# An Unsupervised Method to Select a Speaker Subset from Large Multi-Speaker Speech Synthesis Datasets

*Pilar Oplustil Gallegos, Jennifer Williams, Joanna Rownicka, Simon King*

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom

P.S.Oplustil-Gallegos@sms.ed.ac.uk

## Abstract

Large multi-speaker datasets for TTS typically contain diverse speakers, recording conditions, styles and quality of data. Although one might generally presume that more data is better, in this paper we show that a model trained on a carefully-chosen subset of speakers from LibriTTS provides significantly better quality synthetic speech than a model trained on a larger set. We propose an unsupervised methodology to find this subset by clustering per-speaker acoustic representations.

**Index Terms**: speech synthesis, data, clustering, speaker representation, sequence-to-sequence models, multi-speaker

## 1. Introduction

State-of-the-art Text-to-Speech (TTS) synthesis models have achieved near human quality, especially for intelligibility, by leveraging advances in deep learning which usually require training on very large datasets to perform well. These very large speech datasets frequently contain "found data", e.g. data of varying recording quality that was not elicited with the purpose of training TTS systems. However, found data can be very inconsistent in terms of sample quality, recording condition, speaker variety, and speaking style, among other factors [1].

Although in machine learning – and deep learning in particular – it is considered a general rule that more training data usually improves model performance, the spurious features of found data can significantly affect the training and final quality of TTS models [1]. Earlier TTS frameworks, notably statistical parametric speech synthesis (SPSS), have benefited from efforts to understand the interaction of the data characteristics with the training of the models [2]. However, for TTS frameworks based on neural networks, there is not yet enough work to explain how the latest sequence-to-sequence (S2S) architectures interact with particular data qualities and quantities.

Data selection is one tool for understanding this interaction. Diversity of speakers is one of the most important factors of variation in the data, and the easiest way to obtain more data is by combining speakers. While it is widely believed that inherent characteristics of some speakers are better or worse for TTS models [3], we don't yet have a clear understanding of what these characteristics are. Moreover, when we are dealing with large datasets, manual curation of the raw data is not feasible, making automated approaches essential.

Our goal is to train a TTS model from a large found database (here, LibriTTS [4]) that leads to the most natural synthetic speech, having no particular target speaker identity in mind. We describe a new unsupervised speaker selection method based on clustering per-speaker acoustic representations. We use this to identify a subset of speakers for training and show that this subset leads to significant improvements in quality compared to training our S2S TTS model using the entire dataset.

## 2. Related Work

Data selection techniques have been widely explored in TTS research. In general, there are two approaches: 1) extracting acoustic features from the **speech signal** for statistical analysis of diverse aspects of speech such as pitch or speech rate [5], hypo- and hyper-articulation [6], mel cepstral distortion [1]; 2) measuring **data quality** in terms of alignment errors [7], phonetic coverage [1] or noise [4]. Although most of these methods use per-utterance statistics, [2] showed that data selection at the speaker level outperforms utterance-level selection. Their approach ranks speakers into three groups (high, medium, low) using extracted acoustic values and automatic transcription word error rates (WER). Models trained on the *speaker group* with the lowest WER resulted in higher intelligibility than models trained on the lowest WER *utterances*.

It is not clear if those improvements came from selecting speakers with overall high quality or from finding a subset that is homogeneous. [8] experimented with several criteria to group similar speakers in order to train average speaker models for Hidden Markov Model-based TTS (HMM-TTS). Grouping speakers by listeners' judgements of perceived similarity outperformed any signal-based criteria. [9] obtained positive results for methods finding homogeneous subsets of speakers, with the aim of training multi-language HMM-TTS. The two-step approach merges models based on gender, age and smoking habits, then uses 6 utterances per speaker to perform hierarchical agglomerative clustering of speakers, by comparing distances between Gaussian Mixture Models trained on the data.

Most speaker selection methods are extrinsic and therefore should also be useful for the latest TTS S2S models trained on large multi-speaker corpora. In recent work, [10] aimed to find the smallest amount of multi-speaker data needed in addition to the limited data from a single target speaker. They showed that small multi-speaker datasets can outperform larger speaker-dependent datasets, where most of the improvements come from learning more stable models.

## 3. Data Preparation

LibriTTS[1] [4] is a large multi-speaker dataset, recently released by Google, created from the LibriSpeech dataset for the purpose of training TTS models. LibriSpeech comprises audiobooks read and recorded by non-professional speakers, generally in sub-optimal recording conditions. We worked with the two "clean" training subsets defined by LibriTTS that contain a total of 245 hours of speech from 1 151 speakers. We first applied the two pre-processing stages detailed in the next two subsections.

