ACOUSTIC SCENE CLASSIFICATION: FROM A HYBRID CLASSIFIER TO DEEP LEARNING

Anastasios Vafeiadis[1], *Dimitrios Kalatzis*[1], *Konstantinos Votis*[1], *Dimitrios Giakoumis*[1], *Dimitrios Tzovaras*[1], *Liming Chen*[2], *Raouf Hamzaoui*[2]

[1] Information Technologies Institute,
Center for Research & Technology Hellas, Thessaloniki, Greece
{anasvaf, dkal, kvotis, dgiakoum, tzovaras}@iti.gr
[2] Faculty of Technology, De Montfort University, Leicester, UK
{liming.chen, rhamzaoui}@dmu.ac.uk

ABSTRACT

This report describes our contribution to the 2017 Detection and Classification of Acoustic Scenes and Events (DCASE) challenge. We investigated two approaches for the acoustic scene classification task. Firstly, we used a combination of features in the time and frequency domain and a hybrid Support Vector Machines - Hidden Markov Model (SVM-HMM) classifier to achieve an average accuracy over 4-folds of 80.9% on the development dataset and 61.0% on the evaluation dataset. Secondly, by exploiting data-augmentation techniques and using the whole segment (as opposed to splitting into sub-sequences) as an input, the accuracy of our CNN system was boosted to 95.9%. However, due to the small number of kernels used for the CNN and a failure of capturing the global information of the audio signals, it achieved an accuracy of 49.5% on the evaluation dataset. Our two approaches outperformed the DCASE baseline method, which uses log-mel band energies for feature extraction and a Multi-Layer Perceptron (MLP) to achieve an average accuracy over 4-folds of 74.8%.

Index Terms— Acoustic scene classification, feature extraction, deep learning, spectral features, data augmentation

1. INTRODUCTION

Environmental sounds hold a large amount of information from our everyday environment. Sounds can be captured unobtrusevily with the help of mobile phones (MEMS microphones) or microphones (Soundman OKM II Klassik/studio A3) [1].

The process of acoustic scene classification involves the extraction of features from sound and the use of these features to identify the class of the scene.

Over the last few years, many researchers have worked on acoustic scene classification, by recognizing single events in monophonic recordings [2] and multiple concurrent events in polyphonic recordings [3]. Different approaches to feature extraction have been introduced [4], data augmentation techniques [5], use of hybrid classifiers [6] and neural networks [7] and finally comparisons between well-known classifiers and deep learning models using public datasets [8]. However, it must be noted that the problem of audio-based event recognition remains a hard task. This is because features and classifiers that work extremely well for a specific dataset may fail for another.

In this report we present two approaches for acoustic scene classification using the DCASE 2017 development dataset for training and validation and the unlabeled DCASE 2017 evaluation dataset for testing. Our first approach combines time and frequency domain features, applies statistical analysis for dimensionality reduction, and uses a hybrid SVM-HMM for classification. Our second approach uses a CNN for classification and exploits data augmentation techniques. It differs from other CNN-based methods [9, 10] first, in that we feed the whole segment as input to the network (as opposed to splitting it in sub-sequences) and second, in that we apply max pooling to both dimensions of the input (i.e. both time and frequency). By doing that, we reduce the dimensionality of the input in a more uniform manner, thus preserving more of the segment's spatio-temporal structure, yielding more salient features with each consecutive convolutional-max pooling operation.

The remainder of the report is organized as follows. Chapter 2 describes the steps in acoustic scene classification. Chapter 3 presents the first approach using the SVM-HMM classifier and the results obtained. Chapter 4 describes the CNN model and its performance. Finally, chapter 5 concludes the report.

2. ACOUSTIC SCENE CLASSIFICATION FRAMEWORK

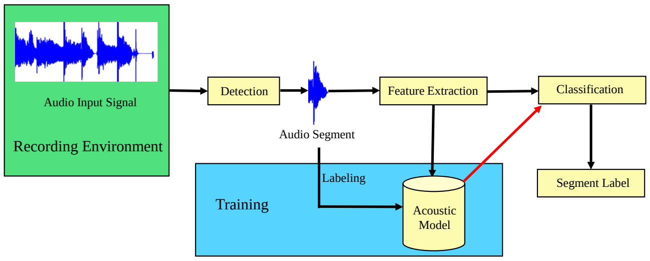

Figure 1: Typical Acoustic Scene Classification system.

Fig. 1 shows a typical Acoustic Scene Classification (ASC) system and its main components. The detection module first segments of the sound events from the continuous audio signal. Then features are extracted to characterize the acoustic information. Finally, classification matches the unknown features with an acoustic model, learnt during a training phase, to output a label for the segmented sound event.

