPS videos, we use cubic spline interpolation to up-sample the generated videos to 29.97 FPS to match with the ground truth face landmarks. There are a total of 4 speakers, each of which has 400 sentences, 200 *High-Probability SPIN Sentences* and 200 *Nonsense Sentences*. The duration of each sentence is around 2 to 3 seconds.

We use DLIB [122] and [153] to extract face landmarks from these videos according to Section ?? for training, validation and testing. To verify the validity of the extracted face landmarks, we employ a two-step approach. First, we run a script that automatically identifies wrong landmarks by comparing the upper and lower lip landmark positions and eliminates invalid landmark sequences. This script was applied to all extracted landmarks. In the second step, we manually check the landmarks and eliminate problematic sequences. Since manual verification is costly, the second step is only applied

to the STEVI dataset and the test set of the Obama dataset. In this way, we further improve the quality and validity of the evaluation data.

To create noisy speech input, we employ a noise dataset named Sound Ideas [65] that contains 138 different noise types including non-stationary noises from various environments such as nature, city, domestic, office, traffic, and industry. A noisy speech is created by mixing a clean speech file with a randomly selected noise file in 6 to 30 dB SNRs with 3 dB increments.

### Implementation Details

Our system was trained to generate 25 FPS videos, i.e., the system generates a talking face every 40 ms. We include the context information to our input speech. Specifically, we concatenate 3 frames from past and future, totaling 7 frames. For 8 kHz speech signals, a 40 ms window contains 320 data points. The input speech size becomes  $7 \times 320 = 2240$  as shown in Table ???. The networks were trained for 100 epochs, and the weights were saved only if the validation loss was improved for each epoch. We implemented our method in PyTorch [158]. The mini-batch size and learning rate were set to 128 and  $10^{-4}$ , respectively. We used Adam [113] optimizer during training.

We compared our method with Suwajanakorn et al.’s [4] landmark generation method denoted as *BL1* and our preliminary work [5] denoted as *BL2*. *BL1* utilizes a single LSTM layer with a time delay to generate 20 PCA coefficients for the mouth landmarks. The input of their network is the 13 MFCCs plus the log mean energy and their first temporal derivatives. *BL2* accepts first and second derivatives of 13 MFCCs of speech as input and outputs PCA coefficients for the whole face landmarks. There are 4 LSTM layers in the network architecture.

For single-speaker experiments, we trained all of the above-mentioned methods on the Obama dataset, while for multi-speaker experiments, we trained them on GRID and evaluated them on STEVI.

### Objective Evaluation

We used the root-mean-squared error (RMSE) between the ground-truth and predicted face landmark sequences and their first and second derivatives for evaluation. Although our system generates

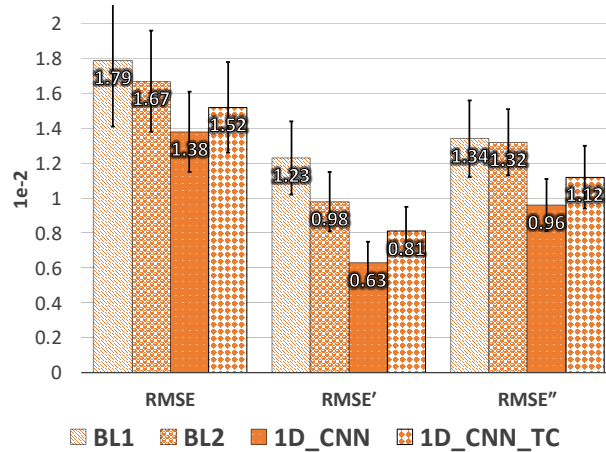


Figure 3.9: Single speaker objective evaluation results for the *BL1* [4], *BL2* [5], *1D\_CNN* and *1D\_CNN\_TC* methods. We calculate the root-mean-squared error (RMSE) between generated and ground-truth 2D mouth landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation.

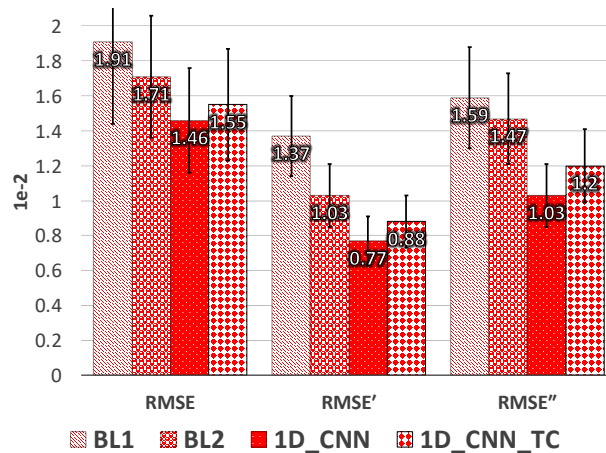


Figure 3.10: Multi speaker objective evaluation results for the *BL1* [4], *BL2* [5], *1D\_CNN* and *1D\_CNN\_TC* methods. We calculate the root-mean-squared error (RMSE) between generated and ground-truth 2D full face landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation.

3D landmarks, we used only x and y coordinates (2D landmarks) of the results of our system for these calculations since the baseline can only generate 2D face landmarks. Therefore, all numbers reported in this section were obtained from 2D landmarks. Before we evaluated the landmarks, we normalized the values between 0 and 1. Therefore, each 0.01 RMSE value corresponds to approximately 1 percent of the face length.



For the single-speaker setting, we evaluated our systems and the baseline systems with the test set of Obama dataset by using only the mouth landmarks. For the multi-speaker setting, we used unseen speakers from STEVI corpus. Figures 3.9 and 3.10 show the single- and multi-speaker results, respectively, for the baseline methods (*BL1* and *BL2*), and two versions of our proposed methods (*ID\_CNN* and *ID\_CNN\_TC*).

For the single-speaker setting, the results show that the *ID\_CNN* method yields the best objective results with an RMSE value of  $1.38 \times 10^{-2}$  followed by *ID\_CNN\_TC* with an RMSE value of  $1.52 \times 10^{-2}$ . For the multi-speaker setting, the trends are similar: the *ID\_CNN* method yields the best objective results with an RMSE value of  $1.46 \times 10^{-2}$  followed by *ID\_CNN\_TC* with an RMSE value of  $1.55 \times 10^{-2}$ . There is a significant improvement over the *BL1* method that has an RMSE value of  $1.91 \times 10^{-2}$ .

*ID\_CNN\_TC* results are smoother due to the temporal constraint. However, the resulting mouth movement of the talking faces has weaker high-frequency movements. This can also be observed from the multi-speaker objective results. The  $RMSE'$  and  $RMSE''$  are higher for *ID\_CNN\_TC* ( $0.88 \times 10^{-2}$ ,  $1.2 \times 10^{-2}$ ) compared to *ID\_CNN* ( $0.77 \times 10^{-2}$ ,  $1.03 \times 10^{-2}$ ); and both of them are better than the baseline methods. A paired t-test shows that results of both proposed systems are statistically significantly better than the baseline at a significance level of 0.01 for all the three measures.

There is a trade-off between these two versions of our method. From our observations of the generated outputs, *ID\_CNN* yields better mouth movement and mouth shape match, where *ID\_CNN\_TC* yields more stable and smoother shape changes over time. One may prefer *ID\_CNN* for applications that focus on improving speech comprehension since high-frequency mouth movement is essential in such cases, and one may prefer *ID\_CNN\_TC* for general speech animation applications. An example result for the word “ashes” has been shown in Figure 3.11.

### Analysis of the Network

We further analyze the *ID\_CNN* network architecture by changing the number of convolutional layers, the number of filters in each layer, and the input speech size in multi-speaker setting using STEVI corpus.



Figure 3.11: The example output showing the pronunciation of the word “ash”. The speech sample was taken from STEVI corpus. The first row shows the result generated by *ID\_CNN*. The second row shows the comparison of the result generated by *ID\_CNN* and the ground-truth (dotted red line). The third and fourth rows show the result generated by *ID\_CNN\_TC* and comparison with the ground-truth (dotted red line). Columns show every three frames.

**Number of Layers** The original configuration contains four convolutional layers. We conducted experiments with 2, 3, 4, 5 and 6 convolutional layers, and compared the objective results. For fifth and sixth layers, we used 512 filters.

The results are shown in Figure 3.12. The 4-layer configuration achieves the best results, where 5-layer configuration has the worst results. An interesting outcome is that the 2-layer configuration has the second best results. For  $RMSE''$ , there is a big gap between the 4-layer configuration and others. A paired t-test shows that  $RMSE'$  and  $RMSE''$  results of 4-layer configuration is statistically better at a significance level of 0.01 compared to other configuration results. In conclusion, we selected 4-layer configuration in our final models.

**Number of Filters** Table 3.2 shows the number of filters for the convolutional layers, which are  $x = 64$ ,  $2x = 128$ ,  $4x = 256$  and  $8x = 512$  for the four layers, respectively, in the original configuration.

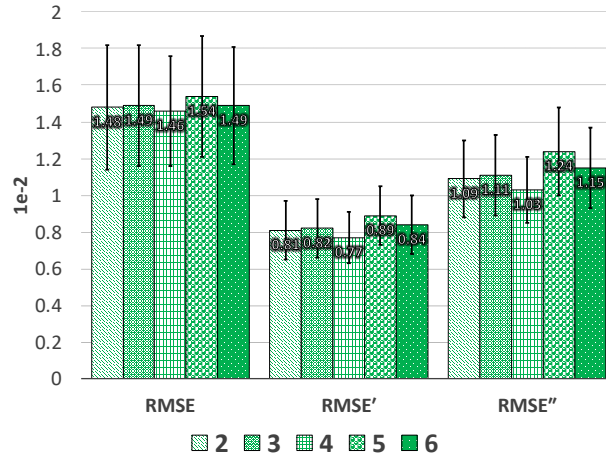


Figure 3.12: Comparison of *ID\_CNN* configurations with different number of convolution layers. The number of filters for Layers 1 to 4 is shown in Table 3.2. The number of Layers 5 and 6 is both 512. We compare the root-mean-squared error (RMSE) between generated and ground-truth landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation.

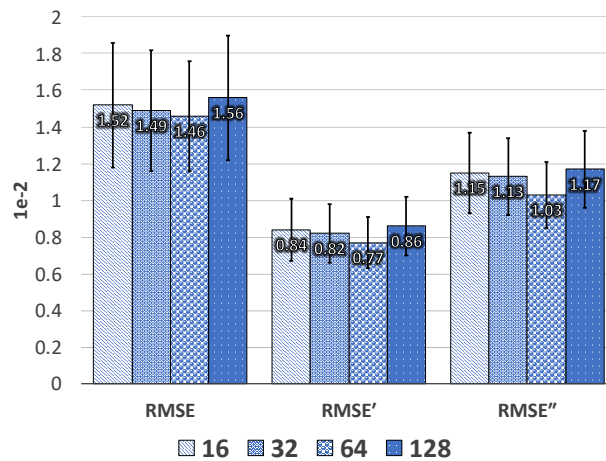


Figure 3.13: The comparison of *ID\_CNN* configurations that has a different number of filters in convolutional layers is shown. The number of filters in the first layer is displayed, which are 16, 32, 64 and 128. After the first layer, the filters are doubled with each following convolutional layer. We compare the root-mean-squared error (RMSE) between generated and ground-truth landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation.

We varied  $x$  to have values of 16, 32, 64, and 128 and compared the objective results.

Figure 3.13 shows the results. Networks with  $x = 16$  and  $x = 32$  performs similarly for all metrics. The network with  $x = 128$  has the worst performance compared to other configurations; We suspect that this is due to over-fitting given its largest capacity. The network with  $x = 64$  performs

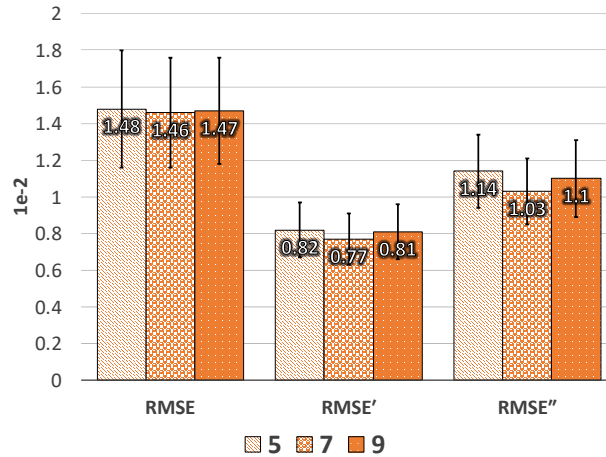


Figure 3.14: The comparison of results for different sizes of the input speech is shown for *ID\_CNN* network. The number of frames is displayed, which are 5, 7, 9. Each frame spans 40 ms speech. We predict the middle frame and use previous and past frames as context information. We compare the root-mean-squared error (RMSE) between generated and ground-truth landmarks, and its first order and second-order temporal derivatives. Error bars show the standard deviation.

better than other configurations. A paired t-test shows that RMSE' and RMSE'' results of  $x = 64$  configuration is statistically better at a significance level of 0.01 compared to other configuration results. Therefore, we selected  $x = 64$  as the final parameter for our networks.

**Input Speech Size** The input speech includes context information of past and future frames as described in Section 3.2.4. In the original configuration, we use 7 frames of speech, including 3 frames before and 3 frames after the current frame. Each frame corresponds to 40 ms of speech. In this section, we vary the input size from 5, 7, and 9 frames and compare the performance.

The results are shown in Figure 3.14. The RMSE results are similar; However, for RMSE' and RMSE'', 7 frames configuration has better results. A paired t-test shows that RMSE' and RMSE'' results of 7 frames configuration is statistically better at a significance level of 0.01 compared to 5 and 9 frames configuration results. In our final network, we selected 7 frames of speech as our input.

### Resilience Against Noise

In this section, we evaluate our system on noisy conditions. We consider five types of noise for the evaluations, namely babble, factory, speech-shaped noise (SSN), motorcycle and cafeteria. We mix

Table 3.3: Objective results for the *ID\_CNN* method and noise-resilient (NR) version of it (*ID\_CNN\_NR*) for clean and noisy speech input. We present results for Babble, Factory, SSN, Motorcycle and Cafeteria noises at 5 and 10 dB SNRs, none of which were not included in the training noise corpus. Best results in each noise setting are bolded.

Noise Type	SNR	Method	RMSE (1e-2)	RMSE' (1e-2)	RMSE'' (1e-2)
Clean	-	ID_CNN	1.46	<b>0.77</b>	<b>1.03</b>
		ID_CNN_NR	<b>1.45</b>	0.81	1.09
Babble	5	ID_CNN	1.53	0.84	1.11
		ID_CNN_NR	<b>1.46</b>	<b>0.82</b>	<b>1.10</b>
	10	ID_CNN	1.52	0.83	1.10
		ID_CNN_NR	<b>1.44</b>	<b>0.79</b>	<b>1.07</b>
Factory	5	ID_CNN	1.55	0.86	1.13
		ID_CNN_NR	<b>1.48</b>	<b>0.84</b>	<b>1.12</b>
	10	ID_CNN	1.54	0.84	1.11
		ID_CNN_NR	<b>1.48</b>	<b>0.82</b>	<b>1.10</b>
SSN	5	ID_CNN	1.50	0.81	1.12
		ID_CNN_NR	<b>1.48</b>	<b>0.80</b>	<b>1.09</b>
	10	ID_CNN	1.50	0.81	1.11
		ID_CNN_NR	<b>1.45</b>	<b>0.80</b>	<b>1.09</b>
Motorcycle	5	ID_CNN	1.50	0.79	<b>1.06</b>
		ID_CNN_NR	<b>1.43</b>	<b>0.79</b>	1.07
	10	ID_CNN	1.49	0.78	1.05
		ID_CNN_NR	<b>1.42</b>	<b>0.77</b>	<b>1.04</b>
Cafeteria	5	ID_CNN	1.52	0.82	1.08
		ID_CNN_NR	<b>1.46</b>	<b>0.81</b>	<b>1.06</b>
	10	ID_CNN	1.50	0.81	1.07
		ID_CNN_NR	<b>1.45</b>	<b>0.79</b>	<b>1.05</b>

the speech files of STEVI corpus with the noises described above in 5 and 10 dB signal-to-noise ratio (SNR) values and report the RMSE values of the generated faces. Note that the noise types used in evaluations were not included in the training set, and obtained from a different source (i.e., different recording conditions).

For the noise-resilient training method, we initialized the weights using the pre-trained weights from the clean version of our network and reduced the learning rate to  $10^{-5}$ . We conducted experiments and varied the  $\lambda$  parameter in Equation 3.6 between 1 and 0, and found that  $10^{-2}$  performs the best. Therefore, we set  $\lambda$  to  $10^{-2}$ . The network was trained for 100 epochs. The noise resilient (NR) version of our network is denoted as *ID\_CNN\_NR* throughout the rest of the paper.

Table 3.3 shows results of the proposed system with and without the noise-resilient training method on both clean and noisy speech. For the clean speech, noise-resilient training (*ID\_CNN\_NR*) provides a slight improvement on RMSE at the cost of RMSE' and RMSE'' compared to the originally proposed method. For noisy speech, however, the noise-resilient training yields significant improve-

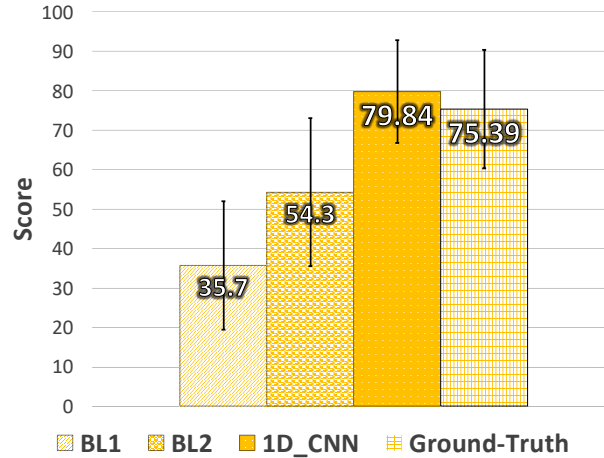


Figure 3.15: The results for the subjective test of speech-mouth match. The bars show the average score for the baseline method, proposed method (1D\_CNN) and ground-truth face landmarks. Error bars show the standard deviation.

ments over the proposed method for all unseen noise types and SNRs on all evaluation measures. We conducted a paired t-test between the results of *ID\_CNN* and *ID\_CNN\_NR* for each noise and SNR category as shown in Table 3.3. The results show that the bolded RMSE values are statistically significant at a significance level of 0.01. Among these non-stationary noises, the babble and cafeteria noises are the most challenging ones for the proposed system. The motorcycle noise is the least challenging noise, and *ID\_CNN\_NR* can yield objective results even better than those for clean speech.

### Subjective Evaluation

To further evaluate the match between generated face landmarks and the input speech, we conducted a subjective Turing test in the multi-speaker setting. We recruited 20 volunteers as our evaluators. We presented each evaluator a random selection of 16 samples generated by *BL1*, 16 samples generated by *BL2*, 16 samples generated by the proposed system, and 16 samples of ground-truth landmarks. All of the speech samples were taken from the STEVI dataset, which was not used for training. For the *BL1*, we retrained the system with full 68 face landmarks' PCA coefficients instead of just the mouth landmarks' PCA coefficients in order to conduct the subjective tests. We found out that using only the mouth region compared to using all 68 face landmarks does not change mouth

movements. This is due to the alignment of face landmarks in pre-processing; the regions besides the mouth region do not change much.

The generated landmarks were painted and added teeth and eyes in order for evaluators to easily recognize the faces and mouth movements. During evaluation, a few ground-truth talking face samples were shown to each evaluator. Then, the 64 samples were presented to the evaluator in a random order, and the evaluator was asked to assign a score between 0 (worst) and 100 (best) based on the match between the speech and mouth movement. Each sample was presented twice before the evaluator was asked to assign a score.

The results are shown in Figure 3.15. The proposed method significantly scores higher than the baseline methods. These results show comparable scores for our method and the ground-truth face landmarks, indicating that our system can generalize well to unseen speakers and can convince evaluators that speech and articulation match strongly. A paired t-test shows that the *ID\_CNN* results are statistically significantly better compared to the both baseline results at the significance level of 0.01.

### **Limitations**

As a data-driven approach, the performance of our method highly depends on the the training data. The dataset should contain a wide variety of phonemes, ideally uniformly distributed. However, the GRID dataset is limited in terms of the words and phonemes it includes. Our future work includes expanding the training set to include more data that has rich phonetic content and balancing the data in order to have uniformly distributed phoneme content.

The performance of our system is proportional to the performance of the face landmarks extractor on the training data. The extractor we used in this study works on each single frame and does not consider temporal relations across frames. This might be the main reason for noisy mouth movements in the extracted landmarks. We believe that by utilizing a video-based face landmark extractor that models temporal dependencies of landmarks, the quality of landmark extraction and our trained model will be improved.