The **Audio Input Signal** collection is the first step in the process. This step depends on the corresponding classification task.

For instance, in handwriting recognition, this step involves splitting each sentence into separate words and letters and performing other initial tasks. For sound recognition, this step involves capturing a sound from the environment and loading it into a computer. This task is typically performed using a microphone. In addition, a computer converts the analog signal to the digital format via sampling and quantization.

Feature Extraction is the second step in the process. Feature extraction involves selecting pieces of the input data that uniquely characterize that information. The choice of features depends on the application and it is based on the belief of which feature most accurately characterizes the sound.

All these levels of understanding should be combined to produce a system that is able to extract the best features. For example, a speech recognition system could use statistical techniques to identify when speech is passed into a microphone (speech/non-speech detection). Syntactical techniques could then split the speech into separate words. Each word could then be recognized and then a semantic technique could be used to interpret each word using a dictionary.

Classification is the third step in the process. For sound recognition, many techniques have been used, including Hidden Markov Models, Neural Networks and Reference Model Databases (as used with Dynamic Time Wrapping) [11]. All of these techniques use a training/testing paradigm. Training gives the system a series of examples of a particular item, so the system can learn the general characteristics of this item. Then, during testing, the system can identify the class of the item being tested.

However, classification faces one challenge. It is important to ensure that the testing and the training sets are recorded in the same conditions in order to get optimum results. In an analysis of training and testing techniques for speech recognition, Murthy, et al. [12] explains how training data must be collected from within a variety of different environments to make sure that a representative set of training data is stored in the database. They use of a filter bank to remove erroneous environmental sounds from the sound sample to ensure that these do not affect classification. Hence, robust recognition techniques are most useful if noise and other factors affect the training data.

3. PROPOSED SVM-HMM SYSTEM

In this section we describe the hybrid SVM-HMM system that was implemented using the baseline code that was provided by the organizers. We have used well-known features from the field of speech recognition and previous works in environmental sound classification.

3.1. Feature Extraction

In the feature extraction phase all audio files are transformed into the frequency domain through a 2048-sample Short-Time Fourier Transform (STFT) with 50% overlap, in order to avoid loss of information. Each frame has a window size of 40 ms with a 20 ms hop size from the next one. In our approach, we convert the 24-bit depth stereo audio recordings to mono, then the spectrum is divided into 40 mel-spaced bands, and the following features are extracted for each band: *Spectral Rolloff (SR), Spectral Centroid (SC), Mel-Frequency Cepstral Coefficients (MFCC) (static, first and second order derivatives) and Zero-Crossing Rate (ZCR).*

For each mel band there are 12 cepstral coefficients + 1 energy coefficient, 12 delta cepstral coefficients + 1 delta energy coefficient and 12 double delta cepstral coefficients + 1 double delta energy coefficient; making a total of 39 MFCC features.

Taking the average ZCR gives a reasonable way to estimate the frequency of a sine wave. ZCR was important in recordings such as the cafe/restaurant, grocery store, metro station, tram and train, in order to separate the speech from the non-speech components.

SC and SR are defined based on the magnitude spectrum of the STFT. They measure the average frequency weighted by amplitude of a spectrum as well as the frequency below which 90% (in our case) of the magnitude distribution is concentrated.

Statistics such as the mean, variance, skewness, first and second derivatives are computed to aggregate all time frames into a smaller set of values representing each of features for every mel-band. One of the main problems is that whenever there is a large dataset, using a large number of features can slow down the training process [13]. We used the Sequential Backward Selection (SBS) [14], which sequentially constructs classifiers for each subset of features by removing one feature at a time from the previous set and finally outputs the classification error rate. The combination of all the features along with SBS increased the classification accuracy in 4-folds from 77.1% to 80.9%

Table 1 shows a comparison between our hybrid SVM-HMM approach, the DCASE2017 baseline based on Gaussian Mixture Model (GMM), using the development dataset, and the performance of our SVM-HMM system with the evaluation dataset.