### 3.2.5 Conclusion

In this work, we proposed a new noise-resilient neural network architecture to generate 3D face landmarks from speech in an online fashion that is robust against unseen non-stationary background noise. The network predicts active shape model (ASM) coefficients of face landmarks from input speech. In one version of the system, we further added the predicted ASM coefficients in the previous frame to the network input to improve the smoothness of frame transitions. We conducted objective evaluations on landmark prediction errors and subjective evaluations on audio-visual coherence. Both objective and subjective evaluations showed that the proposed method statistically significantly improves over state-of-the-art baseline methods. Detailed analyses of network hyper-parameters were also provided to gain insights into the architecture design. To promote scientific reproducibility, we provided the research community with our pre-trained models, code and generation examples.



## Chapter-4

# Generating Talking Faces From Speech: Image-Based Methods

## 4.1 End-to-End Talking Faces from Speech

### 4.1.1 Introduction

Visual signals influence speech communication in several ways. First, the presence of lip images/videos has been shown to increase speech comprehension [8, 9, 10, 11]. The benefit is more prominent in scenarios where there is background noise or when the speech signal is corrupted by other effects such as channel compression and transmission loss. Next, the meaning of the message can be better interpreted since seeing the facial expressions improves the ability to infer emotions, which can change the meaning of the message.

In this work, we present an end-to-end talking face generation system that works with an arbitrarily long speech input and a single reference image of a target face. The network utilizes an attention mechanism [159]. The raw speech waveform is processed by a speech encoder to extract short-term speech features. The image quality and speech-mouth synchronization are further improved by employing generative adversarial networks.

### 4.1.2 Related Work

Some of the works in this field leverage sparse face points to generate talking face videos. These works usually contain two networks, one of which predicts face landmarks from speech, while the other network maps the face landmarks to images. These networks can be trained independently, i.e., the training is not end-to-end. Suwajanakorn et al. [4] proposed such a system. Their system can generate videos of President Barack Obama from his speech. First, they predict the PCA coefficients of the mouth landmarks from speech features (13 MFCCs plus the energy) using an LSTM network. The texture is synthesized according to the predicted PCA coefficients by selecting a few nearest candidate frames from the dataset that contains the images of the target identity and applying the weighted mean to them. However, this method is designed for a single-person and requires a considerable amount of data.

Chen et al. [19] proposed a similar method that can work on unseen speakers. First, an LSTM network generates the PCA coefficients of 68 face landmarks from MFCC coefficients. Another network, which receives the reference identity, reference face landmarks and the predicted face landmarks, generates the talking faces. The two networks are trained separately, where the second network is trained with ground-truth landmarks. During inference, the talking face landmarks are predicted using the first network, and the second network converts the landmarks into images. The network contains an attention mechanism, which can emphasize the image regions that correlate with landmark movements.

Another common approach is to generate talking faces from speech features. Chung et al. [20] proposed a method that generates talking faces from 12 MFCCs and a single frame target image. However, the generated images are blurry. Therefore, the authors trained a separate deblurring module to sharpen the images, meaning that it is not an end-to-end system. Similarly, Chen et al. [21] proposed a lip generation system with an adversarial loss function in addition to a reconstruction loss. The network accepts the MFCC features and a target lip image as inputs and outputs 16 frames of talking lip images.

Song et al. [22] improved Chung et al.'s work by proposing a recurrent neural network (RNN) that

is conditioned on the audio and reference image features. Their system uses a multi-task discriminator to improve the image quality of each frame and the natural movement of the video. A similar work that uses temporal GANs is proposed by Vougioukas et al. [23]. The system uses an RNN to process the noise that is generated independently for each frame, similar to [160]. Different from Song et al. [22], their system utilizes two discriminators, one for improving the image quality and the other for improving the natural video movements. Also, most importantly, their system is end-to-end.

### 4.1.3 Method

In this section, we describe how our system works, and show the details of each component of our system.

#### System Overview

Our system contains an image encoder, a speech encoder, an encoder for processing latent noise, a generator, a frame discriminator, and a pair discriminator. The image encoder accepts the reference face image and outputs image features, where the speech encoder accepts the raw speech waveform and produces short-term speech features. An encoder that contains an LSTM layer processes a normally distributed random noise vector for each frame into temporally meaningful latent variables. The generator maps these features into the talking face videos. The objective function is formed from the reconstruction loss and the adversarial losses from the pair and frame discriminators. The system overview is shown in Figure 4.1.

#### Mouth Region Mask

The movement of the regions in the image other than the mouth does not depend on the input speech. Therefore, we apply the reconstruction loss only to the mouth region. We detect the center point of the mouth using a face landmark detector and place a 2D Gaussian mask at that point. We call this mask the mouth region mask ( $M_{MR}$ ). Figure 4.2 shows an example mouth region mask.

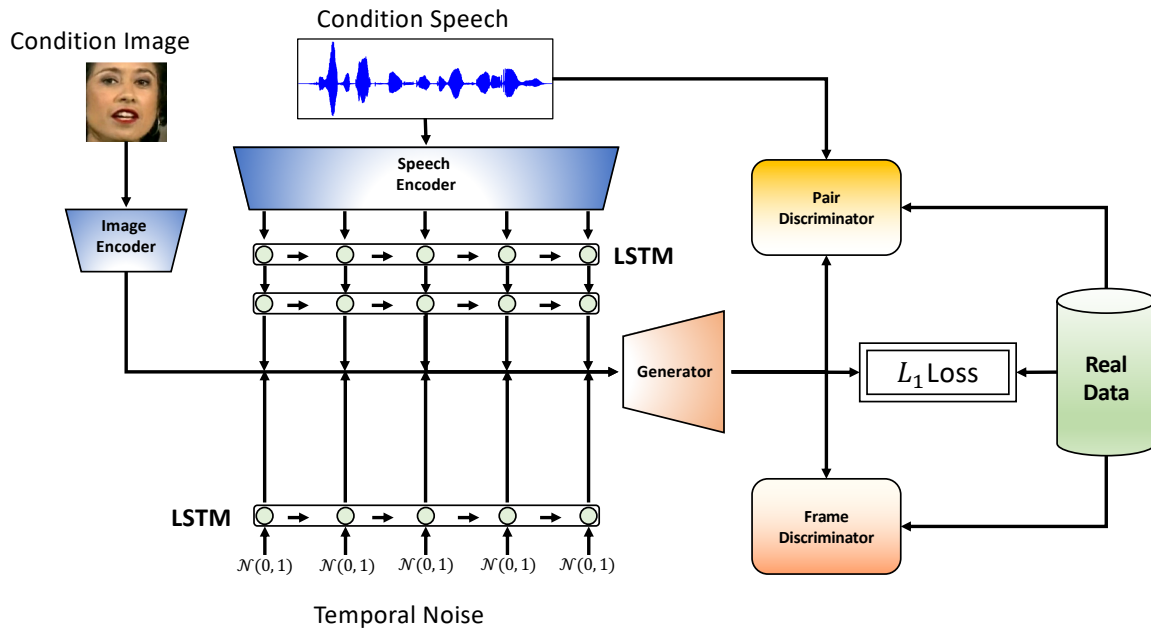


Figure 4.1: The proposed end-to-end talking face generation system overview. The input reference image and raw speech waveform are processed by the image encoder and speech encoder, respectively. For each frame, a normally distributed random noise vector is generated and fed to the noise encoder that contains an LSTM layer. The image, speech, and noise features are sent to the generator. During training, we use both the adversarial loss and the reconstruction loss. The frame discriminator improves the image quality, where the pair discriminator improves the mouth movements and speech synchronization.

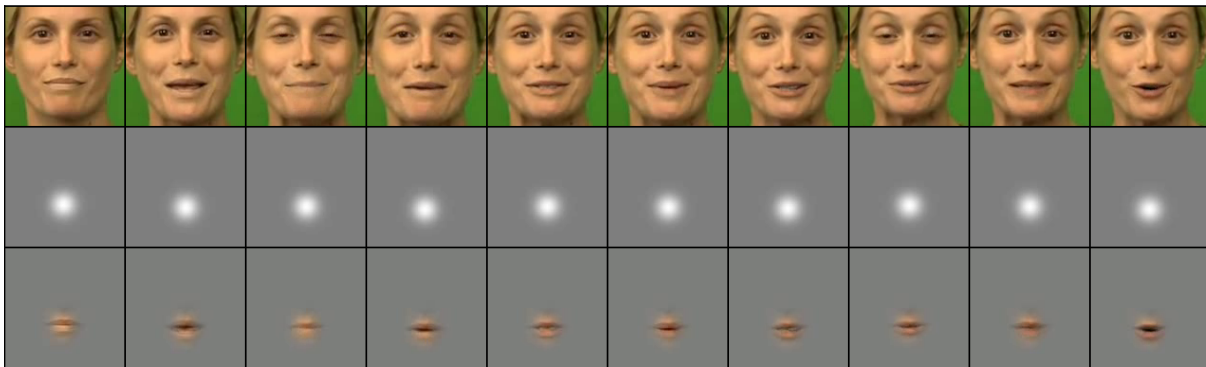


Figure 4.2: An example of the mouth region mask is shown. The mouth is located using the mean point of mouth face landmarks. A 2D Gaussian is placed at the mean point to isolate the mouth. The first row shows the original frame, the second row shows the mouth region mask, and the third row shows the masked mouth.

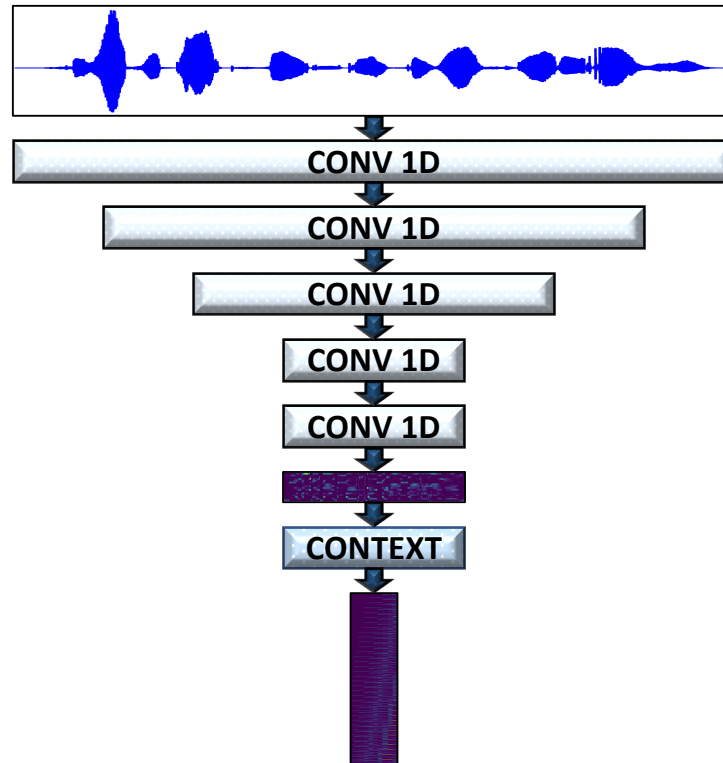


Figure 4.3: The architecture of the speech encoder. The network accepts an arbitrarily long speech waveform and processes it frame by frame through five convolutional layers. The resulting embedding is a time-series corresponding to these frames. Past and future frames as context information are also fed to the network as input when the network processes each frame. For the beginning and ending frames of the waveform, we concatenate zeros as the context information. Every fifth frame is kept to form the final speech features.

### Spec Encoder

The speech encoder takes the raw speech waveform as an input and outputs short-term speech features, similar to what the short-term Fourier transform (STFT) does. The encoder contains only 1D convolutional layers, enabling arbitrarily long speech inputs. It is designed for 8 kHz speech signals; for 1 second of speech (8000 data points), the output feature size is  $125 \text{ frames} \times F_{speech}$ , where  $F_{speech}$  is the dimension of features in a single frame. The receptive field is approximately 86 ms for each frame. The architecture is shown in Figure 4.3 and Table 4.1.

The stride of the first two convolutional layers is 4, and it is reduced to 2 for the following two layers. The filter size of the first layer is 63, and it gradually decreases to 31, 17, and 9 in the next layers. Each convolutional layer is followed by a batch normalization layer, a leaky ReLU activation,

Table 4.1: Detailed parameters of the proposed network architecture. The number of filters and hidden units, filter sizes, strides, activation functions, and output shapes are shown for each layer.

Net	Layers	Number of Filters or Hidden Units	Filter Size	Strides	Activation	Output Shape
Speech Encoder	Input	-	-	-	-	$(N_{speech}, 1)$
	Conv	64	(63, 1)	(4, 1)	LeakyReLU	$(N_{speech}/4, 64)$
	Conv	128	(31, 1)	(4, 1)	LeakyReLU	$(N_{speech}/16, 128)$
	Conv	256	(17, 1)	(2, 1)	LeakyReLU	$(N_{speech}/32, 256)$
	Conv	512	(9, 1)	(2, 1)	LeakyReLU	$(N_{speech}/64, 512)$
	Conv	$F_{speech}$	(1, 1)	(1, 1)	LeakyReLU	$(N_{speech}/64, 512)$
Image Encoder	Input	-	-	-	-	(128, 128, 1)
	Conv	64	(7, 7)	(1, 1)	LeakyReLU	(128, 128, 64)
	Conv	64	(3, 3)	(2, 2)	LeakyReLU	(64, 64, 64)
	Conv	128	(3, 3)	(2, 2)	LeakyReLU	(32, 32, 128)
	Conv	256	(3, 3)	(2, 2)	LeakyReLU	(16, 16, 256)
	Conv	512	(3, 3)	(2, 2)	LeakyReLU	(8, 8, 512)
	Conv	512	(3, 3)	(2, 2)	LeakyReLU	(4, 4, 512)
Noise Encoder	Input	-	-	-	-	$(t, F_{noise})$
	LSTM	$F_{noise}$	-	-	Tanh	$(t, F_{noise})$

and a dropout layer. The last convolutional layer’s filter size and stride is set to 1, and it does not include batch normalization and dropout.

We concatenate the  $N_{ctx}$  past and future frames to include context information. Adding context information allows us to reduce the frames even further: we only include every 5th frame to build our final speech features  $z_{speech}$ . For 1 second of speech, the dimension of  $z_{speech}$  becomes  $25 \times F_{speech}((2N_{ctx}) + 1)$ . We aim to generate 25 frames-per-second videos, and by using our speech encoder, we can design a sequence-to-sequence generator that works with arbitrary lengths of speech input.

### Image Encoder

The image encoder takes the reference image, i.e., the target identity and outputs the image features. We use 2D convolutional layers followed by leaky ReLU activation in each layer. The parameters of the architecture are shown in Table 4.1. The image encoder sends all intermediate features to the generator. They are concatenated to the generator’s intermediate layers as in a U-net architecture [161].

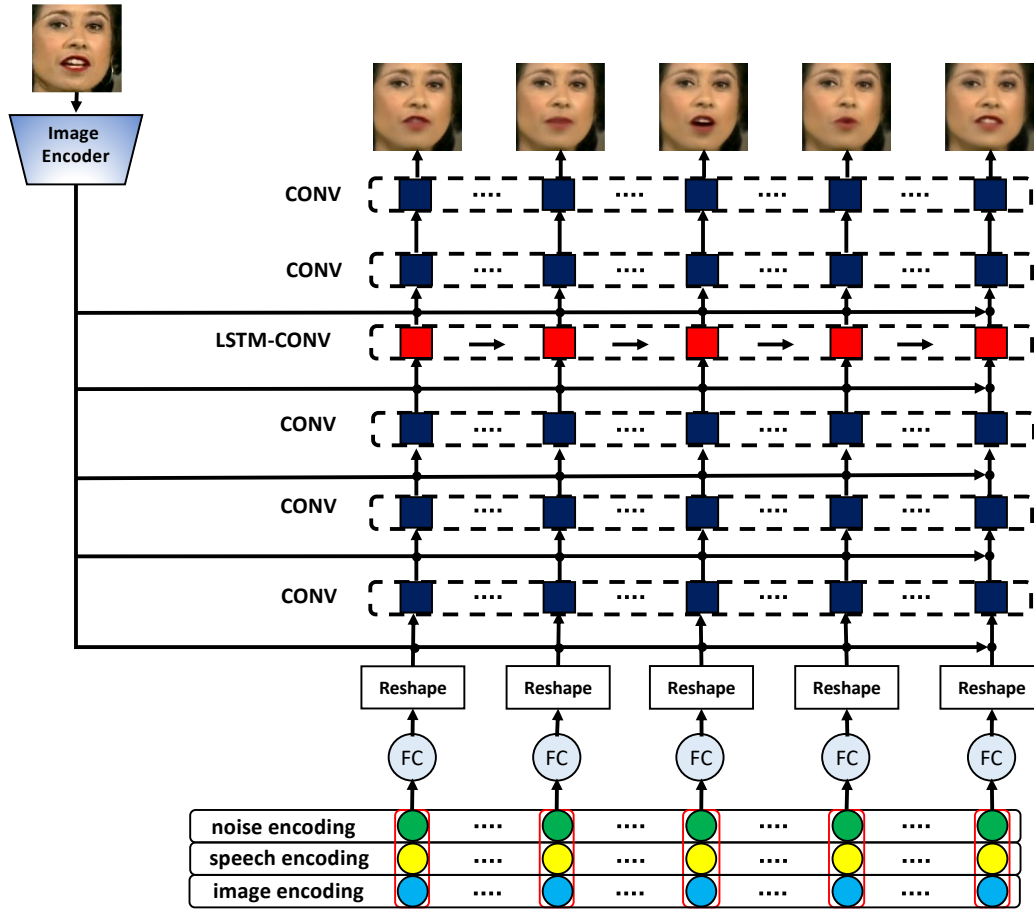


Figure 4.4: The architecture of the generator. The noise, image, and speech features are concatenated at each frame and fed into a fully connected layer, and the output is reshaped. Then, the results are concatenated with skip connections from the image decoder and fed into a convolutional layer. This is repeated for all layers except for the last convolutional layer.

## Generator

The generator takes the speech and reference image features, and noise as inputs. The noise is a sequence sampled from  $N(0, 1)$  for each frame, which is processed by a single layer unidirectional LSTM to produce a temporally meaningful noise sample.

For each frame, the latent noise code and speech features are concatenated and fed into a fully connected layer. Then, it is reshaped and fed into a convolutional layer. The skip connections coming from the image encoder are concatenated with the intermediate features in each layer except for the last layer. We employ an LSTM-Convolutional layer to improve the movements of the mouth over

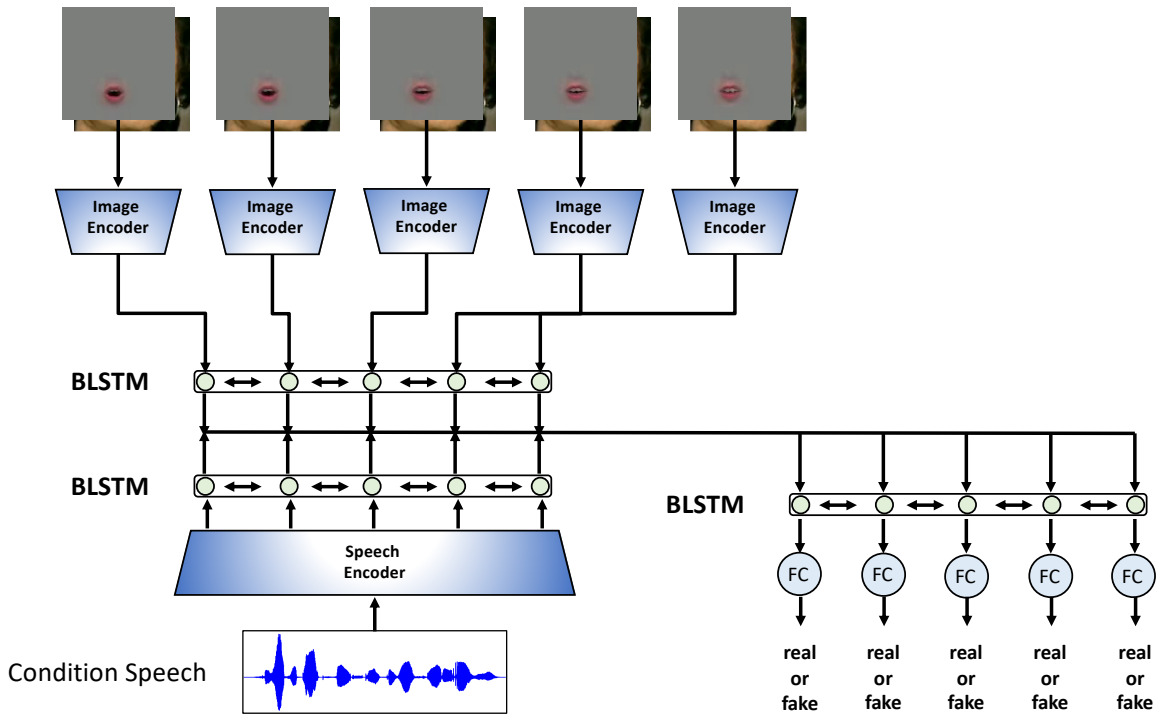


Figure 4.5: The architecture of the pair discriminator is shown. The input is the masked mouth frames, condition speech, and condition image. The image and speech encoders are identical to our main speech and image encoders, but the parameters are updated only during discriminator training. Each frame is classified as real or fake.

time.