Table 1: Performance comparison (averaged over 4-folds) between the DCASE2017 baseline based on GMM and our hybrid SVM-HMM approach

Class	**Baseline GMM w/ MFCC features (%) (development dataset)**	**Our approach SVM-HMM w/ MFCC, ZCR, SR. SC features (%) (development dataset)**	**Our approach SVM-HMM w/ MFCC, ZCR, SR. SC features (%) (evaluation dataset)**
Beach	75.0	78.8	23.1
Bus	84.3	90.1	42.6
Cafe/Restaurant	81.7	68.3	58.3
Car	91.0	94.2	66.7
City center	91.0	91.3	77.8
Forest path	73.4	85.6	86.1
Grocery store	67.9	80.8	64.8
Home	71.4	74.5	94.4
Library	63.5	65.7	39.8
Metro station	81.4	89.1	92.6
Office	97.1	99.0	54.6
Park	39.1	59.0	20.4
Residential area	74.7	79.8	72.2
Train	41.0	63.8	81.5
Tram	79.2	85.6	39.8
Average	***74.1***	***80.9***	***61.0***

3.2. Classification

The development dataset is split by the organizers in 4-folds each containing 3510 training recordings and 1170 testing recordings (75/25 split). For the training, we use the features that were mentioned in the previous section as an input to the HMM. Then, the most probable model is associated with every sequence which needs to be classified. The HMM output, which can be considered as a further refinement of the HMM input features is in turn fed to the SVM classifier in the testing phase, as it was originally proposed by Bisio et al. [16] for gender-driven emotion recognition. For the SVM, we used the Radial-Basis Function (RBF) kernel and after performing grid search, we found that the best parameters were σ = 0.1 and C = 100. The parameter σ of the RBF kernel handles the non-linear

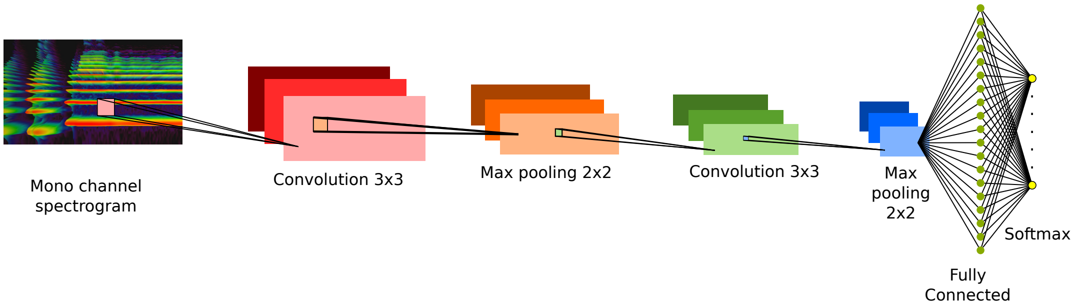

Figure 2: Block diagram of a Convolutional Neural Network.

classification and is considered to be a similarity measure between two points. C is the cost of classification.

Fig.3 shows the Receiver Operating Characteristics (ROC) curves of the SVM-HMM model. The system was not able to create a good model for classes such as: library, park, train and cafe/restaurant.

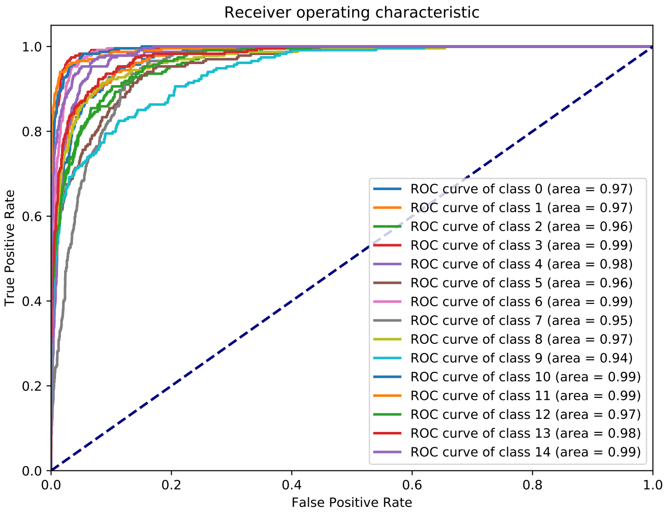

Figure 3: ROC curves of the SVM-HMM model. Classes 0-14 represent the alphabetical order of the classes from the challenge.

4. PROPOSED CNN SYSTEM

In this section we describe the CNN system that was implemented in Python using Librosa [17] for feature extraction and Keras [18] for the development of the model.The network was trained on NVIDIA GeForce GTX 1080 Ti and Tesla K40M GPUs.

4.1. Data augmentation

Environmental audio recordings have different temporal properties. Therefore, we need to make sure that we have captured all the significant information of the signal in both the time and frequency domain. Most environmental audio signals have non-stationary noise, which is often time-varying correlated and non-Gaussian.