### Frame Discriminator

The frame discriminator takes the individual frames of the generated and real videos, concatenated to the reference image, to improve the quality of the generated frames. The frame discriminator is a six-layer convolutional neural network that outputs binary patches as in a pix2pix network [162].



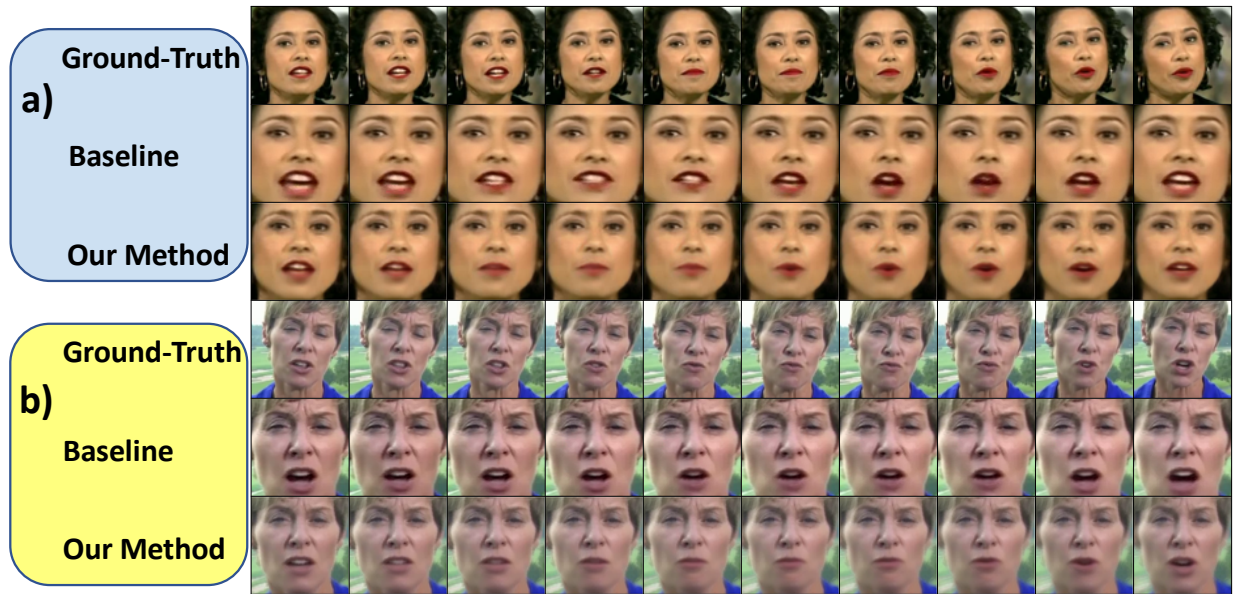


Figure 4.6: Example generation results shown along with the baseline results and the ground-truth. a) shows an utterance of “POINT” and b) shows an utterance of “RUSSIANS”.

### Pair Discriminator

The pair discriminator tries to distinguish the real and fake pairs of videos and speech inputs. We utilize a separate speech encoder that has the same architecture described in Section 4.1.3. After the speech features are extracted, they are processed by a BLSTM layer. For the video frames, we utilize a separate image encoder that has the same architecture we use for our main image encoder. The extracted image encoding for each frame is fed into BLSTM layer. The outputs of both BLSTMs are concatenated and fed into another BLSTM. Each output of the final BLSTM is classified as real or fake. The pair discriminator is shown in Figure 4.5.

The purpose of this discriminator is to check each frame to see if they are aligned with the input speech.

## 4.1.4 Experiments

### Dataset

In our experiments, we use the lip reading in the wild (LRW) dataset [163]. This is a large-scale audio-visual dataset that contains short video clips (1.16 seconds) of people uttering a single word. There are in total 500 words in the dataset and 1000 clips for each word.

### Results

The generated talking faces are shown in Figure 4.6, along with the ground-truth and the baseline [19] results. From visual inspections of a small subset of the results, we find that our method seems to generate mouth shapes that better match with those of the ground truth. This is still ongoing work, and large-scale subjective and objective evaluations will be conducted in the near future.

## 4.1.5 Conclusion

We developed an end-to-end talking face generation system that operates on a raw speech waveform and a reference image. The system works with arbitrary length speech inputs and utilizes generative adversarial networks to improve image quality and mouth-movement-speech synchronization. This is ongoing work; large-scale objective and subjective evaluations will be used to characterize the performance of this end-to-end talking face generation system.

## **Chapter-5**

# **Automatic Speech Emotion Recognition (ASER)**

## **5.1 Introduction**

In this chapter, we introduce the automatic speech emotion recognition (ASER) problem and analyze the different aspects and challenges of ASER. Our final goal is to generate emotionally expressive talking faces. However, first, we need to be able to estimate emotions from speech accurately and use this information as a condition when generating the talking faces. Therefore, this chapter's primary focus is estimating emotions from speech.

## **5.2 Amazon Mechanical Turk Study**

### **5.2.1 Introduction**

Emotion classification is a fundamental task for humans in order to interpret social interactions. Although emotions are expressed at various levels (e.g., behavioral, physiological), vocal and verbal communication of emotions is a central domain of communication research [164]. Classification accuracy is essential in order to be ensured of the validity and reliability of emotional constructs used in psychological research. Given the importance of accurately classifying emotions to understanding human interactions, many researchers have developed automatic emotion classification computer systems. There are a number of modalities that can be used to determine one's emotions, including facial expression, body movement, physiological measures such as galvanic skin response, and voice.

While automatic emotion classification systems have been developed that use all of these modalities, individually and in concert [165, 166, 167, 168], several systems have focused on classifying emotions using speech features in particular [169, 170, 171, 172]. There are a number of reasons for this, including the fact that speech is relatively easy to capture and is less intrusive than other methods for capturing emotional state. While these speech-based automatic emotion classification systems all provide reasonable accuracy in their classification results, it is not known how well these systems, which in many applications would replace a human's classification of the emotion, compare to a naive human coder performing the same emotion classification task.

In this work, we compare how well an automated computer system can perform at the task of emotion classification from speech samples compared with naive human coders. In particular, we asked Amazon Mechanical Turk workers (Turkers) to listen to speech samples from the LDC dataset of emotions [173] and classify them in three ways: 1) determine whether the conveyed emotion was active, passive or neutral; 2) determine whether the conveyed emotion was positive, negative or neutral; and 3) determine which of six emotions (happy, neutral, sad, anger, disgust, fear) was being conveyed. We also asked the Turkers how confident they were in their classification. We compared the Turkers' accuracy with that achieved by a speech-based automated emotion classification system [169], using a leave-one-subject-out (LOSO) approach for evaluating the system.

Our results show that the automated system has a higher emotion classification accuracy compared with the Turkers' accuracy averaged over all six emotions, with the automated system able to achieve close to 72% accuracy compared with the Turkers' accuracy of only about 60%. Additionally, while the automated system can achieve even better accuracy by rejecting samples when its confidence in the classification is low, the Turkers' results for the samples in which they were confident about their classification did not show any significant improvement compared with the accuracy of all their responses. These results suggest that an automated speech-based emotion classification system can potentially replace humans in scenarios where humans cannot be easily trained.

## 5.2.2 Related Work

To date, only a few studies have been conducted to compare the performance of automatic systems with that of humans for emotion classification. Some of these studies use visual facial expressions to determine emotion [174, 175, 176], but these are out of the scope of this study, which focuses on comparing human and machine performance for emotion classification based on speech.

For the existing studies on human emotion classification from speech, the number of human subjects used is relatively small. In addition, whether the human subjects were trained for the specific emotion classification task or not is not always specified. In [177], Shaukat et al. compared a psychology-inspired automatic system that utilizes a hierarchical categorization model based on multiple SVMs with humans' ability to classify emotions from speech on two databases, the Serbian Emotional Speech Corpus (GEES) and the Danish Emotional Speech Corpus (DES). For the experiments with humans, there were 30 subjects for the GEES, and 20 subjects for the DES. Results showed that the automatic system slightly underperformed humans for both databases.

In [178], Esparza et al. employed a multi-class SVM system to classify speech emotions, and compared its performance with humans on two German databases, the "corpus of spoken words for studies of auditory speech and emotional prosody processing" (WaSeP), and the Berlin Database of Emotional Speech (EmoDB). The WaSeP corpus was evaluated by 74 native German speakers with an accuracy of 78.5%, and the EmoDB corpus was evaluated by 20 native German speakers with an accuracy of 84.0%. Computer system accuracies were 84.0% and 77% for the WaSeP and EmoDB databases, respectively. In this case the results (whether humans or the automated system perform better) were mixed. A final study considered a Hungarian database evaluated by both humans and an automated emotion classification system that utilized Hidden Markov Models (HMMs) [179]. The evaluation was performed by 13 subjects, where the subjects never heard the same speaker successively. The evaluation included utterances that contained emotion as well as neutral utterances. The authors evaluated the 4 best emotional categories for the computer system with average accuracy around 85%, and they evaluated the 8 best emotion categories for the human subjects, with average accuracy of 58.3%. The results showed that the humans provided better evaluations for the sad and

disgust emotion categories, while the computer system provided better evaluations for the surprised and nervous emotion categories.

In this work, we conducted a large scale comparison between a state-of-the-art speech-based emotion classification system with the performance of 138 human subjects classifying 7270 audio samples. These human subjects were recruited using the Amazon Mechanical Turk service. Compared to existing studies, our experiment used more human subjects with much higher diversity both demographically and geographically. In addition, these human subjects were not trained on the dataset used in the experiment.

### 5.2.3 LDC Dataset

In this study, we use the LDC dataset, a collection that includes speech samples with 14 distinct emotion categories recorded by professional actors, 3 male and 4 female, reading semantically neutral utterances such as dates and times [173]. Note that using semantically neutral utterances is a common practice in speech-based human emotion classification studies [178, 177, 179]. The length of the utterances varies between one and two seconds. In our study, we used a total of 727 utterances that contained the emotions happy (136), neutral (67), sad (157), anger (136), disgust (108), and fear (123). Each emotion was also labeled as active (happy, anger, fear), passive (sad, disgust) and neutral as well as positive (happy), negative (sad, anger, disgust, fear) and neutral.

### 5.2.4 Automated Emotion Classification System

There are a number of systems that automatically classify emotions from speech [170, 171, 180, 172, 181]. In this work, we use the one described in [169], as it has been shown to achieve similar or better classification accuracy than several other state-of-the-art systems [170, 171, 172] and it has the added advantage that it can reject samples as unclassified if it is not a confident classification. The rejection mechanism is useful in scenarios where classification is not required on all samples and the cost of an incorrect classification is high; hence, it is better to simply not classify some samples in order to achieve a much higher classification accuracy on all classified samples. Here, we briefly

overview this emotion classification system.

In this system, speech utterances are divided into 60 ms frames with a hop size of 10 ms. For all voiced frames (frames that contain voiced speech), several features are calculated, including: fundamental frequency ( $F_0$ ), energy, frequency and bandwidth of the first four formants, and 12 mel-frequency cepstral coefficients (MFCCs), zero crossing rate, spectral rolloff, brightness, centroid, spread, skewness, kurtosis, flatness, entropy, roughness, irregularity and the derivative of all features [182]. Five statistics of these features (mean, standard deviation, min, max, and range) are then calculated over all speech frames to obtain utterance-level features. Additionally, speaking rate is calculated for each utterance. This provides a total of 331 features for each utterance.

A classification system with 6 one-against-all (OAA) support vector machines (SVM), one for each emotion, with radial basis function (RBF) kernels, is then trained using the features extracted from training data together with their ground-truth emotion labels. This system is then able to classify new unseen utterances. For an unseen utterance, each OAA classifier outputs a confidence value, indicating the classifier's confidence that the utterance conveys that particular emotion. The confidence values of all 6 classifiers are compared, and the final emotion label of the utterance is determined by the classifier with the highest confidence.

In many scenarios, a classification does not have to be made for every utterance, yet when a classification is made, the cost of an incorrect classification is high. To deal with these scenarios, the system is also equipped to perform thresholding fusion, as per the approach in [183]. If the highest confidence value is below a threshold, the system rejects the sample. Only if the confidence value is above a threshold will the system provide a label for the utterance.

The system also employs speaker normalization, training set over-sampling, and feature selection to enhance the classification performance [169]. Speaker normalization (z-score normalization [184]) is used to normalize the distribution of the features of each speaker. This is to cope with the problem that different speakers may have distinct speech characteristics such as loudness. Training set over-sampling is used to overcome the problem of having an unbalanced training. SMOTE [185] over-sampling method is used, where synthetically created samples are added to the training set to balance the training data set. Feature selection is employed to select the most effective features from the 331

features for the classification. While in prior work [169], Mutual Information (MI) was used, here we use an SVM Recursive Feature Elimination [186] method instead, as we found that this approach can provide overall better performance in terms of classification accuracy using a subset of the features.

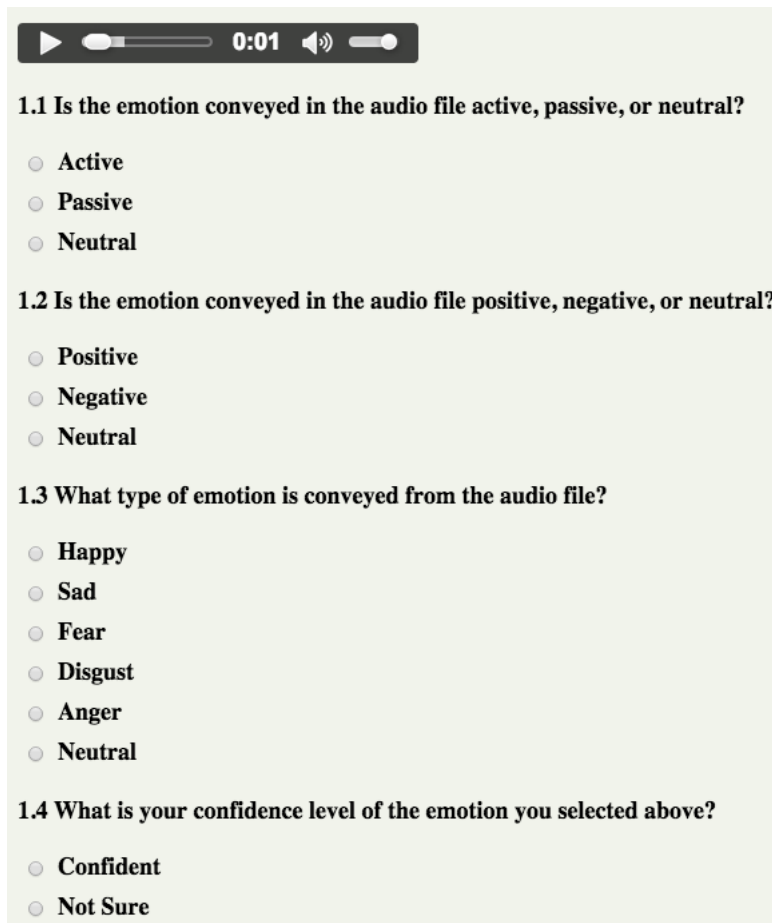
Similar OAA-SVM classification systems were trained for active-passive-neutral (APN) and positive-negative-neutral (PNN). These systems also use the thresholding fusion mechanism to reject utterances for which they are not confident enough, in order to improve the classification accuracy of those utterances that are ultimately classified.

To evaluate these systems, we conduct leave-one-subject-out (LOSO) tests, where the OAA binary classifiers are trained using speech utterances from all but one of the speakers in the dataset, and then tested using the one speaker left out of the training phase. In this way, we can determine the performance of the system when it has not been trained on the individual speaker, as would be the case for a number of applications where the system can be trained on the class of speakers it will encounter but it cannot be trained using samples from the target person. The results represent the average over all 7 LOSO cases using the 7 speakers in the LDC dataset.

### 5.2.5 Amazon's Mechanical Turk Setup

For the MTurk experiment, each Turker was provided with initial instructions about the task. These instructions included a sample of each of the different emotion categories to provide some minimal training of the Turkers. After reviewing the instructions, the Turkers were presented with a random selection of 10 to 100 audio samples to classify. After listening to each audio sample, as shown in Figure 5.1, Turkers were asked if the emotion conveyed in the sample was 1) active, passive or neutral, 2) positive, negative or neutral, and 3) one of the six emotions. Additionally, the Turkers were asked to rate their confidence in their labeling of the emotion. Finally, they were asked to transcribe what they heard in the audio file in order to ensure that the Turker actually listened to the audio sample. After completing the classification of all audio samples, the Turkers were asked to provide demographic information, including gender, age and race. Note that only Turkers whose native language is English are requested for this task. Once the MTurk task was completed and





▶ 0:01 🔊

**1.1 Is the emotion conveyed in the audio file active, passive, or neutral?**

- Active
- Passive
- Neutral

**1.2 Is the emotion conveyed in the audio file positive, negative, or neutral?**

- Positive
- Negative
- Neutral

**1.3 What type of emotion is conveyed from the audio file?**

- Happy
- Sad
- Fear
- Disgust
- Anger
- Neutral

**1.4 What is your confidence level of the emotion you selected above?**

- Confident
- Not Sure

Figure 5.1: Questions shown to Turkers.

approved by us, the Turkers were paid \$0.50 for each group of 10 speech samples they classified, in exchange for their time.

Table 5.1 shows the number of samples classified by the Turkers, broken down by gender and age. There were 138 unique Turkers that classified 7,270 audio samples, with individual Turkers classifying between 10 and 100 audio samples.

## 5.2.6 Evaluation

The goal in our experiments is to compare the performance of the automatic emotion classification system described in Section 5.2.4 with the performance of naive human coders, the Turkers from our MTurk experiment described in Section 5.2.5. In order to provide a fair comparison, we present results for the leave-one-subject-out (LOSO) case for the automatic emotion classification system. In

this case, the training set does not contain any samples from a particular speaker who is used in the test set.

In this section, we compare three different ways of classifying the utterances: 1) classifying the utterances to one of 6 different emotions; 2) classifying the utterances as active, passive or neutral; and 3) classifying the utterances as positive, negative or neutral. For each case, we compare the results of the automatic system with the Turkers when classifying all samples as well as when classifying only those samples for which they are confident. Additionally, we provide data showing the performance of the male and female Turkers, and for the performance for female and male utterances separately.

### **Classifying the Utterances: 6 Emotions**

The first task is to classify the samples into the six emotion categories mentioned in Section 5.2.3. Table 5.2 shows the accuracy values for the computer system and the Turkers for this task. Overall, the average accuracy for the Turkers is 60.4%, which is 12.5% worse than the automatic emotion classification system, which provides an accuracy of 72.9%. As shown in this table, the Female Turkers performed slightly better than the Male Turkers (1.1% improvement).

Also shown in Table 5.2 are the different accuracy values for the computer system and the Turkers when considering only the female or male utterances. It is interesting to note that the computer system performs slightly better (1.2%), while the Turkers perform significantly better (10.8%) for the female utterances.

We compare the accuracy values for the samples where the Turkers were confident about their classification with the accuracy values when the automatic emotion classification system is confident (here we use two different thresholds such that either 50% or 80% of the samples are classified, with all others being rejected). If we compare the Turkers' accuracy in classifying the emotions when all samples are classified with the accuracy when only those samples for which they were confident in their classification are considered, we see very little difference in the accuracy values (60.4% vs. 60.6%). This tells us that humans are not able to accurately estimate their performance and reliability on the emotion classification task. On the other hand, if we look at the automatic emotion classification system results, we see that when the computer rejects as unclassified the samples for

Table 5.1: Number of samples classified by Turkers.

All (7270)								
Female (2850)				Male (4350)				NA (70)
Ages				Ages				Ages
18-29	30-39	40-49	50-59	18-29	30-39	40-49	50-59	18-29
1300	630	620	300	2610	940	550	250	70

Table 5.2: Accuracy values (%) for six emotions.

Accuracy	Overall	Speaker Gender		Classification Confidence Level			
		Female	Male	Confident (80%)	Confident (50%)	Unsure (20%)	Unsure (50%)
Computer System	72.9	73.2	72.0	77.7	85.4	61.2	55.3
All Turkers	60.4	64.9	54.1	60.6 (80.5% confident)		59.6 (19.5% unsure)	
Female Turkers	61.2	64.4	57.1	60.4 (78.4% confident)		62.9 (21.6% unsure)	
Male Turkers	60.1	65.4	52.5	60.8 (82.0% confident)		57.9 (18.0% unsure)	

Table 5.3: Confusion matrix for the automatic classification system (GT = ground truth).

	Anger	Disgust	Fear	Happy	Neutral	Sad
Anger (GT)	92.9	0.0	2.4	2.5	0.0	2.2
Disgust (GT)	0.9	80.7	0.9	6.0	1.1	10.3
Fear (GT)	4.3	0.0	85.2	8.9	0.0	1.6
Happy (GT)	5.6	3.5	8.2	79.0	1.5	2.2
Neutral (GT)	0.0	4.2	0.0	2.4	86.3	7.2
Sad (GT)	0.0	5.1	0.0	0.8	1.5	92.6

which the confidence values from the OAA SVM are low, the accuracy of those samples that are classified increases from 72.9% to 77.7% when 80% of the samples are classified and to 85.4% when 50% of the samples are classified. Hence, we see that one clear advantage of an automatic emotion classification system over human coders is this ability to improve classification accuracy by rejecting to classify some samples. In applications where not all samples must be classified and the cost of mis-classification is high, this can be a valuable means to increase emotion classification accuracy.

The final set of numbers shows the accuracy of the utterances that are rejected by the automatic classification system or for which the Turkers were unsure of their classification. From this data, we can see that there is not much difference in accuracy for the set where the Turkers were confident (60.6%) and for the set where the Turkers were not confident (59.6%). Additionally, this data shows that when 20% of the samples are rejected by the automatic classification system, the accuracy on those rejected samples is 55.3%. Hence, some of the rejected samples (55.3%) were actually correctly classified. However, it is impossible to know which ones, and including this set of classifications makes the overall classification accuracy drop, and in some applications this is not a good trade-off

Table 5.4: Confusion matrix for the Turkers (GT = ground truth).

	Anger	Disgust	Fear	Happy	Neutral	Sad
Anger (GT)	69.0	14.7	4.6	6.8	3.5	0.7
Disgust (GT)	7.8	32.5	9.4	6.8	28.0	15.0
Fear (GT)	11.2	3.6	67.2	11.3	4.2	2.3
Happy (GT)	3.3	6.3	8.0	54.7	22.9	4.3
Neutral (GT)	0.9	2.1	0.4	1.8	78.4	15.8
Sad (GT)	0.5	3.7	5.6	0.3	25.2	64.3

Table 5.5: Accuracy values (%) for APN and PNN.

	Samples			Classification Confidence	
	All	Female	Male	Confident (80%)	Unsure (20%)
Computer (APN)	89.3	86.8	92.4	94.4	73.1
Turkers (APN)	70.5	71.5	69.0	71.0	67.9
Computer (PNN)	82.9	82.9	82.4	88.0	62.0
Turkers (PNN)	71.8	75.5	66.6	72.1	70.7

to make. Nevertheless, it is interesting to see that the computer system’s accuracy on the rejected samples is very close to that obtained even by confident Turkers, which further shows the superiority of the computer system over naive human coders on this dataset.

Confusion matrices for the automatic emotion classification system’s classification and the Turkers’ classification for the 6 emotions are shown in Tables 5.3 and 5.4, respectively. Note that in these tables, the rows are the ground truth (GT) labels, and they sum to 100%. From these tables, we see that the automatic classification system is classifying each emotion better than the Turkers.

### Classifying into Active-Passive-Neutral

Next, we explore the results when classifying the samples according to the three arousal categories: active, passive and neutral (APN). As can be seen in Table 5.5, the Turkers achieved 70.5% accuracy in their classification of the utterances into active, passive and neutral categories, while the computer system achieved 89.3% accuracy. As for the 6 emotion classification task, the accuracy for the samples for which the Turkers are confident in their classification does not improve significantly compared with the accuracy for all the samples, while the computer system does have an increase in accuracy when only classifying samples for which it is confident in the response.

### Classifying into Positive-Negative-Neutral

For the final classification task, we explore the results when classifying the samples according to the three valence categories: positive, negative and neutral (PNN). As can be seen in Table 5.5, the Turkers achieved 71.8% accuracy in their classification of the utterances into positive, negative and neutral categories, while the computer system achieved 82.9% accuracy. Once again, the same conclusions hold for the confident utterances.

### 5.2.7 Discussions

It is important to note that the expression and perception of emotion are very subjective. For the same utterance, different listeners may perceive different emotions, and all of them may be different from the emotion that the speaker intended to convey. Therefore, for an emotion classification task, obtaining the ground-truth emotion labels is not trivial. To obtain the ground-truth “perceived” emotion, one could ask some listeners to label the utterance, but these labels are ambiguous due to their subjective nature. Our results also show that different Turkers do sometimes disagree with each other.

Due to this difficulty in obtaining ground-truth emotion labels, in our study we used acted emotions. On the one hand, one may criticize that these utterances may not be “natural”. On the other hand, however, the ground-truth labeling is not an issue. Each utterance is labeled to the emotion that the speaker wants to convey, hence the ground-truth labels are “expressed” emotions. Consequently, the classification errors that the Turkers made simply indicate the mismatches between the emotions that the speakers wanted to convey and the emotions that the Turkers perceived, i.e., the effectiveness of the emotion communication through these utterances.

Compared to the automated classification system, emotion communication between humans is apparently less effective according to the results in our study. One of the most important reasons, we argue, is due to the lack of training. The computer system was trained and tested on the same dataset. Although utterances from different speakers were used for training and testing, they did share some common characteristics such as the type of speech content and the recording environment. The Turkers, however, were only provided with 1 sample recording for each emotion of the dataset.

Although the Turkers have experienced numerous samples of these emotions in their daily lives, they are still considered “naive” for this dataset.

While it is feasible to train computer systems for specific types of data (e.g., in a certain environment), it is often not possible to provide similar training to humans and hence they will always be operating in the “naive” mode. Some applications where trained automatic classification systems can replace naive human coders include: 1) warning managers at call centers when either a customer or the customer service representative displays a negative emotion (such as anger, frustration, etc.); 2) applications where there is sensitive data and the content should not be released to human workers due to privacy issues; 3) a vehicular application that warns a driver about negative emotions to avoid road rage; and 4) applications that help those unable to decode emotions accurately, such as those with autism or certain cognitive degeneration diseases. In these systems, it is not required to classify every “sample” (e.g., each 2-3 s of audio); instead, it is important that when an emotion classification label is added to the data, that classification is accurate. As shown in our study, an automatic classification system is able to meet this requirement, providing quite high accuracy values by classifying between 50% and 80% of the data.

One interesting question for future work is how quickly humans would be able to be trained on a particular dataset, and once trained, would they be able to provide accuracy performance similar to the automatic classification system? If humans could be trained quickly, then this would be a feasible option for some applications; however, if the cost (time and resources) to train humans is large, the automatic classification system remains an attractive alternative.

## 5.2.8 Conclusions

This study compares the performance of a speech-based automatic emotion classification system with the performance of naive human coders in classifying emotions for speech utterances. The results show that the automatic system achieves much better accuracy in almost all cases. Additionally, the automatic system can improve the classification accuracy by rejecting to classify samples for which it is not confident in the classification, while the naive human coders were not able to improve their

accuracy through specifying their confidence in their classification. These results show that a speech-based automatic emotion classification system is feasible as a replacement for applications that utilize naive human coders to classify emotion.

## 5.3 WISE: Web-based Interactive Speech Emotion Classification

### 5.3.1 Introduction

Accurately estimating emotions of conversational partners plays a vital role in successful human communication. A social-functional approach to human emotion emphasizes the interpersonal function of emotion for the establishment and maintenance of social relationships [187], [188], [189]. According to [187] “Emotions are not mere feelings, but rather are processes of establishing, maintaining, or disrupting relations between the person and the internal or external environment, when such relations are significant to the individual.” Thus, the expression and recognition of emotions allows the facilitation of social bonds through the conveyance of information about one’s internal state, disposition, intentions, and needs.

In many situations, audio is the only recorded data for a social interaction, and estimating emotions from speech becomes a critical task for psychological analysis. Today’s technology allows for gathering vast amounts of emotional speech data from the web, yet analyzing this content is impractical. This fact prevents many interesting large-scale investigations.

Given the amount of speech data that proliferates, there have been many attempts to create automatic emotion classification systems. However, the performance of these systems is not as high as necessary in many situations. Many potential applications would benefit from automated emotion classification systems, such as call-center monitoring [190, 191], service robot interactions [192, 193] and driver assistance systems [194, 195]. Indeed, there are many automated systems today that focus on speech [172, 196, 171, 170, 197, 198]. However, emotion classification accuracy of fully automated systems is still not satisfactory in many practical situations.

In this study, we propose WISE, a web-based interactive speech emotion classification system.

This system uses a web-based interface that allows users to easily upload a speech file to the server for emotion analysis, without the need for installing any additional software. Once the speech files are uploaded, the system classifies the emotions using a model trained on previously labeled training samples. Each classification is also associated with a confidence value. The user can either accept or correct the classification, to “teach” the system the user’s specific concept of emotions. Over time, the system adapts its emotion classification models to the user’s concept, and can increase its classification accuracy with respect to the user’s concept of emotions.

The key contribution of our work is that we provide an interactive speech-based emotion analysis framework. This framework combines the machine’s computational power with human users’ high emotion classification accuracy. Compared to purely manual labeling, it is much more efficient. Compared to fully automated systems, it is much more accurate. This opens up possibilities for large-scale speech emotion analysis with high accuracy.

The proposed framework only considers offline labeling and returns labels in three categories: emotion, arousal and valence with time codes. To evaluate our system, we have simulated the user-interface interactions in several settings, by providing ground truth labels on behalf of the user. One of the scenarios is designed to be a baseline, with which we can compare the remaining scenarios. In another scenario, we test if the system can adapt to the samples whose speaker is unknown to the system. The next scenario tests how the system’s classification confidence of a sample affects the system’s accuracy. The full system is available for researchers to use. <sup>1</sup>

### 5.3.2 Related Work

All-in-one frameworks for automatic emotion classification from speech, such as EmoVoice [199] and OpenEar [200], are standalone software packages with various capabilities, including audio recording, audio file reading, feature extraction, and emotion classification.

EmoVoice allows the user to create a personal speech-based emotion recognizer, and it can track the emotional state of the user in real-time. Each user records their own speech emotion corpus to train the system, and the system can then be used for real-time emotion classification for the same

---

<sup>1</sup><http://www.ece.rochester.edu/projects/wcng>



user. The system outputs the x- and y-coordinates of an arousal-valance coordinate system with time codes. It is reported in [199] that EmoVoice has been used in several systems including humanoid robot-human and virtual agent-human interactions. EmoVoice does not consider user feedback once the classifier is trained, whereas in our system, the user can continually train and improve the system.

OpenEar is an emotion classification multi-platform software package that includes libraries for feature extraction written in C++ and pre-trained models as well as scripts to support model building. One of its main modules is named SMILE (Speech and Music Interpretation by Large-Space Extraction), and it can extract more than 500K features in real-time. The other main module allows external classifiers and libraries such as LibSVM [201] to be integrated and used in classification. OpenEar also supports popular machine learning frameworks' data formats, such as the Hidden Markov Model Toolkit (HTK) [202], WEKA [203], and scikit-learn for Python [204], and therefore allows easy transition between frameworks. OpenEar's capability of batch processing, combined with its advantage in transitioning to other learning frameworks, makes it appealing for large databases.

ANNEMO (ANNotating EMOtions) [205] is a web-based annotation tool that allows labeling arousal, valence and social dimensions in audio-visual data. The states are represented between -1 and 1, where the user changes the values using a slider. The social dimension is represented by categories rather than numerical values, and those are agreement, dominance, engagement, performance and rapport. No automatic classification/labeling modules are included in ANNEMO.

In contrast, WISE is a web-based system and can be used easily without installing any software, unlike EmoVoice and OpenEar. WISE is similar to ANNEMO in terms of the web-based labeling aspect, however WISE only considers audio data and provides automatic classification as well.

### 5.3.3 Web-based Interaction

Our system's interface, shown in Figure 5.3, is web-based, allowing easy, secure access and use without installing any other software except a modern browser.

When a user uploads an audio file, the waveform appears on the main screen, allowing the user to select different parts of the waveform. Selected parts can be played and labeled independently. These

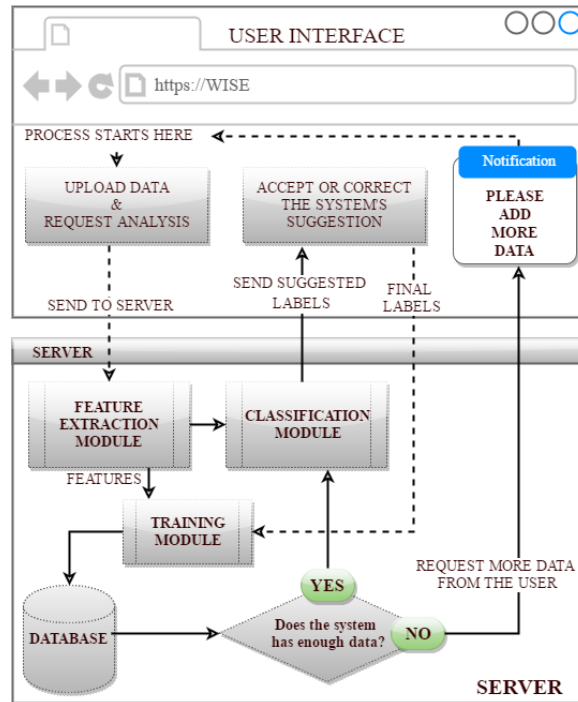


Figure 5.2: Flow chart showing the operation of WISE.

selected parts will also be added to a list, as shown in the bottom-left side of Figure 5.3. The user can download this list by clicking on the “save” button in the interface.

The labeling scheme is restricted to three categories: emotion, arousal and valence. Emotion category elements are anger, disgust, fear, happy, neutral, sadness. Arousal category elements are active, passive and neutral, and valence category elements are positive, negative and neutral. Our future work includes adding user defined emotion labels into the system.

The user can request labels from the automated emotion classifier by clicking on the “request label” button. The system then shows suggested labels to the user.

The next section describes the automated speech-based emotion classification system used in WISE.

### 5.3.4 Automated Emotion Classification System

There are various automated speech-based emotion classification systems [172, 196, 171, 170, 197] that consider different features, feature selection methods, classifiers and decision mechanisms.

The screenshot displays the WISE (Web-based Interactive Speech Emotion Classification System) interface. At the top, there is a navigation bar with 'Welcome at Home About Contact' and a 'Logout' link. The main area features a large blue grid background with a white waveform of a speech sample. Below the waveform are playback controls: Play, Stop, Loop, Remove, and Save. A file name '31957908-31957-0006.wav' is shown in a dropdown menu, with 'Upload file' and 'Delete File' buttons. A 'Suggested labels: Disgust, Active, Negative' box is present, along with 'Accept suggestion and add to training' and 'Discard suggestion' buttons. On the left, a dark sidebar contains a 'console' section with a message 'You have connected to the server.', a 'models' section with an 'LDC' dropdown and volume controls, and several action buttons: 'Inspect model', 'Add to training', 'Update model', and 'Request Analysis'. At the bottom, a 'Segments' table is visible, showing three segments with their respective start and end times, emotion labels, arousal and valence levels, and loop status.

ID	Start	End	Emotion	Arousal	Valence	Loop
1	0.30	1.80	Disgust	Active	Negative	false
2	2.10	3.26	Disgust	Active	Negative	false
3	3.68	5.25	Happy	Active	Positive	false

Figure 5.3: WISE user interface screenshot.

Our system is based on [198], which provides a confidence value along with the classification label.

## Features

Speech samples are divided into overlapping frames for feature extraction. The window and hop sizes are set to 60 ms and 10 ms, respectively. For every frame that contains speech, the following features are calculated: fundamental frequency ( $F_0$ ), 12 mel-frequency cepstral coefficients (MFCCs), energy, frequency and bandwidth of first four formants, zero-crossing rate, spectral roll-off, brightness, centroid, spread, skewness, kurtosis, flatness, entropy, roughness, and irregularity, in addition to the derivatives of these features. Statistical values such as minimum, maximum, mean, standard deviation and range (i.e., max-min) are calculated from all frames within the sample. Additionally, speaking rate is calculated over the entire sample. Hence, the final feature vector length is 331.

## Feature Selection

The system employs the support vector machine (SVM) recursive feature elimination method [186]. This approach takes advantage of SVM weights to detect which features are better than others. After the SVM is trained, the features are ranked according to the order of their weights. The last ranked feature is eliminated from the list and the process starts again, until there are no features left. Features are ranked in reverse order of elimination order. The top 80 best features are chosen to be used in the classification system. Note that in Section 5.3.5, the features are selected beforehand and not updated when a new sample is added to the system.

## Classifier

Our system uses a one-against-all (OAA) binary SVM with radial basis function (RBF) for each emotion, arousal and valence category element, for a total of 12 SVMs. The trained SVMs calculate confidence scores for any sample that is being classified. The system labels the sample with the class of the binary classifier with maximum classification confidence on the considered sample.

### 5.3.5 Evaluation

To evaluate WISE and the benefit of user-assisted labeling of the data, we have simulated user-interface interactions using the LDC database as the source of data for training, validation and testing.

#### Dataset

We use the Linguistic Data Consortium (LDC) Emotional Prosody Speech and Transcripts [173] database in our simulations. The LDC database contains samples from 15 emotion categories; however, in our evaluation, we only use 6 of the emotions as listed in Section 5.3.3. The LDC database contains acted speech, voiced by 7 professionals, 4 female and 3 male. The transcripts are in English and contain semantically neutral utterances, such as dates and times.

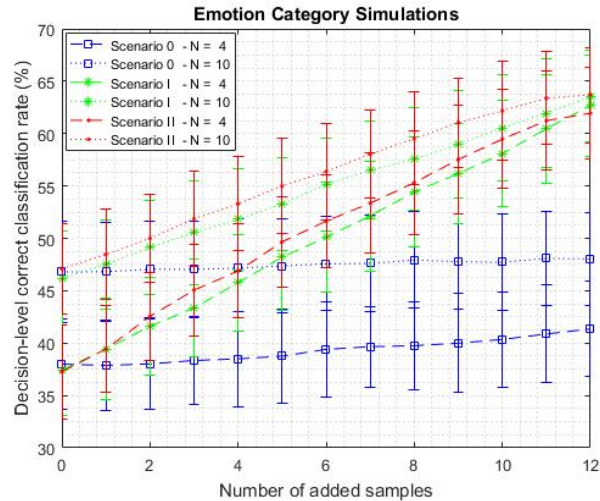


Figure 5.4: The results of emotion category for Scenarios I-III.

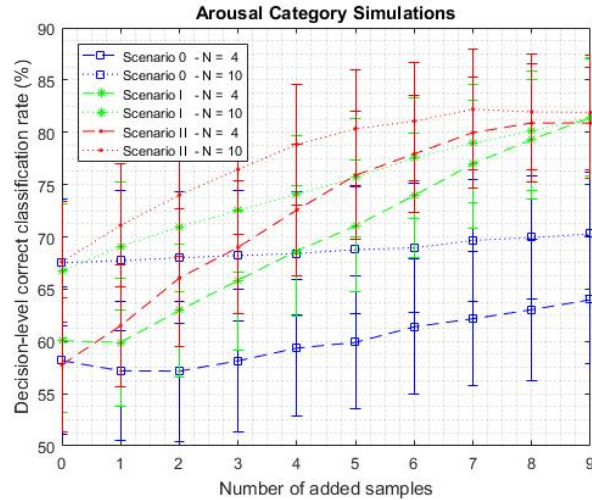


Figure 5.5: The results of arousal category for Scenarios I-III.

## Simulations

We have simulated user-interface interactions in different scenarios for which WISE can be used to enable user feedback to improve classification accuracy. In these simulations, there are three data groups: training, test and validation. We assume that validation data represents the samples where the user provides the “correct” label. In each iteration, the system evaluates the test data using the current models, and at the end of each iteration, a sample from the validation data is added to the training data to update the models. Next, we describe the different scenarios in detail.

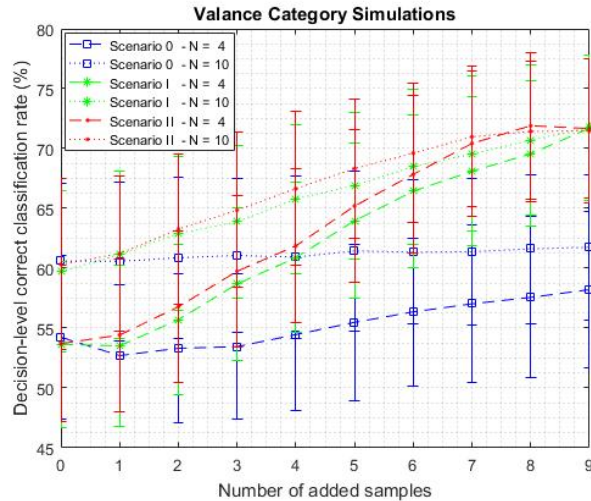


Figure 5.6: The results of valance category for Scenarios I-III.

### Scenario 0 - Baseline

In this scenario, the data from 1 of the 7 speakers is used for testing, while the remaining 6 speakers' data are used for training and validation. Since only a limited amount of data is available from each speaker in the next scenarios, we also limit the amount of the validation data in this scenario. In this way, the baseline becomes more comparable to the other scenarios.

The training data starts with  $N$  samples from each class for each category. For the emotion classification, there are only 2 samples available in each class (emotion) for the validation data. However, the arousal and valance categories have half the number of classes that the emotion category has, therefore, there are 3 samples available in each class that can be used in validation data for these categories. After the data are chosen randomly, the system simulates the interaction process. This process is repeated for all speakers, and the results are averaged over all 7 speakers and 200 trials.

### Scenario I

This scenario has the same settings as Scenario 0, except this time, the testing data, as well as the validation data are chosen from a speaker, and the training data is chosen among the remaining 6 speakers' data.