Based on previous research [5, 19], data augmentation proved to significantly improve the total performance of the classification system. In our approach we produced two additional augmented recordings from the original ones. Hence the total training audio files of each fold were increased from 3510 to 10530 and the testing from 1170 to 3510. For the first recording we added Gaussian noise over the 10 seconds of the recording; hence it has an average time domain value of zero. This allowed us to train our system better, since the evaluation recordings would also introduce various noises (e.g. kids playing on the beach). For the second recording we re-sampled the original signal from 44.1 kHz to 16 kHz. We kept the same length as the original recording and padded with zeros where necessary. We found that a lot of information at around 11 kHz was necessary for classes such as ”beach” where there was a lot of noise from the wind and the sea waves.

4.2. Feature Extraction

All the recordings were converted into mono channels. In this approach, we use the mel-spectrogram with 128 bins which is a sufficient size to keep spectral characteristics while greatly reduces the feature dimension. Each frame has a window size of 40 ms with a 20 ms hop size from the next one. We normalized the values before using them as an input into the CNN network by subtracting the mean and dividing by the standard deviation.

4.3. CNN description

Our network architecture consists of 4 convolutional layers (Fig.2). In detail, the first layer performs convolutions over the spectrogram of the input segment, using 3x3 kernels. The output is fed to a second convolutional layer which is identical to the first. A 2x2 max pooling operation, then, follows the second layer and the subsampled feature maps are fed to two consecutive convolutional layers, each followed by max pooling operations. Each convolution operation is followed by batch normalization [20] of its outputs, before the element-wise application of the ELU activation function [21] to facilitate training and improve convergence time. After each max pooling operation, we apply dropout [22] with an input dropout rate of 0.2. The number of kernels in all convolutional layers is 5.

The resulting feature maps of the consecutive convolution-max pooling operations are then fed as input to a fully-connected layer with 128 logistic sigmoid units to which we also apply dropout with a rate of 0.2, followed by the output layer which computes the softmax function. Classification is, then, obtained through hard assignment of the normalized output of the softmax function. I.e.:

$$c = arg\max_{i} y_i, \quad for\ \ i = 1, \ldots, n \tag{1}$$

Table 2: Comparison of recognition accuracy between the proposed system and the second baseline system based on Log-mel band energies and MLP for the DCASE 2017 dataset averaged over 4-folds

Class	**Baseline** **Log-mel band energies** **MLP** **(%)** **(development dataset)**	**Our System (with data augmentation)** **Log-mel spectrogram** **CNN** **(%)** **(development dataset)**	**Our System (with data augmentation)** **Log-mel spectrogram** **CNN** **(%)** **(evaluation dataset)**
Beach	75.3	97.8	35.2
Bus	71.8	92.3	23.1
Cafe/Restaurant	57.7	96.2	58.3
Car	97.1	97.4	63.0
City center	90.7	99.6	90.7
Forest path	79.5	100.0	90.7
Grocery store	58.7	99.6	57.4
Home	68.6	98.3	61.1
Library	57.1	95.3	20.4
Metro station	91.7	92.3	38.0
Office	99.7	100.0	53.7
Park	70.2	90.6	25.9
Residential area	64.1	90.2	45.4
Train	58.0	93.2	59.3
Tram	81.7	97.0	48.1
Average	***74.8***	***95.9***	***49.5***

$$y_i = \frac{\exp x_i}{\sum_{j=1}^{N} \exp x_j} \tag{2}$$

where, c is the argmax-index position of each row (class) i in the set $1, ..., N$ for which y_i is maximum and x is the net input.

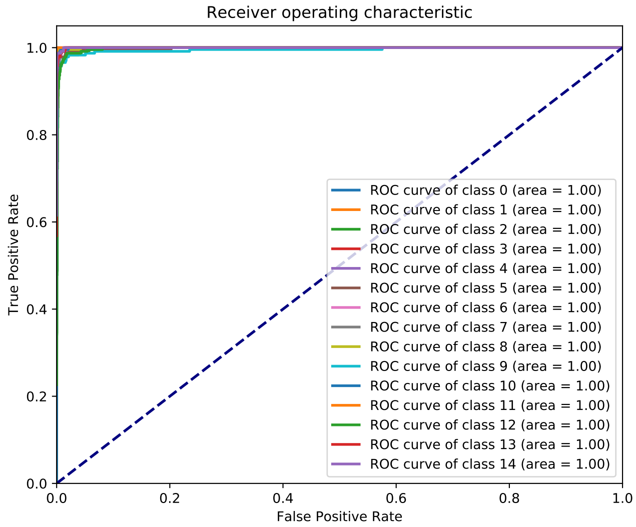

Figure 4: ROC curves of the final CNN model. Classes 0-14 represent the alphabetical order of the classes from the challenge.