## Scenario II

This scenario has the same settings as Scenario I with a single difference: in each round, the validation data has been ordered in ascending order of the classifier's confidence level in classifying them. Therefore in each iteration, the sample, on which the system has the least confidence, is added to the training data from the validation data.

## Discussion

Figures 5.4-5.6 show the classification accuracy versus the number of added samples for each scenario for the emotion, arousal and valence, respectively. Note that the error bars represent the standard deviation of the results over the 7 speakers and 200 trials.

Scenario I shows the ability of WISE to enable adaptation of the models. In many situations, trained models of automatic systems have no information on the speaker to be classified. The comparison of classification accuracy between Scenario 0 and Scenario I shows that adaptation to unknown data is vital for accurate emotion estimation, as the accuracy increases greatly when data from the new user are added.

For example, in Scenarios I and II, when  $N$  is 4 for the emotion category, the system's initial accuracy starts around 37% and increases to approximately 63%, as can be seen in Figure 5.4, where on the other hand in Scenario 0, accuracy can only increase to approximately 41%. In Scenarios I and II, when  $N$  is 10, the classification accuracy starts higher than the previous case, yet with the same number of added samples, they converge to the same percentage. This enables the possibility of using pre-trained models in our system that are trained on available databases.

The results of Scenario II suggest that adding the samples with low classification confidence are slightly more beneficial than adding a sample for which the system already has more confidence. Figures 5.4-5.6 show that the classifier in Scenario II converges to a slightly higher classification accuracy than the one in Scenario I. This can be seen especially in the arousal category results.

### 5.3.6 Conclusions

This study introduced and evaluated the WISE system, which is an interactive web-based emotion analysis framework to assist in the classification of human emotion from voice data. The full system is available for the community to use. The evaluation results show that the system can adapt to the user's choices and can increase the future classification accuracy when the speaker of the sample is unknown. Hence, WISE will enable adaptive, large scale emotion classification.

## 5.4 Unsupervised Learning Approach to Feature Analysis for Automatic Speech Emotion Recognition

### 5.4.1 Introduction

Emotions are a vital part of social interactions. Designing computational models to recognize emotions is key to an automatic understanding of social interactions. In recent years, researchers have developed automatic emotion recognition systems using different data modalities, including physiological signals [206], facial expressions and body gestures [207], and speech [208]. Among these modalities, speech is more accessible and less intrusive in daily life. Therefore, automatic speech emotion recognition (ASER) has received much attention in this field.

ASER is a challenging task. While automatic systems have been shown to outperform naive human listeners on speech emotion classification [27], unlike speech and image classification tasks, current ASER systems are still not competitive to trained human listeners. One bottleneck for improving ASER is the lack of training data. Recording and annotating emotional speech is a very time-consuming process. Compared to general speech datasets, publicly available speech emotion recognition datasets are much more limited in the number of speakers and utterances, and the coverage of vocabulary and recording conditions [208].

One way to alleviate the data lacking issue is to transfer knowledge learned from unlabeled data or data in other related tasks (source tasks) to the task at hand (target task) [209]. One technique is unsupervised feature learning, which does not utilize the label information but aims to learn robust



features that can capture the intrinsic structures of the data. These features are also often discriminative to train better classification models for the target task [210, 211]. For ASER, the most natural and available data sources are general speech. They may not carry strong emotions, but features learned from these data may capture intrinsic structures of speech and be useful for ASER.

Unsupervised feature learning has been rarely explored in ASER beyond autoencoders (AE) [212] and denoising autoencoders (DAE) [210]. AE and DAE aim to learn features that are good for the reconstruction of the input. More advanced techniques, such as variational autoencoders (VAE) [213] and generative adversarial networks (GAN) [96], do not aim to reconstruct the input, but aim to *generate* data that come from the same distribution as the input. This relaxation tends to put more emphasis on the modeling of intrinsic structures of the data during feature learning [213, 96, 211].

In this chapter, we describe our design of a convolutional neural network (CNN)-based ASER system and make the first systematic exploration of various kinds of unsupervised learning techniques to improve the speaker-independent emotion recognition accuracy. These techniques include the denoising autoencoder (DAE), variational autoencoder (VAE), adversarial autoencoder (AAE) and adversarial variational Bayes (AVB). We compare these systems with two baselines (SVM and CNN) that work on hand-crafted features without unsupervised feature learning. Experiments show that unsupervised feature learning significantly improves the ASER performance, when trained on a large scale general speech dataset, regarding unweighted accuracy rating (UAR) and F1-score. Furthermore, the latent variable models including VAE, AAE, and AVB improve the ASER performance more than the DAE and other baselines. This suggests that unsupervised learning with these latent variable models are useful practices for ASER, where training data is insufficient.

## 5.4.2 Related Work

Traditional ASER systems that utilize Gaussian mixture models (GMMs) [214, 171, 215], hidden Markov models (HMMs) [216, 180], and support vector machines (SVMs) [217, 170, 218], rely on well-established hand-crafted speech features. These features usually include spectral, cepstral, pitch, and energy features of the speech signal at the frame level. Statistical functionals of these features are

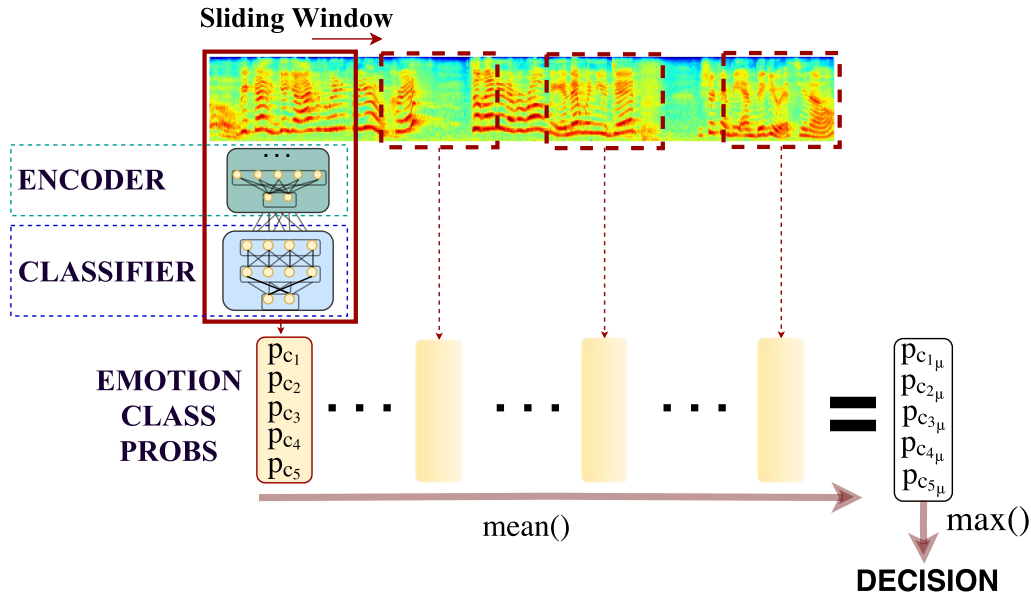


Figure 5.7: Proposed ASER system overview. The dashed red windows represent the sliding window with 50% overlap. From each window, emotion class probabilities ( $p_1, p_2, p_3, p_4$  and  $p_5$ ) are predicted and the average of these vectors is calculated over all windows is calculated for each utterance. The emotion that has the highest probability is predicted as the emotion of the utterance.

then applied across multiple frames to obtain an utterance-level feature vector.

Some researchers explored deep learning methods to find robust features for the ASER task. Xia et al. [219] proposed a modified DAE that maps input speech to two hidden representations, a neutral representation learned by reconstructing neutral speech beforehand, and an emotional representation learned by reconstructing emotional speech with the neutral representation fixed. During testing, the emotional representation of a test speech sample is fed to an SVM classifier for emotion classification. In their follow-up work [220], Xia et al. incorporated the speaker gender information which resulted in further improvements.

Ghosh et al. [221, 222] trained stacked DAEs and a bidirectional long short-term memory (BLSTM) AE to obtain a latent representation of the input spectrogram extracted from the speech and the glottal flow waveform. These latent representations were then fed to a multilayer perceptron (MLP) with a softmax output for 4-class emotion classification.

Deng et al. [223] proposed a single-layer sparse autoencoder (SAE) for feature transfer learning between different emotion corpora. One SAE was trained for each emotion class in the source do-

main using hand-crafted features as the input. Then each training sample in the target domain was reconstructed by the SAE of the corresponding class. Finally, an SVM model was trained on the reconstructed data to classify the original test samples without going through the SAEs. Deng et al. [224] obtained further improvements by replacing the SAEs with denoising autoencoders (DAEs).

Although these studies have demonstrated the benefits of unsupervised feature learning using DAEs, more advanced latent variable methods such as VAE, AAE, and AVB have not been explored for ASER. These methods attempt to model the distribution of data and are likely to learn more meaningful, controllable and discriminative features, leading to better classification performance, especially when the amount of labeled data is small [211].

### 5.4.3 Method

We propose to adopt a convolutional neural network (CNN)-based architecture (shown in Fig. 5.7) for ASER and to investigate the effects of different unsupervised learning techniques. Specifically, the network contains a pre-trained encoder network to extract features from the log-Mel spectrogram of the input speech, and a fully connected (FC) network to classify their emotions. The encoder includes three convolutional layers with a leaky rectified linear unit (LReLU) activation and an FC layer with a linear activation as shown in Table 5.6. The encoder gradually reduces the dimension of the input into the latent dimension. During classification, the encoder network weights are frozen. The classifier consists of three fully connected layers with LReLU activations except for the last activation, which uses softmax to represent probabilities of each emotion class. There are two dropout layers with 0.25 drop rate between FC layers. The categorical cross-entropy loss is used during the training of the FC.

The proposed network processes each utterance by segments that are 1 second long. During training, we randomly choose patches to form training batches from each utterance and use the utterance-level label as the label for the segment. During testing, we segment each utterance into 1-second long segments with a 0.5-second overlap. We predict the emotion probabilities in each segment and then average the probabilities across all segments. We finally choose the emotion category, which has the highest mean probability, as the utterance-level emotion classification result.

Net	Layers	Activ.	F. No	F. Size	Strides	Output Shape
encoder ( $q_\theta$ )	Input (x)	-	-	-	-	$64 \times 64 \times 1$
	Conv2D	LReLU	32	$9 \times 9$	$2 \times 2$	$32 \times 32 \times 32$
	Conv2D	LReLU	64	$7 \times 7$	$2 \times 2$	$16 \times 16 \times 64$
	Conv2D	LReLU	128	$5 \times 5$	$2 \times 2$	$8 \times 8 \times 128$
	Flatten	-	-	-	-	8192
	FC	Linear	-	-	-	256
decoder ( $p_\phi$ )	Input (z)	-	-	-	-	256
	FC	LReLU	-	-	-	8192
	Reshape	-	-	-	-	$8 \times 8 \times 128$
	Conv2DT	LReLU	128	$5 \times 5$	$2 \times 2$	$16 \times 16 \times 128$
	Conv2DT	LReLU	64	$7 \times 7$	$2 \times 2$	$32 \times 32 \times 64$
	Conv2DT	LReLU	32	$9 \times 9$	$2 \times 2$	$64 \times 64 \times 32$
	Conv2D	Sigmoid	1	$1 \times 1$	$1 \times 1$	$64 \times 64 \times 1$
AAE discriminator	Input (z)	-	-	-	-	256
	FC	LReLU	-	-	-	2048
	FC	LReLU	-	-	-	2048
	FC	LReLU	-	-	-	2048
	FC	Sigmoid	-	-	-	1
AVB discriminator	Input (z)	-	-	-	-	256
	FC	LReLU	-	-	-	4096
	Reshape	-	-	-	-	$64 \times 64 \times 1$
	Input (x)	-	-	-	-	$64 \times 64 \times 1$
	Concat	-	-	-	-	$64 \times 64 \times 2$
	Conv2D	LReLU	32	$9 \times 9$	$2 \times 2$	$32 \times 32 \times 32$
	Conv2D	LReLU	64	$7 \times 7$	$2 \times 2$	$16 \times 16 \times 64$
	Conv2D	LReLU	128	$5 \times 5$	$2 \times 2$	$8 \times 8 \times 128$
	Flatten	-	-	-	-	8192
	FC	LReLU	-	-	-	256
FC	Sigmoid	-	-	-	1	
classifier	Input (z)	-	-	-	-	256
	FC	LReLU	-	-	-	1024
	Dropout	-	-	-	-	1024
	FC	LReLU	-	-	-	1024
	Dropout	-	-	-	-	1024
	FC	Softmax	-	-	-	5

Table 5.6: The architecture of the encoder, decoder, discriminator and emotion classifier networks. AEs share the encoder and decoder structures, except AVB where we modify the encoder to accept external noise input similar to AVB discriminator architecture. *Conv2D* is a 2-d convolution layer, where *Conv2DT* is a transposed 2-d convolution (or deconvolution) layer. *Concat* is the concatenation layer. *F. No* is the number of filters, where *F. Size* is the filter size.

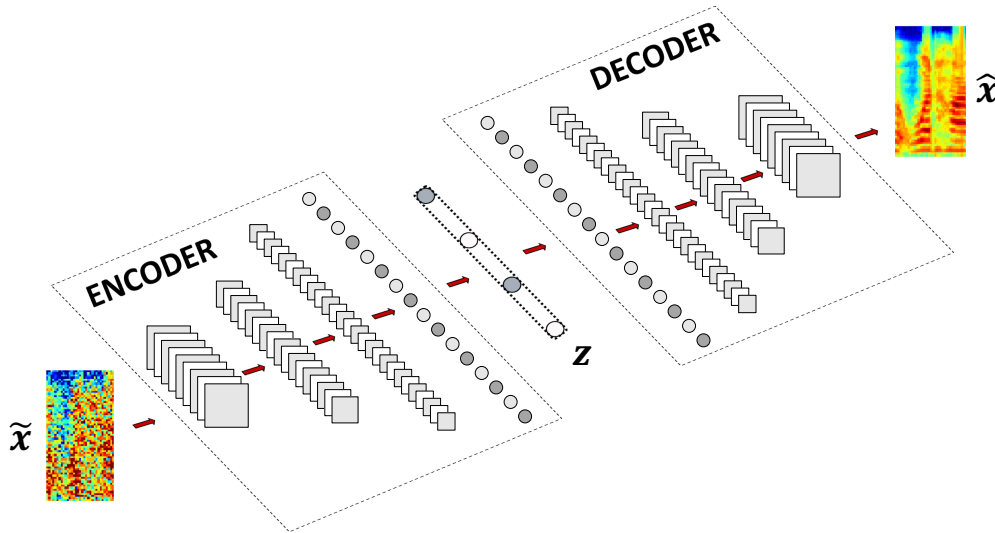


Figure 5.8: DAE network architecture: reconstructing the clean spectrogram from noisy input

In the following, we describe different architectures, and inference models for the encoder explored in this work, including denoising autoencoder (DAE), variational autoencoder (VAE), adversarial autoencoder (AAE) and Adversarial Variational Bayes (AVB).

### Denoising Autoencoder (DAE)

Denoising autoencoders (DAEs) [210] aim to extract robust features by reconstructing clean data from their corrupted versions. They have been applied to ASER systems [219, 220, 221, 222] and yielded performance increase. The model can be expressed as:

$$z \sim q_{\theta}(z|\tilde{x}), \quad (5.1)$$

$$\hat{x} \sim p_{\phi}(x|z), \quad (5.2)$$

where  $z$ ,  $x$ ,  $\tilde{x}$  and  $\hat{x}$  are the latent representation, clean data, corrupted data and reconstructed clean data, respectively.  $q_{\theta}$  and  $p_{\phi}$  are the probabilistic notation of the encoder and decoder networks, where  $\theta$  and  $\phi$  are the trainable parameters of the networks. When cross-entropy is used to measure

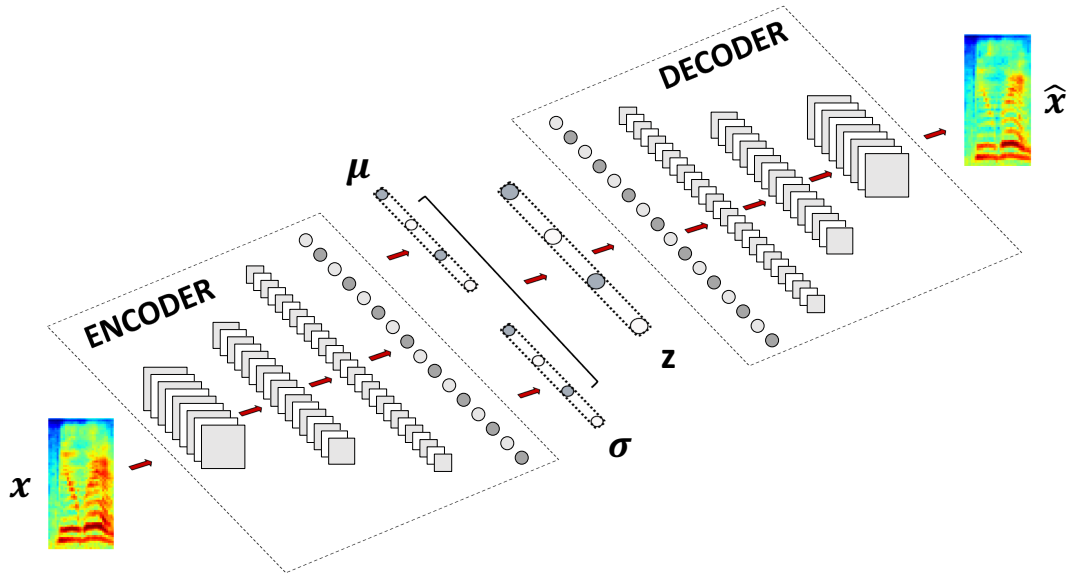


Figure 5.9: VAE network architecture: variational inference on auto-encoder by constraining the latent representation to follow a normal distribution

the reconstruction error, the loss function is defined as:

$$\min_{\theta, \phi} -\mathbb{E}_{z \sim q_{\theta}(z|\hat{x})} [\log p_{\phi}(x|z)]. \quad (5.3)$$

As we do not have an estimation nor control of the distribution of the latent representation, it is difficult to generate new but realistic data using the decoder of DAEs.

We train a DAE using the same encoder-decoder architecture as shown in Table 5.6. The encoder and decoder networks are symmetrical except for the last layer of the decoder network.

### Variational Autoencoder (VAE)

VAE [213] is another version of AE that performs variational inference by constraining the latent representation to match an explicit distribution such as a normal distribution. The latent representation is defined as follows:

$$(z_{\mu}, z_{\sigma}) \sim q_{\theta}(z_{\mu}, z_{\sigma}|x), \quad (5.4)$$

$$z = z_{\mu} + z_{\sigma} \odot \mathcal{N}(0, I), \quad (5.5)$$

where  $z_\mu, z_\sigma$  are the mean and standard deviation obtained from the encoder network, and  $\mathcal{N}(0, I)$  is the Gaussian distribution with zero mean and unit standard deviation. The loss function is defined as

$$\min_{\theta, \phi} KL(q_\theta(z|x)||p(z)) - \mathbb{E}_{q_\theta(z|x)}[\log p_\phi(x|z)], \quad (5.6)$$

where  $p(z) = \mathcal{N}(z; 0, I)$  is the prior multivariate Gaussian distribution that we want latent representation to match and  $KL$  is the *Kullback-Leibler* (KL) divergence respectively. The first term regularizes the output latent distribution of the encoder and the second term is the reconstruction loss of AE. Since the latent representation distribution is controlled, new but realistic samples can be easily generated by feeding to the decoder the randomly drawn latent representations according to the normal distribution.

We train a VAE using the same architecture as the encoder-decoder shown in Table 5.6 except that we modify the encoder network by replacing the last layer with two fully connected layers, which output  $z_\mu$  and  $z_\sigma$ . We calculate the latent representation  $z$  using Eq. (5.4), and feed it to the decoder network.

### Adversarial Autoencoder (AAE)

Generative adversarial networks (GANs) have achieved remarkable success in generating realistic data [96]. GANs are zero-sum two player game where the players are the counterfeiter and the police. The counterfeiter forges a fake sample and presents it to the police, and the police try to distinguish between real and fake samples. In neural network terminology, the counterfeiter is called the generator network and the police is called the discriminator network.

Adversarial autoencoders (AAEs) [225] are a type of AE that performs variational inference by constraining the latent distribution to match a specified distribution  $p(z)$  through adversarial training. In GAN terms, the encoder  $q_\theta(z|x)$  tries to fool the discriminator by generating latent codes that

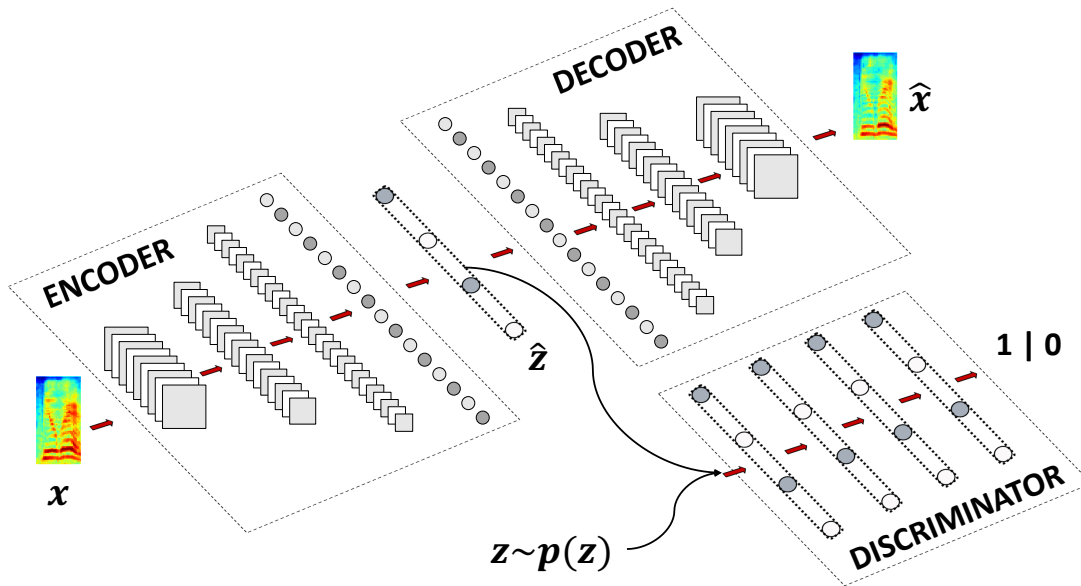


Figure 5.10: AAE network architecture: variational inference on auto-encoder by constraining the latent representation through adversarial training

mimic  $p(z)$ . The min-max game can be expressed as:

$$\begin{aligned} \min_{\theta, \phi} \max_{\psi} \mathbb{E}_{z \sim p(z)} [\log D_{\psi}(z)] + \\ \mathbb{E}_{x \sim p_{data}} [\log(1 - D_{\psi}(q_{\theta}(z|x)))] - \\ \mathbb{E}_{z \sim q_{\theta}(z|x)} [\log p_{\phi}(x|z)], \end{aligned} \quad (5.7)$$

where  $D_{\psi}(\cdot)$  is the discriminator, and  $\psi$  is its parameter. The first two terms are the GAN loss involving the encoder and the discriminator, while the third term is the reconstruction loss involving the encoder and the decoder. AAEs rely on reconstruction loss to capture the data distribution where adversarial loss acts as a regularization term over latent distribution to match the prior distribution.

We use the same architecture that is used for the other AEs for the encoder and decoder networks. We add a discriminator network shown in Table 5.6 to distinguish between real and fake latent codes.

### Adversarial Variational Bayes (AVB)

AVB is a training technique for VAEs that replaces the KL term with an adversarial loss [226]. The discriminator inputs are pairs of  $(x, z)$  where  $x$  is sampled from the real data distribution and  $z$  is either sampled from the prior distribution or obtained from the inference model. The discriminator



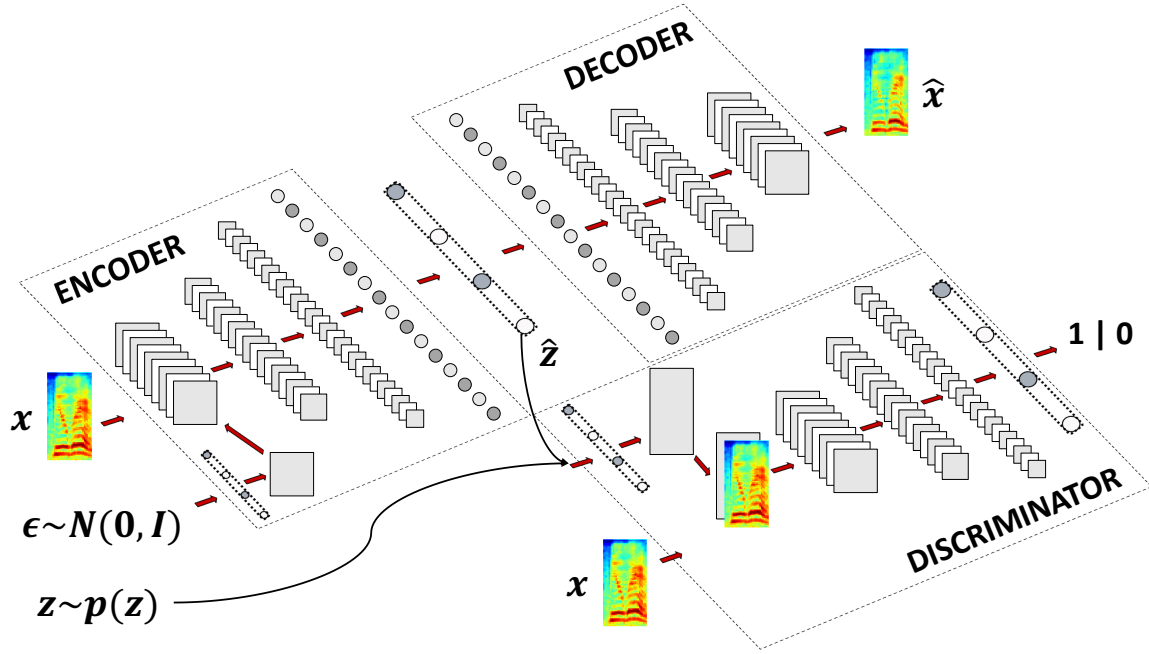


Figure 5.11: AVB network architecture: unifying VAE and generative adversarial networks (GANs)

tries to distinguish whether the pairs are sampled from the prior distribution or the inference model.

The encoder-decoder model parameters are updated with Eq. (5.8) where the discriminator parameters are updated with Eq. (5.9).

$$\min_{\phi, \theta} \mathbb{E}_{x \sim p_{data}} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} [D_{\psi}(x, q_{\theta}(z|x, \epsilon))] - \mathbb{E}_{z \sim q_{\theta}(z|x, \epsilon)} [\log p_{\phi}(x|z)], \quad (5.8)$$

$$\max_{\psi} \mathbb{E}_{x \sim p_{data}} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} [\log D_{\psi}(x, q_{\theta}(z|x, \epsilon))] + \mathbb{E}_{x \sim p_{data}} \mathbb{E}_{z \sim p(z)} [\log(1 - D_{\psi}(x, z))], \quad (5.9)$$

We modify the discriminator to accept both the data and latent code. The latent code dimensionality is increased by an FC layer than added to the data as a second channel. The architecture is shown in Table 5.6. We modify the encoder network to accept external noise  $\epsilon \sim \mathcal{N}(0, I)$ ; we follow the same steps described for the discriminator network to merge  $\epsilon$  into the data as a second channel.

## 5.4.4 Experiments

### The Data

In our experiments we use USC-IEMOCAP audio-visual dataset [227] that contains scripted and improvised interactions between actors, we only use the audio files. There are five sessions totaling about 12 hours of data, where each session includes interactions between a female and a male. There are three annotators, where annotations include both categorical and real-valued. Categorical emotions include anger, disgust, excitement, fear, frustration, happiness, neutral, sadness and surprise. We only considered categorical annotations that are agreed by at least two annotators. This database is commonly used in the ASER literature [220, 215, 222].

While most existing work on this database considered only four emotion categories, we consider five, which are anger (972 samples), excited (948), frustration (1670), neutral (1507) and sadness (1039). In all of our experiments, we apply leave-one-session-out cross-validation, where for each rotation we train on four sessions (from eight speakers) and test on the other session (from the other two speakers). This assures that the evaluation is speaker-independent. To tune hyperparameters and decide early stopping, we reserve 20% of training data as the validation set for each rotation.

### The Baseline Models

We use the SVM based ASER system described in [27] as one of the baseline models. We extract frame-level features that include 13 Mel-frequency cepstral coefficients (MFCCs), first four formant frequencies and bandwidths, zero-crossing rate (ZCR), fundamental frequency ( $F_0$ ), root-mean-square (RMS) energy and their first and second-time derivatives, totaling 72 features per frame. We apply mean, std, min, max, and range functionals to frame-level features to obtain utterance-level features, which have a dimensionality of  $72 \times 5 = 360$ . We normalize each dimension of the utterance-level features of the entire training samples to the range between 0 and 1; we normalize the test data using the same scaling factor. We then train a one-against-all binary SVM for each emotion category, with a radial-basis function kernel. During testing, we calculate the probabilities for each class and select the maximum one as the final emotion class for each test sample.

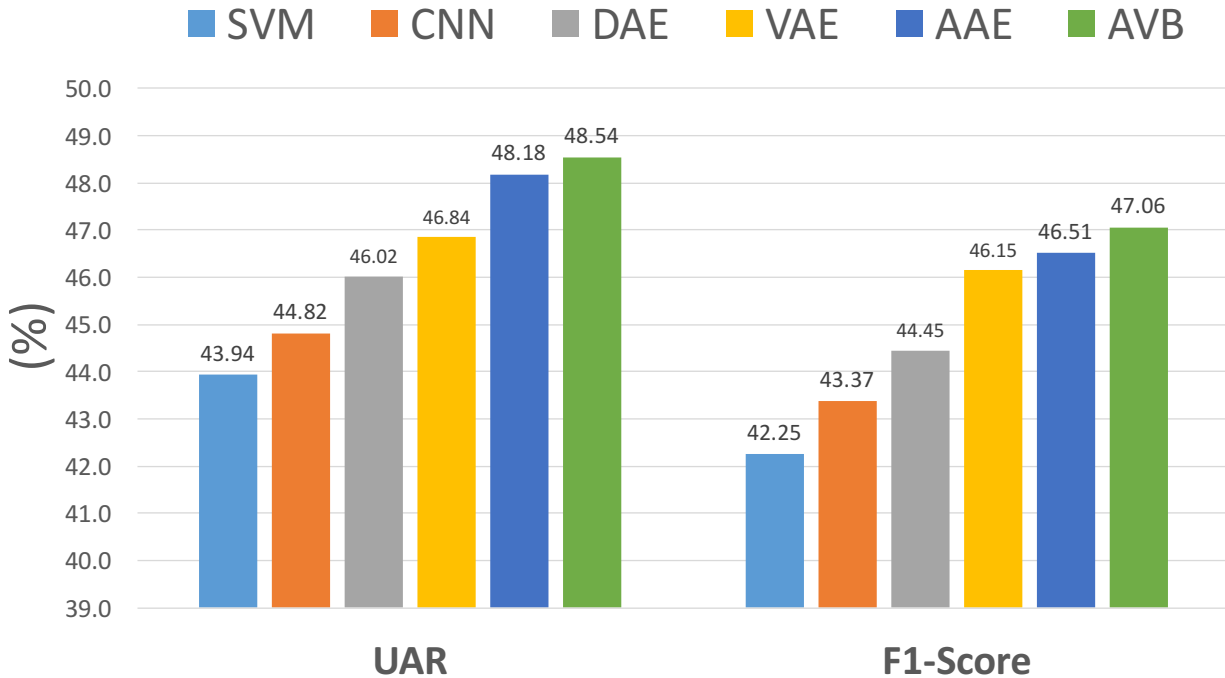


Figure 5.12: The unweighted accuracy rating (UAR) and F1-score results for the baseline systems and the proposed systems. F1-score is calculated for each class, and their unweighted mean is presented.

We design another CNN-FC network as our second baseline system. It takes the same hand-crafted features used in the SVM baseline with a temporal length of 64 (approximately 1 second) as inputs to the CNN encoder network. The CNN output is then fed to an FC network for classification. The architectures for the CNN encoder and the classifier are shown in Table 5.6. Note that the input dimension of the encoder is different, which is  $64 \times 72 \times 1$ . We train this network with Adam optimizer and 0.0002 learning rate. We adopted early stopping criteria, where the training stops if the validation loss is not improved for four epochs.

For the third baseline, we construct another CNN-FC network to take the log-Mel spectrogram directly as input, the same as the proposed four networks. This is to directly test the benefit of the adopted four unsupervised feature learning methods. For this purpose, we use the CNN encoder and FC classifier shown in Table 5.6 and train them from scratch. The resulting system, however, yielded very poor results, close to the chance performance. Therefore, we do not include it in Figure 5.12. We believe that the poor results were due to the scarcity of the training data (only 6136 samples) and the complexity of the CNN network taking log-Mel spectrogram inputs.

## Proposed Models

The AEs presented in Section 5.4.3 are trained by an Adam optimizer with a learning rate of 0.0002. As for the training dataset, we select the Librispeech automatic speech recognition (ASR) corpus [64], which contains read speech that is often emotionally neutral. We calculate a 64-bin log-Mel spectrogram for each utterance with a 32 ms window size and a 16 ms hop size. We normalize the spectrogram values between 0 and 1 per utterance. We form training batches with a size of 256, by selecting random segments with a temporal length of 64 (approximately 1 seconds) from the utterances. The AEs are trained for 200 epochs.

The proposed ASER systems described in Section 5.4.3 are trained with the four pre-trained inference models (encoders), whose parameters are frozen, by an Adam optimizer with a learning rate of 0.001. We adopt an early stopping criterion, where training ends if the validation loss is not improved for four epochs. The emotion models are trained up to 50 epochs. The number of samples in each training batch is set to 256.

## Results

We report the unweighted accuracy ratings (UARs) and F1-score in Figure 5.12 for the SVM and CNN baselines and the proposed systems. Several interesting observations are made. First, the CNN baseline yields slightly better UAR and F1-score than the SVM method. This suggests that deep models, taking the same hand-crafted features as inputs, outperform shallow models. Second, for both metrics, we are able to verify that the DAE-based unsupervised feature learning method using an external emotion-neutral dataset improves the ASER performance over SVM and CNN baselines that do not have the unsupervised feature learning module. This suggests that the learned features from the external emotion-neutral dataset are better than hand-crafted features (SVM baseline) and deep features learned only on the emotion dataset (CNN baseline). Third, the latent variable models VAE, AAE, and AVB outperform the DAE model in terms of both metrics, although they learn features from the same external dataset. This suggests that the latent variable models capture the more discriminative inherent structures of speech data than the reconstruction models such as the DAE.

Fourth, adversarial models AAE and AVB achieve the best result, showing the importance of GAN loss on feature learning. In particular, AVB, which defines the GAN loss on input-code pairs, behaves the best.

### 5.4.5 Conclusions

In this work, we systematically explored the unsupervised methods in the context of ASER. We utilize unsupervised methods namely, DAE, VAE, AAE, AVB and trained on general speech, and use the learned features for ASER task. We show that these methods yield UAR and F1-score increase over the SVM and CNN baselines. Furthermore, we demonstrated that the inference models VAE, AAE, and AVB, outperform the reconstruction model DAE for unsupervised feature learning for ASER.

## Chapter-6

# Generating Emotionally Expressive Talking Faces

In this chapter, we propose a system that can generate emotionally expressive talking faces by merging the components described in previous chapters.

## 6.1 System Overview

We employ the system described in Chapter 4 with a few modifications: 1) we add a speech emotion recognition module, and 2) utilize an emotion discriminator to generate emotional expressions. The overall system is shown in Figure 6.1.

## 6.2 Speech Emotion Recognition Module

This module classifies speech features into emotion classes. The intuition behind this module is to drive speech features to capture the emotion information, which will help generate emotional expressions. The module contains two long short-term memory (LSTM) layers followed by a fully connected layer that outputs the emotion probabilities. The architecture of the speech emotion recognition module is shown in Figure 6.2.

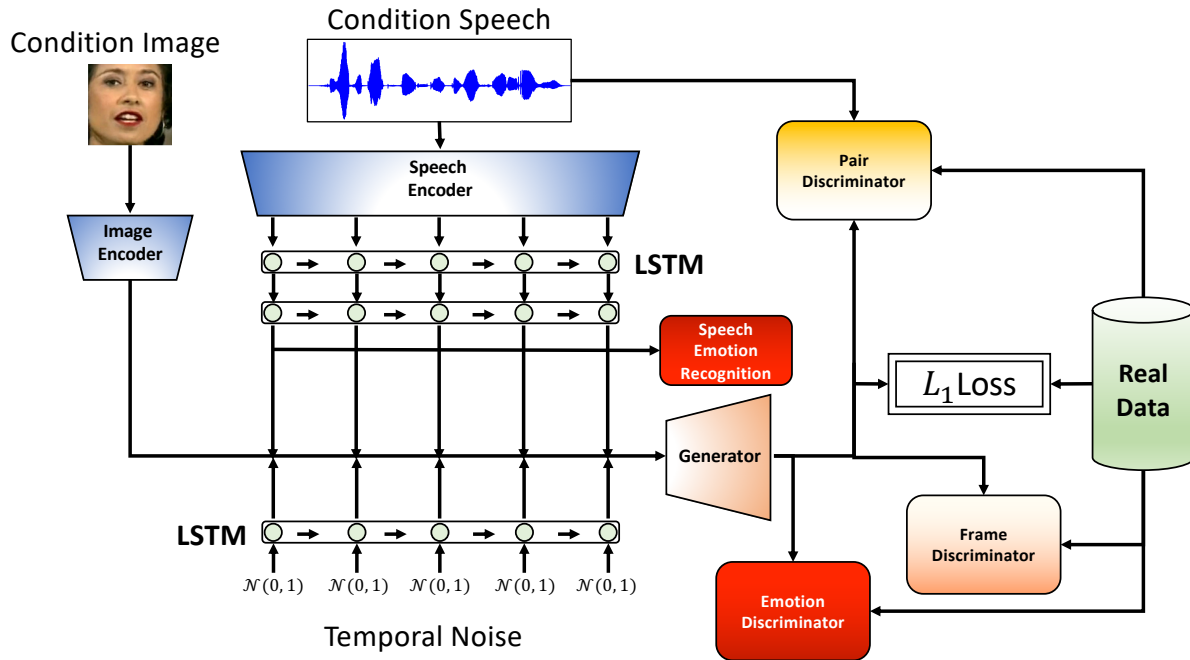


Figure 6.1: The proposed end-to-end emotionally expressive talking face generation system overview. There are two modifications compared to the base system described in Chapter 4. The first modification is to add a speech emotion recognition module that classifies the input speech’s emotion. The second modification is to use another discriminator that checks if the video contains the given emotion.

### 6.3 Emotion Discriminator

The emotion discriminator takes the video and emotion labels as input and decides if the frames and emotion label match. A dedicated image encoder processes the video frames and extracts the image embeddings. The image embeddings are concatenated with the emotion embeddings and are fed to a bidirectional LSTM (BLSTM) layer. Each frame of the output of the BLSTM layer is fed into a fully connected (FC) layer that classifies the frame as real or fake.

The architecture of the emotion discriminator is shown in Figure 6.3. The usage of BLSTM

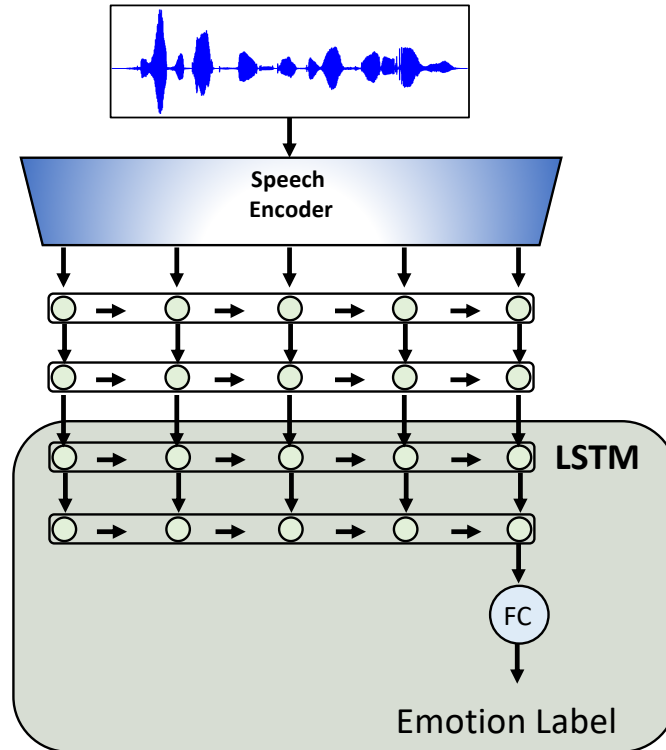


Figure 6.2: The automatic speech emotion recognition module is shown. The module accepts speech features as input to two LSTM layers followed by a fully connected layer that outputs a probability for each emotion class.

allows the discriminator to model the temporal relations, which allow the generation of emotional expressions.

### 6.3.1 Experiments

#### Dataset

In our experiments, we use the audio-visual CREMA-D dataset [228]. This dataset contains six emotions: anger, disgust, fear, happy, sad, and neutral. There are 7,442 short-clips from 91 actors. We split the dataset into training (73 speakers), validation (8 speakers) and testing (10 speakers) sets.