Fig. 4 shows the ROC curves of our CNN model. It proves that we have a good model, as the area under the ROC curve (AUC) is approximately 0.99. Table 2 compares the classification accuracies between the baseline model and the proposed CNN model.

5. CONCLUSIONS

We presented two systems that use environmental sounds for event detection in an indoor or an outdoor environment. In order to further evaluate the performance of the proposed systems we have to test it extensively with more public datasets (i.e. UrbanSounds 8K, ESC-50, Chime Home, etc.)

Our system severely underperformed in the evaluation set, with performance dropping by almost 50%. We attribute this to a combination of inadequate feature extraction and model capacity. While our extracted features were adequate enough to encode information present in the development set (and thus lead to good development held out performance) they seem to have captured mostly local information, or at least failed to encapsulate the global structure hidden in the data. This, coupled with the relatively small capacity of our model (only 5 convolutional kernels) played a significant role in the worsening of the model's performance in the evaluation set.

We plan to explore statistical feature selection with Analysis Of Variance(ANOVA) and SBS for the CNN and compare the performance with the addition of bidirectional Long Short-Term Memory (LSTM) layers. The data augmentation technique used for the CNN will be tested with well-known classifiers. Furthermore, we will use a Variational Auto-Encoder data augmentation method, since it has proven to create robust models in the field of speech recognition [23]. Finally, tests with binaural recordings will be conducted to evaluate the performance.

6. ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 676157, project ACROSSING.

7. REFERENCES

[1] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "Dcase 2017 challenge setup: tasks, datasets and baseline system."

[2] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, May 2015.

[3] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, June 2017.

[4] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "CP-JKU submissions for DCASE-2016: a hybrid approach using binaural i-vectors and deep convolutional neural networks," DCASE2016 Challenge, Tech. Rep., Sept. 2016.

[5] J. Salamon and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, Mar. 2017.

[6] J. Liu, X. Yu, W. Wan, and C. Li, "Multi-classification of audio signal based on modified svm," in *IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2009)*, Dec. 2009, pp. 331–334.

[7] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. B. Jackson, and M. D. Plumbley, "Unsupervised feature learning based on deep models for environmental audio tagging," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1230–1241, June 2017.

[8] J. Li, W. Dai, F. Metze, S. Qu, and S. Das, "A comparison of deep learning methods for environmental sound detection," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 126–130.

[9] T. Lidy and A. Schindler, "CQT-based convolutional neural networks for audio scene classification," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Sept. 2016, pp. 60–64.

[10] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, *DCASE 2016 Acoustic Scene Classification Using Convolutional Neural Networks*. Tampere University of Technology. Department of Signal Processing, 9 2016.

[11] P. Khunarsal, C. Lursinsap, and T. Raicharoen, "Very short time environmental sound classification based on spectrogram pattern matching," *Information Sciences*, vol. 243, pp. 57 – 74, 2013.

[12] H. A. Murthy, F. Beaufays, L. Heck, and M. Weintraub, "Robust text-independent speaker identification over telephone channels," *IEEE Transactions on Speech and Audio Processing*, vol. 7/5, Sept. 1999.

[13] R. Murata, Y. Mishina, Y. Yamauchi, T. Yamashita, and H. Fujiyoshi, "Efficient feature selection method using contribution ratio by random forest," in *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, Jan. 2015, pp. 1–6.

[14] S. Visalakshi and V. Radha, "A literature review of feature selection techniques and applications: Review of feature selection in data mining," in *2014 IEEE International Conference on Computational Intelligence and Computing Research*, Dec. 2014, pp. 1–6.

[15] A. Kumar, B. Elizalde, A. Shah, R. Badlani, E. Vincent, B. Raj, and I. Lane, "DCASE challenge task 1," DCASE2016 Challenge, Tech. Rep., Sept. 2016.

[16] I. Bisio, A. Delfino, F. Lavagetto, M. Marchese, and A. Sciarrone, "Gender-driven emotion recognition through speech signals for ambient intelligence applications," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 2, pp. 244–257, Dec. 2013.

[17] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and Music Signal Analysis in Python," in *Proceedings of the 14th Python in Science Conference*, Kathryn Huff and James Bergstra, Eds., 2015, pp. 18 – 25.

[18] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[19] B. McFee, E. Humphrey, and J. Bello, "A software framework for musical data augmentation," in *16th International Society for Music Information Retrieval Conference*, ser. ISMIR, 2015.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 448–456.

[21] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015.

[22] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] W.-N. Hsu, Y. Zhang, and J. Glass, "Learning latent representations for speech generation and transformation," in *Interspeech*, 2017, pp. 1273–1277.