#### Results

The generated talking faces are shown in Figure 6.4 for different emotions using the same condition image. The network is able to generate different emotional expressions when the same condition



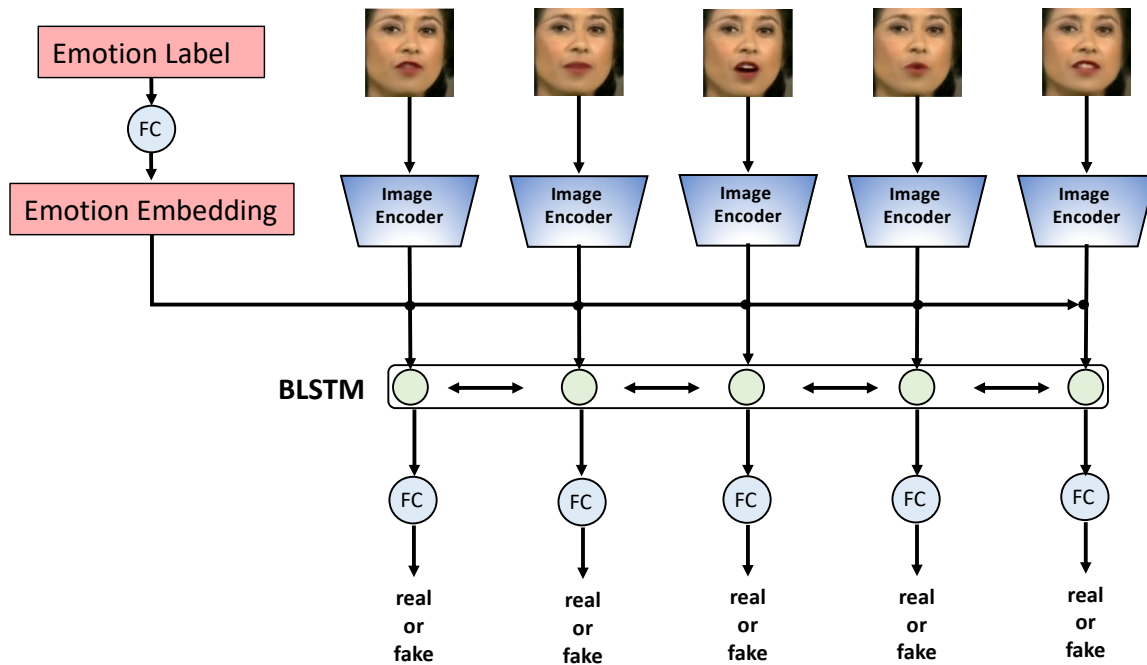


Figure 6.3: The architecture of the emotion discriminator. The video frames are fed into the image encoder, and the resulting embeddings are concatenated with emotion embeddings and are fed into a BLSTM layer. The output is fed into an FC layer that classifies the frames as real or fake.

image and different emotional speech inputs are used. Compared to the results in Chapter4, the results obtained with this network has more facial movements and looks more natural. This is ongoing work; large-scale subjective and objective evaluations will be conducted in the near future.

### 6.3.2 Conclusion

We developed an end-to-end emotionally expressive talking face generation system that operates on a raw speech waveform and a reference image. With this work, we show that by leveraging emotion labels during training, we can generate emotional expressions directly from speech. This is still ongoing work; large scale objective and subjective evaluations will be added soon.

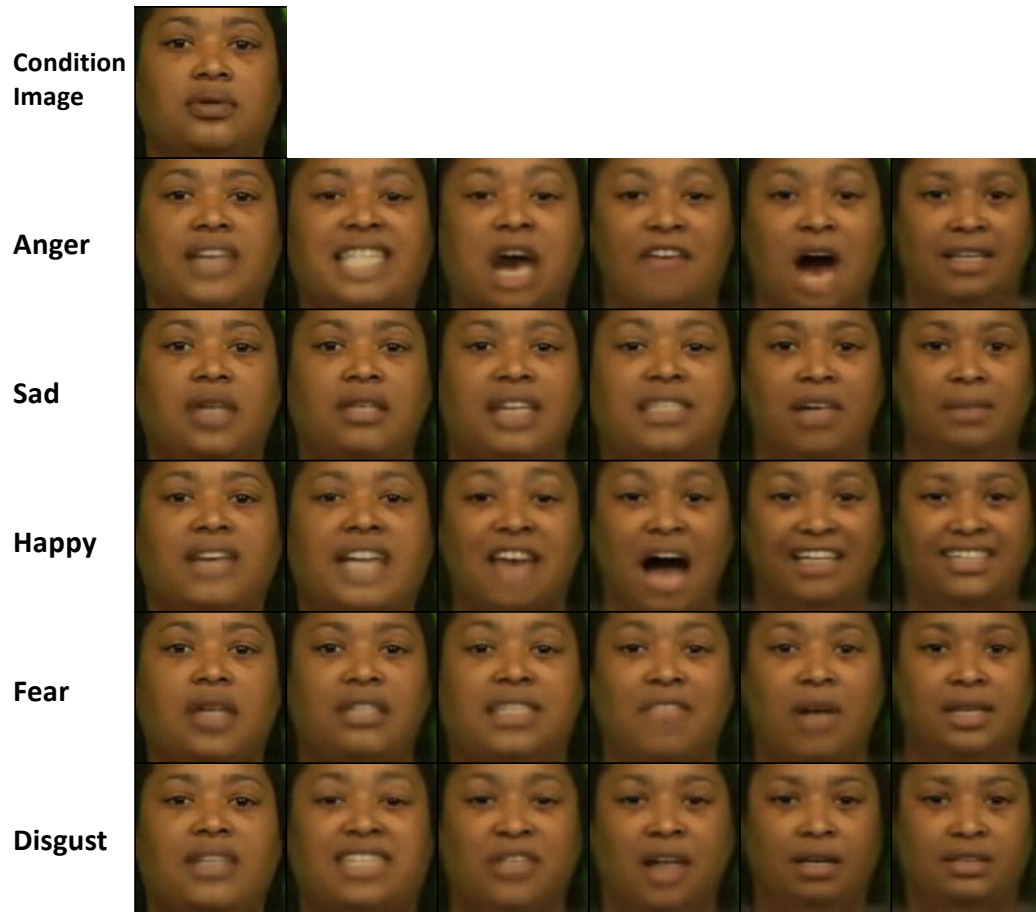


Figure 6.4: This example shows different emotions using the same condition image.

## Chapter-7

# Conclusions and Future Work

## 7.1 Conclusions

As described in this dissertation, I have worked on problems in the fields of speech enhancement (SE), speech animation (SA), and automatic speech emotion recognition (ASER). The specific contributions of this research are summarized below.

- I proposed two deep neural network (DNN) architectures for SE, and I compared the performance of the proposed networks with existing work. I concluded that our DNN based SE approaches provide benefits for speaker verification performance, speech quality, and speech intelligibility compared to the existing methods.
- I proposed a speech super-resolution system that utilizes a generative adversarial network that can work on edge devices. I concluded that the performance of the proposed system is better than the DNN-based baseline methods, supported by the objective evaluations and perceptual listening tests.
- I proposed a system that can generate landmark points of a talking face from acoustic speech in real time. I concluded that generating landmark points of talking faces from unseen speakers is realistic and can convince the volunteers who participated in the subjective tests that the images are real.
- I extended the landmark generation work by including noise-resilient training. The proposed

system can suppress unseen non-stationary noise and can generate plausible talking faces. The new architecture operates directly on raw waveforms and contains convolutional layers with 1D kernels, and outputs PCA coefficients of the face landmarks.

- I proposed an end-to-end image-based talking face generation system that can accept an arbitrarily long speech signal and outputs the talking face video in sync with the speech input. The system accepts a speech file and a reference image of a person's face and can work with unseen identities in both modalities.
- I concluded that a speech-based automatic emotion classification system is feasible as a replacement for applications that utilize naive human coders to classify emotion by showing that the computer system outperforms naive Turkers in a speech-based emotion classification task.
- I showed that an interactive speech emotion classifier, which adapts to the user's choices over time, is beneficial in situations where manually classifying emotions in a large dataset is costly, yet trained models alone will not be able to classify the data accurately.
- I concluded that pre-training autoencoders using only neutral speech data and using its encoders as feature extractors could boost the ASER performance. I also showed that variational autoencoder (VAE), adversarial autoencoder (AAE) and adversarial variational Bayes (AVB) methods, which control the distribution of the latent representation, outperform denoising autoencoder (DAE) that does not control such distribution.
- I have merged the ideas described in this thesis to propose a robust end-to-end emotionally expressive talking face generation system.

## 7.2 Future Work

One of the challenges for generating emotionally expressive talking faces is to obtain audio-visual datasets that have emotion labels, which are scarcely available in the research community. These datasets usually are designed for audio-visual emotion recognition. I believe, currently, the best

dataset for generating emotionally expressive talking faces is the Ryerson dataset [229] and CREMA-D [228]. However, these datasets contain limited vocabulary. This limitation impairs speech-mouth synchronization. Besides, the number of samples is low compared to other datasets ([163, 230]), which impairs the generalization capability. Nevertheless, one can still develop systems that can generate emotionally expressive talking faces in the same setting as the training data, e.g., using a test set that contains the same constraint vocabulary and the same image distribution, such as having only a white background. This is not a preferred approach since the system should work in the wild and must be robust against unseen inputs.

In order to overcome the lack of labeled data, two-stage systems can be utilized as a next step. The first stage can include estimating the emotions from speech, and the second stage can be a video generation from the emotion label input. These systems can be trained separately, allowing the usage of only one modality dataset at a time (speech or video only). Furthermore, to include talking faces, the video generation system can be first trained with a large scale audio-visual dataset without any emotion labels, and can be fine-tuned with emotion labels afterwards.

Another future direction to overcome the lack of data is to use parametric facial expression models such as active appearance models or any deformable face models. These models can be combined with deep neural networks since the parameter space has a low dimensionality, which can be learned with a small number of samples.

One of the possible next steps for generating emotionally talking faces is to create a large scale dataset that contains various emotions. Ideally, the emotions should be exaggerated, which is not preferred for emotion recognition research. Natural emotions are harder to detect, even for humans as shown in this thesis, and will be even harder to generate. Therefore, as the next step to generate emotionally expressive talking faces, researchers can focus on generating acted emotions and move towards generating natural emotions afterwards. Another direction can be generating intense emotions, such as laughter, crying, and screaming. Generating such emotions enable natural interactions between humans and computers. These intense emotions can be included in dataset design to move towards passing the Turing test for talking face generation systems.

## Bibliography

- [1] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.
- [2] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.
- [3] S. R. Park and J. Lee, “A fully convolutional neural network for speech enhancement,” *Computing research repository*, vol. abs/1609.07132, 2016. [Online]. Available: <http://arxiv.org/abs/1609.07132>
- [4] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, “Synthesizing obama: learning lip sync from audio,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 95, 2017.
- [5] S. E. Eskimez, R. K. Maddox, C. Xu, and Z. Duan, “Generating talking face landmarks from speech,” in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2018, pp. 372–381.
- [6] L. R. Rabiner and B. Gold, “Theory and application of digital signal processing,” *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p.*, 1975.
- [7] P. B. Denes and E. Pinson, *The speech chain*. Macmillan, 1993.
- [8] C. A. Binnie, “Bi-sensory articulation functions for normal hearing and sensorineural hearing loss patients,” *Journal of the Academy of Rehabilitative Audiology*, vol. 6, no. 2, pp. 43–53, 1973.

- [9] K. S. Helfer and R. L. Freyman, "The role of visual speech cues in reducing energetic and informational masking," *The Journal of the Acoustical Society of America*, vol. 117, no. 2, pp. 842–849, 2005.
- [10] J. G. Bernstein and K. W. Grant, "Auditory and auditory-visual intelligibility of speech in fluctuating maskers for normal-hearing and hearing-impaired listeners," *The Journal of the Acoustical Society of America*, vol. 125, no. 5, pp. 3358–3372, 2009.
- [11] R. K. Maddox, H. Atilgan, J. K. Bizley, and A. K. Lee, "Auditory selective attention is enhanced by a task-irrelevant temporally coherent visual stimulus in human listeners," *eLife*, vol. 4, 2015.
- [12] M. Schröder, "Emotional speech synthesis: A review," in *Seventh European Conference on Speech Communication and Technology*, 2001.
- [13] F. Burkhardt and N. Campbell, "Emotional speech synthesis," in *The Oxford Handbook of Affective Computing*, 2014.
- [14] Y. Cao, W. C. Tien, P. Faloutsos, and F. Pighin, "Expressive speech-driven facial animation," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 4, pp. 1283–1302, 2005.
- [15] Z. Deng, U. Neumann, J. P. Lewis, T.-Y. Kim, M. Bulut, and S. Narayanan, "Expressive facial animation synthesis by learning speech coarticulation and expression spaces," *IEEE transactions on visualization and computer graphics*, vol. 12, no. 6, pp. 1523–1534, 2006.
- [16] H. X. Pham, S. Cheung, and V. Pavlovic, "Speech-driven 3d facial animation with implicit emotional awareness: a deep learning approach," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 2328–2336.
- [17] H. X. Pham, Y. Wang, and V. Pavlovic, "End-to-end learning for 3d facial animation from raw waveforms of speech," *arXiv preprint arXiv:1710.00920*, 2017.
- [18] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 94, 2017.

- [19] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, “Hierarchical cross-modal talking face generation with dynamic pixel-wise loss,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [20] J. S. Chung, A. Jamaludin, and A. Zisserman, “You said that?” *arXiv preprint arXiv:1705.02966*, 2017.
- [21] L. Chen, Z. Li, R. K. Maddox, Z. Duan, and C. Xu, “Lip movements generation at a glance,” in *The European Conference on Computer Vision (ECCV)*, September 2018.
- [22] Y. Song, J. Zhu, X. Wang, and H. Qi, “Talking face generation by conditional recurrent adversarial network,” *arXiv preprint arXiv:1804.04786*, 2018.
- [23] K. Vougioukas, S. Petridis, and M. Pantic, “End-to-end speech-driven facial animation with temporal gans,” *arXiv preprint arXiv:1805.09313*, 2018.
- [24] S. E. Eskimez, P. Soufleris, Z. Duan, and W. Heinzelman, “Front-end speech enhancement for commercial speaker verification systems,” *Speech Communication*, vol. 99, pp. 101–113, 2018.
- [25] S. E. Eskimez, K. Koishida, and Z. Duan, “Adversarial training for speech super-resolution,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 347–358, 2019.
- [26] S. E. Eskimez and K. Koishida, “Speech super resolution generative adversarial network,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3717–3721.
- [27] S. E. Eskimez, K. Imade, N. Yang, M. Sturge-Apple, Z. Duan, and W. Heinzelman, “Emotion classification: how does an automated system compare to naive human coders?” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 2274–2278.
- [28] S. E. Eskimez, M. Sturge-Apple, Z. Duan, and W. B. Heinzelman, “Wise: Web-based interactive speech emotion classification.” in *International Joint Conference on Artificial Intelligence*



- (IJCAI) - 4th Workshop on Sentiment Analysis where AI meets Psychology (SAAIP), 2016, pp. 2–7.
- [29] S. E. Eskimez, Z. Duan, and W. Heinzelman, “Unsupervised learning approach to feature analysis for automatic speech emotion recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5099–5103.
- [30] Y. Ephraim and D. Malah, “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.
- [31] X. Lu, S. Matsuda, C. Hori, and H. Kashioka, “Speech restoration based on deep learning autoencoder with layer-wised pretraining,” in *INTERSPEECH*, 2012, pp. 1504–1507.
- [32] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, “Speech enhancement based on deep denoising autoencoder.” in *INTERSPEECH*, 2013, pp. 436–440.
- [33] —, “Ensemble modeling of denoising autoencoder for speech spectrum restoration.” in *INTERSPEECH*, vol. 14, 2014, pp. 885–889.
- [34] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “An experimental study on speech enhancement based on deep neural networks,” *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.
- [35] —, “Dynamic noise aware training for speech enhancement based on deep neural networks.” in *INTERSPEECH*, 2014, pp. 2670–2674.
- [36] —, “Global variance equalization for improving deep neural network based speech enhancement,” in *ChinaSIP*, 2014, pp. 71–75.
- [37] —, “A regression approach to speech enhancement based on deep neural networks,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 23, no. 1, pp. 7–19, 2015.

- [38] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, “Joint optimization of masks and deep recurrent neural networks for monaural source separation,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 12, pp. 2136–2147, 2015.
- [39] J. Chen, Y. Wang, S. E. Yoho, D. Wang, and E. W. Healy, “Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises,” *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2604–2612, 2016.
- [40] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, “Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr,” in *LVA/ICA*. Springer, 2015, pp. 91–99.
- [41] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, “Darpa timit acoustic-phonetic continuous speech corpus cd-rom. nist speech disc 1-1.1,” *NASA STI/Recon Technical Report n*, vol. 93, 1993.
- [42] Y. P. Li and D. L. Wang, “On the optimality of ideal binary time-frequency masks,” *Speech Communication*, vol. 51, no. 3, pp. 230–239, 2009.
- [43] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 708–712.
- [44] A. Narayanan and D. L. Wang, “Ideal ratio mask estimation using deep neural networks for robust speech recognition,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7092–7096.
- [45] Y. Wang, A. Narayanan, and D. L. Wang, “On training targets for supervised speech separation,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1849–1858, 2014.

- [46] D. S. Williamson, Y. Wang, and D. L. Wang, "Complex ratio masking for joint enhancement of magnitude and phase," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5220–5224.
- [47] S. Srinivasan, N. Roman, and D. L. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Communication*, vol. 48, no. 11, pp. 1486–1501, 2006.
- [48] U. Kjems, J. B. Boldt, M. S. Pedersen, T. Lunner, and D. Wang, "Role of mask pattern in intelligibility of ideal binary-masked noisy speech," *The Journal of the Acoustical Society of America*, vol. 126, no. 3, pp. 1415–26, 2009.
- [49] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.
- [50] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-García, D. Petrovska-Delacrétaç, and D. A. Reynolds, "A tutorial on text-independent speaker verification," *EURASIP journal on applied signal processing*, vol. 2004, pp. 430–451, 2004.
- [51] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [52] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE transactions on speech and audio processing*, vol. 13, no. 3, pp. 345–354, 2005.
- [53] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.
- [54] Y. Jiang, K. A. Lee, Z. Tang, B. Ma, A. Larcher, and H. Li, "Plda modeling in i-vector and supervector space for speaker verification," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

- [55] K.-A. Lee, A. Larcher, C. H. You, B. Ma, and H. Li, “Multi-session PLDA scoring of i-vector for partially open-set speaker detection.” in *INTERSPEECH*, 2013, pp. 3651–3655.
- [56] D. Garcia-Romero and C. Y. Espy-Wilson, “Analysis of i-vector length normalization in speaker recognition systems.” in *Interspeech*, vol. 2011, 2011, pp. 249–252.
- [57] K. W. Godin, S. O. Sadjadi, and J. H. Hansen, “Impact of noise reduction and spectrum estimation on noise robust speaker identification.” in *INTERSPEECH*, 2013, pp. 3656–3660.
- [58] X. Zhao, Y. Shao, and D. Wang, “Robust speaker identification using a casa front-end,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5468–5471.
- [59] X. Zhao, Y. Wang, and D. Wang, “Robust speaker identification in noisy and reverberant conditions,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 22, no. 4, pp. 836–845, 2014.
- [60] M. Kolbæk, Z.-H. Tan, and J. Jensen, “Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification,” in *SLT*. IEEE, 2016, pp. 305–311.
- [61] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [62] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,” in *CVPR*. IEEE, 2010, pp. 2528–2535.
- [63] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.
- [64] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

- [65] Sound ideas. [Online]. Available: <https://www.sound-ideas.com/>
- [66] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.
- [67] F. Chollet *et al.*, “Keras,” <https://github.com/fchollet/keras>, 2015.
- [68] A. Varga and H. J. 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.
- [69] Z. Duan, G. J. Mysore, and P. Smaragdis, “Speech enhancement by online non-negative spectrogram decomposition in nonstationary noise environments,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.
- [70] M. Nilsson, S. D. Soli, and J. A. Sullivan, “Development of the hearing in noise test for the measurement of speech reception thresholds in quiet and in noise,” *The Journal of the Acoustical Society of America*, vol. 95, no. 2, pp. 1085–1099, 1994.
- [71] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, “A pitch tracking corpus with evaluation on multipitch tracking scenario.” in *INTERSPEECH*, 2011, pp. 1509–1512.
- [72] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, “Perceptual evaluation of speech quality (pesq), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs,” *ITU-T Recommendation*, vol. 862, 2001.
- [73] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, “An algorithm for intelligibility prediction of time–frequency weighted noisy speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.
- [74] A. Larcher, K. A. Lee, and S. Meignier, “An extensible speaker identification sidekit in python,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5095–5099.

- [75] M. Vondrasek and P. Pollack, "Methods for speech snr estimation: Evaluation tool and analysis of vad dependency," vol. 14, 04 2005.
- [76] K. A. Lee, A. Larcher, W. Guangsen, K. Patrick, N. Brummer, D. van Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma, H. Li, T. Stafylakis, J. Alam, A. Swart, and J. Perez, "The RedDots data collection for speaker recognition," 2015, pp. 2996–3000.
- [77] "The nist year 2006 speaker recognition evaluation plan," 2006. [Online]. Available: [https://catalog.ldc.upenn.edu/docs/LDC2011S10/sre-06\\_evalplan-v9.pdf](https://catalog.ldc.upenn.edu/docs/LDC2011S10/sre-06_evalplan-v9.pdf)
- [78] "The nist year 2008 speaker recognition evaluation plan," 2008. [Online]. Available: [https://catalog.ldc.upenn.edu/docs/LDC2011S07/sre-08\\_evalplan-0408.doc](https://catalog.ldc.upenn.edu/docs/LDC2011S07/sre-08_evalplan-0408.doc)
- [79] ITU, "Paired comparison test of wideband and narrowband telephony," in *Tech. Rep. COM 12-9-E*, Mar. 1993.
- [80] L. J. Kepler, M. Terry, and R. H. Sweetman, "Telephone usage in the hearing-impaired population." *Ear and hearing*, vol. 13, no. 5, pp. 311–319, 1992.
- [81] C. Liu, Q.-J. Fu, and S. S. Narayanan, "Effect of bandwidth extension to telephone speech recognition in cochlear implant users," *The Journal of the Acoustical Society of America*, vol. 125, no. 2, pp. EL77–EL83, 2009.
- [82] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit." University of Edinburgh. The Centre for Speech Technology Research (CSTR), 2016.
- [83] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.
- [84] K. Li and C.-H. Lee, "A deep neural network approach to speech bandwidth expansion," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4395–4399.

- [85] V. Kuleshov, S. Z. Enam, and S. Ermon, “Audio super resolution using neural networks,” *arXiv preprint arXiv:1708.00853*, 2017.
- [86] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, “Stabilizing training of generative adversarial networks through regularization,” in *Advances in Neural Information Processing Systems*, 2017, pp. 2018–2028.
- [87] K.-Y. Park, “Narrowband to wideband conversion of speech using gmm based transformation,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2000, pp. 1843–1846.
- [88] B. Iser and G. Schmidt, “Bandwidth extension of telephony speech,” in *Speech and Audio Processing in Adverse Environments*. Springer, 2008, pp. 135–184.
- [89] H. Seo, H.-G. Kang, and F. Soong, “A maximum a posterior-based reconstruction approach to speech bandwidth expansion in noise,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 6087–6091.
- [90] S. Chennoukh, A. Gerrits, G. Miet, and R. Sluijter, “Speech enhancement via frequency bandwidth extension using line spectral frequencies,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 2001, pp. 665–668.
- [91] P. Jax and P. Vary, “Artificial bandwidth extension of speech signals using mmse estimation based on a hidden markov model,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 2003, pp. I–I.
- [92] G.-B. Song and P. Martynovich, “A study of hmm-based bandwidth extension of speech signals,” *Signal Processing*, vol. 89, no. 10, pp. 2036–2044, 2009.
- [93] J. Abel and T. Fingscheidt, “Artificial speech bandwidth extension using deep neural networks for wideband spectral envelope estimation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 71–83, 2018.

- [94] P. Smaragdis and B. Raj, "Example-driven bandwidth expansion," in *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*. IEEE, 2007, pp. 135–138.
- [95] D. L. Sun and R. Mazumder, "Non-negative matrix completion for bandwidth extension: A convex optimization approach," in *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*. IEEE, 2013, pp. 1–6.
- [96] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [97] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
- [98] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
- [99] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for gans do actually converge?" in *International Conference on Machine Learning*, 2018, pp. 3478–3487.
- [100] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.
- [101] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.
- [102] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, "Amortised map inference for image super-resolution," *arXiv preprint arXiv:1610.04490*, 2016.
- [103] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network." in *CVPR*, vol. 2, no. 3, 2017, p. 4.



- [104] A. Lucas, S. L. Tapia, R. Molina, and A. K. Katsaggelos, “Generative adversarial networks and perceptual losses for video super-resolution,” *arXiv preprint arXiv:1806.05764*, 2018.
- [105] S. Li, S. Villette, P. Ramadas, and D. J. Sinder, “Speech bandwidth extension using generative adversarial networks,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5029–5033.
- [106] 3GPP2 C.S0014-C v1.0, “Enhanced variable rate codec, speech service option 3, 68 and 70 for wideband spread spectrum digital systems.”
- [107] T. Gerkmann, M. Krawczyk, and R. Rehr, “Phase estimation in speech enhancement – important, important, or impossible?” in *IEEE 27th Convention of Electrical & Electronics Engineers in Israel (IEEEI)*. IEEE, 2012, pp. 1–5.
- [108] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.
- [109] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [110] P. Mermelstein, “Evaluation of a segmental snr measure as an indicator of the quality of adpcm coded speech,” *The Journal of the Acoustical Society of America*, vol. 66, no. 6, pp. 1664–1667, 1979.
- [111] J. O. Smith. Digital audio resampling home page center for computer research in music and acoustics (ccrma). [Online]. Available: <https://ccrma.stanford.edu/~jos/resample/>
- [112] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke,

- V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [113] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [114] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.
- [115] G. Hu, “100 nonspeech sounds,” Online: <http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html>, 2006.
- [116] I. Recommendation, “1534-1: Method for the subjective assessment of intermediate quality level of coding systems,” *International Telecommunication Union*, 2003.
- [117] J. W. Lyons, “Darpa timit acoustic-phonetic continuous speech corpus,” *National Institute of Standards and Technology*, 1993.
- [118] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.
- [119] D. Griffin and J. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.
- [120] B. E. Dodd and R. E. Campbell, *Hearing by eye: The psychology of lip-reading*. Lawrence Erlbaum Associates, Inc, 1987.
- [121] 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, 2006.

- [122] D. E. King, “Dlib-ml: A machine learning toolkit,” *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
- [123] S. Richie, C. Warburton, and M. Carter, “Audiovisual database of spoken american english,” *Linguistic Data Consortium*, 2009.
- [124] L. Wang, W. Han, F. K. Soong, and Q. Huo, “Text driven 3d photo-realistic talking head,” in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
- [125] V. Wan, R. Anderson, A. Blokland, N. Braunschweiler, L. Chen, B. Kolluru, J. Latorre, R. Maia, B. Stenger, K. Yanagisawa *et al.*, “Photo-realistic expressive text to talking head synthesis.” in *INTERSPEECH*, 2013, pp. 2667–2669.
- [126] B. Fan, L. Wang, F. K. Soong, and L. Xie, “Photo-real talking head with deep bidirectional lstm,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4884–4888.
- [127] S. Cassidy, B. Stenger, L. V. Dongen, K. Yanagisawa, R. Anderson, V. Wan, S. Baron-Cohen, and R. Cipolla, “Expressive visual text-to-speech as an assistive technology for individuals with autism spectrum conditions,” *Computer Vision and Image Understanding*, vol. 148, pp. 193 – 200, 2016.
- [128] M. Brand, “Voice puppetry,” in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 21–28.
- [129] K. Choi, Y. Luo, and J.-N. Hwang, “Hidden markov model inversion for audio-to-visual conversion in an mpeg-4 facial animation system,” *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 29, pp. 51–61, 2001.
- [130] D. Cosker, D. Marshall, P. Rosin, and Y. Hicks, “Video realistic talking heads using hierarchical non-linear speech-appearance models,” *Mirage, France*, vol. 147, 2003.

- [131] D. Cosker, D. Marshall, P. L. Rosin, and Y. Hicks, "Speech driven facial animation using a hidden markov coarticulation model," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, vol. 1. IEEE, 2004, pp. 128–131.
- [132] L. Xie and Z.-Q. Liu, "A coupled hmm approach to video-realistic speech animation," *Pattern Recognition*, vol. 40, pp. 2325–2340, 2007.
- [133] L. D. Terissi and J. C. Gómez, "Audio-to-visual conversion via hmm inversion for speech-driven facial animation," in *Brazilian Symposium on Artificial Intelligence*. Springer, 2008, pp. 33–42.
- [134] X. Zhang, L. Wang, G. Li, F. Seide, and F. K. Soong, "A new language independent, photo-realistic talking head driven by voice only." in *Interspeech*, 2013, pp. 2743–2747.
- [135] S. Mallick. (2016) Face morph using opencv & c++ / python. [Online]. Available: <http://www.learnopencv.com/face-morph-using-opencv-cpp-python/>
- [136] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [137] J. S. Garofalo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "The darpa timit acoustic-phonetic continuous speech corpus cdrom," *Linguistic Data Consortium*, 1993.
- [138] T. W. Tillman and R. Carhart, "An expanded test for speech discrimination utilizing cnc monosyllabic words: Northwestern university auditory test no. 6," Northwestern University Evanston Auditory Research Lab, Tech. Rep., 1966.
- [139] P. J. Blamey, B. C. Pyman, G. M. Clark, R. C. Dowell, M. Gordon, A. M. Brown, and R. D. Hollow, "Factors predicting postoperative sentence scores in postlinguistically deaf adult cochlear implant patients," *Annals of Otolaryngology, Rhinology & Laryngology*, vol. 101, no. 4, pp. 342–348, 1992.

- [140] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K. Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein, “Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation,” *arXiv preprint arXiv:1804.03619*, 2018.
- [141] T. Afouras, J. S. Chung, and A. Zisserman, “The conversation: Deep audio-visual speech enhancement,” *arXiv preprint arXiv:1804.04121*, 2018.
- [142] Y. Mroueh, E. Marcheret, and V. Goel, “Deep multimodal learning for audio-visual speech recognition,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2130–2134.
- [143] S. Petridis, Z. Li, and M. Pantic, “End-to-end visual speech recognition with lstms,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2592–2596.
- [144] M. S. Hossain, G. Muhammad, M. F. Alhamid, B. Song, and K. Al-Mutib, “Audio-visual emotion recognition using big data towards 5g,” *Mobile Networks and Applications*, vol. 21, no. 5, pp. 753–763, 2016.
- [145] D. Dov, R. Talmon, and I. Cohen, “Audio-visual voice activity detection using diffusion maps,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 732–745, 2015.
- [146] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng, “Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.
- [147] M. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, “A comprehensive survey of deep learning for image captioning,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, p. 118, 2019.

- [148] J. Huang, W. Zhou, H. Li, and W. Li, "Sign language recognition using 3d convolutional neural networks," in *2015 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2015, pp. 1–6.
- [149] H. Tang, Y. Hu, Y. Fu, M. Hasegawa-Johnson, and T. S. Huang, "Real-time conversion from a single 2d face image to a 3d text-driven emotive audio-visual avatar," in *IEEE International Conference on Multimedia and Expo*, 2008, pp. 1205–1208.
- [150] L. Xie, N. Sun, and B. Fan, "A statistical parametric approach to video-realistic text-driven talking avatar," *Multimedia Tools and Applications*, vol. 73, no. 1, pp. 377–396, 2014.
- [151] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
- [152] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.
- [153] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1021–1030.
- [154] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.
- [155] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [156] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.
- [157] 2018. [Online]. Available: <http://www.sens.com/products/stevi-speech-test-video-corpus/>
- [158] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

- [159] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [160] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, “Mocogan: Decomposing motion and content for video generation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1526–1535.
- [161] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [162] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.
- [163] J. S. Chung and A. Zisserman, “Lip reading in the wild,” in *Asian Conference on Computer Vision*. Springer, 2016, pp. 87–103.
- [164] K. R. Scherer, “Vocal affect expression: a review and a model for future research.” *Psychol. Bull.*, vol. 99, no. 2, p. 143, 1986.
- [165] C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan, “Analysis of emotion recognition using facial expressions, speech and multimodal information,” in *Proc. ICMI*. ACM, 2004, pp. 205–211.
- [166] G. Huisman, M. van Hout, E. van Dijk, T. van der Geest, and D. Heylen, “Lemtool: measuring emotions in visual interfaces,” in *Proc. ACM SIGCHI*, 2013, pp. 351–360.
- [167] S. Ozkul, E. Bozkurt, S. Asta, Y. Yemez, and E. Erzin, “Multimodal analysis of upper-body gestures, facial expressions and speech,” in *Proc. ES3*, 2012.
- [168] C.-H. Wu, J.-C. Lin, and W.-L. Wei, “Two-level hierarchical alignment for semi-coupled hmm-based audiovisual emotion recognition with temporal course,” *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1880–1895, 2013.

- [169] N. Yang, “Algorithms for affective and ubiquitous sensing systems and for protein structure prediction,” Ph.D. dissertation, University of Rochester, 2015, <http://hdl.handle.net/1802/29666>.
- [170] D. Bitouk, R. Verma, and A. Nenkova, “Class-level spectral features for emotion recognition,” *Speech Commun.*, vol. 52, no. 7, pp. 613–625, 2010.
- [171] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas, “Emotionsense: a mobile phones based adaptive platform for experimental social psychology research,” in *Proc. 12th ACM Int. Conf. on Ubiquitous Computing*, 2010, pp. 281–290.
- [172] V. Sethu, E. Ambikairajah, and J. Epps, “Empirical mode decomposition based weighted frequency feature for speech-based emotion classification,” in *Proc. IEEE ICASSP*, 2008, pp. 5017–5020.
- [173] M. Liberman, K. Davis, M. Grossman, N. Martey, and J. Bell, “Emotional prosody speech and transcripts,” in *Proc. LDC*, 2002.
- [174] Y. Holkamp and J. Schavemaker, “A comparison of human and machine learning-based accuracy for valence classification of subjects in video fragments,” in *Proc. Meas. Beh.*, 2014, pp. 27–29.
- [175] J. H. Janssen, P. Tacken, J. de Vries, E. L. van den Broek, J. H. Westerink, P. Haselager, and W. A. IJsselsteijn, “Machines outperform laypersons in recognizing emotions elicited by autobiographical recollection,” *J. Hum.-Comput. Int.*, vol. 28, no. 6, pp. 479–517, 2013.
- [176] J. Susskind, G. Littlewort, M. Bartlett, J. Movellan, and A. Anderson, “Human and computer recognition of facial expressions of emotion,” *Neuropsychologia*, vol. 45, no. 1, pp. 152–162, 2007.
- [177] A. Shaukat and K. Chen, “Emotional state categorization from speech: machine vs. human,” *arXiv preprint arXiv:1009.0108*, 2010.



- [178] J. Esparza, S. Scherer, A. Brechmann, and F. Schwenker, "Automatic emotion classification vs. human perception: Comparing machine performance to the human benchmark," in *Proc. IEEE ISSPA*, 2012, pp. 1253–1258.
- [179] S. L. Tóth, D. Sztahó, and K. Vicsi, "Speech emotion perception by human and machine," in *Lect. Notes Artif. Int.* Springer, 2008, pp. 213–224.
- [180] B. Schuller, G. Rigoll, and M. Lang, "Hidden markov model-based speech emotion recognition," in *Proc. IEEE ICASSP*, vol. 2, 2003, pp. 1–4.
- [181] S. Yun and C. D. Yoo, "Loss-scaled large-margin gaussian mixture models for speech emotion classification," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 20, no. 2, pp. 585–598, 2012.
- [182] A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*. New York: Springer, 2006.
- [183] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*. Wiley New York, 1998, vol. 1.
- [184] M. Farrús, P. Ejarque, A. Temko, and J. Hernando, "Histogram equalization in svm multimodal person verification," in *Proc. Adv. in Bio.* Springer, 2007, pp. 819–827.
- [185] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
- [186] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1-3, pp. 389–422, Mar. 2002. [Online]. Available: <http://dx.doi.org/10.1023/A:1012487302797>
- [187] J. J. Campos, R. G. Campos, and K. C. Barrett, "Emergent themes in the study of emotional development and emotion regulation." *Dev Psychol.*, vol. 25, no. 3, p. 394, 1989.
- [188] P. Ekman, "An argument for basic emotions," *Cognition Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

- [189] D. Keltner and A. M. Kring, "Emotion, social function, and psychopathology." *Rev. Gen. Psychol.*, vol. 2, no. 3, p. 320, 1998.
- [190] V. A. Petrushin, "Emotion in speech: Recognition and application to call centers," in *In Engr*, 1999, pp. 7–10.
- [191] P. Gupta, "Two-Stream Emotion Recognition For Call Center Monitoring," in *Interspeech 2007*, 2007. [Online]. Available: <https://ssli.ee.washington.edu/proceedings/interspeech07/IS2007/PDF/AUTHOR/IS070468.PDF>
- [192] J. S. Park, J. H. Kim, and Y. H. Oh, "Feature vector classification based speech emotion recognition for service robots," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1590–1596, August 2009.
- [193] C.-Y. Liu, T.-H. Hung, K.-C. Cheng, and T.-H. S. Li, "Hmm and bpnn based speech recognition system for home service robot," in *Advanced Robotics and Intelligent Systems (ARIS), 2013 International Conference on*. IEEE, 2013, pp. 38–43.
- [194] C. M. Jones and I.-M. Jonsson, "Automatic recognition of affective cues in the speech of car drivers to allow appropriate responses," in *Proceedings of the 17th Australia Conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future*, ser. OZCHI '05, Narrabundah, Australia, Australia, 2005, pp. 1–10. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1108368.1108397>
- [195] A. Tawari and M. Trivedi, "Speech based emotion classification framework for driver assistance system," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 174–178.
- [196] C. Busso, S. Lee, and S. Narayanan, "Analysis of emotionally salient aspects of fundamental frequency for emotion detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 582–596, May 2009.

- [197] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, and B. Schuller, "Deep neural networks for acoustic emotion recognition: Raising the benchmarks," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 5688–5691.
- [198] N. Yang, "Algorithms for affective and ubiquitous sensing systems and for protein structure prediction," Ph.D. dissertation, University of Rochester, 2015, <http://hdl.handle.net/1802/29666>.
- [199] T. Vogt, E. Andr  , and N. Bee, "Emovoice - a framework for online recognition of emotions from voice," in *In Proceedings of Workshop on Perception and Interactive Technologies for Speech-Based Systems*, Springer, Kloster Irsee, 2008.
- [200] F. Eyben, M. W  llmer, and B. Schuller, "openear - introducing the munich open-source emotion and affect recognition toolkit," in *In ACHI*, 2009, pp. 576–581.
- [201] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
- [202] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, version 3.4*. Cambridge, UK: Cambridge University Engineering Department, 2006.
- [203] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: <http://doi.acm.org/10.1145/1656274.1656278>
- [204] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [205] F. Ringeval, A. Sonderegger, J. Sauer, and D. Lalanne, "Introducing the recola multimodal corpus of remote collaborative and affective interactions," in *2013 10th IEEE International*

- Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, April 2013, pp. 1–8.
- [206] J. Wagner, J. Kim, and E. André, “From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification,” in *International Conference on Multimedia and Expo (ICME)*. IEEE, 2005, pp. 940–943.
- [207] H. Gunes and M. Piccardi, “Bi-modal emotion recognition from expressive face and body gestures,” *Journal of Network and Computer Applications*, vol. 30, no. 4, pp. 1334–1345, 2007.
- [208] M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.
- [209] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [210] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.
- [211] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.
- [212] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [213] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [214] D. Neiberg, K. Elenius, and K. Laskowski, “Emotion recognition in spontaneous speech using gmms,” in *Ninth International Conference on Spoken Language Processing*, 2006.

- [215] K. W. Gamage, V. Sethu, P. N. Le, and E. Ambikairajah, "An i-vector gplda system for speech based emotion recognition," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*. IEEE, 2015, pp. 289–292.
- [216] T. L. Nwe, S. W. Foo, and L. C. De Silva, "Speech emotion recognition using hidden markov models," *Speech communication*, vol. 41, no. 4, pp. 603–623, 2003.
- [217] B. Schuller, D. Arsic, F. Wallhoff, G. Rigoll *et al.*, "Emotion recognition in the noise applying large acoustic feature sets," *Speech Prosody, Dresden*, pp. 276–289, 2006.
- [218] N. Yang, J. Yuan, Y. Zhou, I. Demirkol, Z. Duan, W. Heinzelman, and M. Sturge-Apple, "Enhanced multiclass svm with thresholding fusion for speech-based emotion classification," *International Journal of Speech Technology*, vol. 20, no. 1, pp. 27–41, 2017.
- [219] R. Xia and Y. Liu, "Using denoising autoencoder for emotion recognition." in *Interspeech*, 2013, pp. 2886–2889.
- [220] R. Xia, J. Deng, B. Schuller, and Y. Liu, "Modeling gender information for emotion recognition using denoising autoencoder," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 990–994.
- [221] S. Ghosh, E. Laksana, L.-P. Morency, and S. Scherer, "Learning representations of affect from speech," *arXiv preprint arXiv:1511.04747*, 2015.
- [222] ———, "Representation learning for speech emotion recognition." in *INTERSPEECH*, 2016, pp. 3603–3607.
- [223] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2013, pp. 511–516.
- [224] J. Deng, Z. Zhang, F. Eyben, and B. Schuller, "Autoencoder-based unsupervised domain adaptation for speech emotion recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1068–1072, 2014.

- [225] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial autoencoders,” *arXiv preprint arXiv:1511.05644*, 2015.
- [226] L. Mescheder, S. Nowozin, and A. Geiger, “Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks,” *arXiv preprint arXiv:1701.04722*, 2017.
- [227] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, “Iemocap: Interactive emotional dyadic motion capture database,” *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.
- [228] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, “Crema-d: Crowd-sourced emotional multimodal actors dataset,” *IEEE transactions on affective computing*, vol. 5, no. 4, pp. 377–390, 2014.
- [229] S. R. Livingstone and F. A. Russo, “The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english,” *PloS one*, vol. 13, no. 5, p. e0196391, 2018.
- [230] 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.