[1] http://www.openslr.org/60/

### 3.1. Amount of Data Filter

As described in the original LibriTTS paper [4], the amount of data per speaker in the corpus is highly unbalanced, with a median of just 15 minutes. Our method computes a per-speaker representation of their characteristics, so we only retained speakers with at least 20 minutes of data in order to obtain a robust representation. Furthermore, for the purposes of our experiments, we desired relatively balanced data so that no single speaker dominates the data (and thus skews results), so also discarded speakers with more than 30 minutes. This resulted in 120 speakers, each with 20-30 minutes of speech, resulting in 64.3 hours of speech at this stage of data preparation.

### 3.2. Outlier Removal

In the early stages of prototyping our clustering method, we noticed that results were highly affected by outlier speakers, i.e., those with a very different acoustic representation to other speakers. We found that their audio recordings had a bandwidth substantially lower than half the sampling rate (the Nyquist frequency), suggesting that they were made on a device with a digitally-limited bandwidth (e.g., certain USB headsets), or had at some unknown stage in their history been *upsampled*. To identify and remove speakers with any such data, we used long-term spectral analysis, resulting in a final dataset of 88 speakers amounting to 33.8 hours of the 64.3 hours from above.

## 4. Finding Subsets of Speakers

Our goal is to find a subset of speakers for training a TTS model that leads to the highest quality synthetic speech. The proposed method consists of two parts. First, we extract a feature vector from the speech signal of each utterance, average them within speakers, and obtain per-speaker representations. Second, we perform unsupervised clustering of those speaker representations. The data from all speakers within a cluster is combined into a training set for a TTS model. Thus we train multiple cluster-dependent models. There are many possible options for the feature vectors; we explored three, described in Section 4.1. The complete method is summarised in Figure 1.

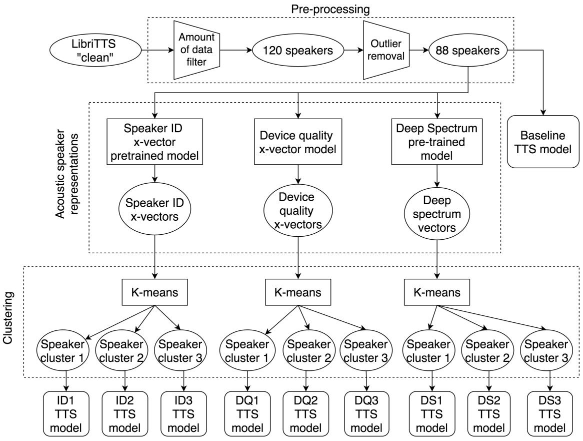


Figure 1: *An overview of the complete method*

### 4.1. Acoustic Speaker Representations

#### *4.1.1. Speaker Identity X-Vectors*

Speaker vectors aim to maximize variance between speakers, while minimizing within-speaker variability such as recording session and speaker mood. We chose this representation as a starting point condition: it should represent differences and similarities between speakers but perhaps not fully capture aspects of recording quality. We extracted utterance-level speaker x-vectors from LibriTTS data using a pre-trained[2] speaker x-vector model [11] and the Kaldi framework [12]. This pre-trained model uses a time-delay neural network (TDNN) and was originally trained using VoxCeleb 1 & 2 corpora, on which it obtains an EER of 3.1% and minDCF(0.01) of 0.33 for speaker ID. For LibriTTS, we obtained an EER of 10.23% and minDCF(p-target=0.01) of 0.68. This indicates that our x-vectors are reasonable, despite potential data mismatch between VoxCeleb and LibriTTS. The $512$-*dim* utterance-level x-vectors were averaged within speakers to obtain per-speaker Speaker Identity representations.

#### *4.1.2. Device Quality X-Vectors*

The idea of using x-vectors to model device quality, rather than speaker identity, was first introduced in [13]. They were later shown to be useful for automatically predicting naturalness of synthetic speech [3]. We hypothesized that this type of speech representation might be able to capture differences in recording quality. We extracted device quality x-vectors using a pre-processing script[3] originally intended for naturalness prediction. We trained a TDNN x-vector extractor on the Physical Access (PA) simulated dataset from the ASV Spoofing Challenge 2019 [14], a dataset created to facilitate research on countermeasures for replay spoofing attacks. The training labels, rather than speaker labels, represented the quality of the replay device (perfect, high, low, or 'not replayed'). The quality of the device depends on the bandwidth captured, its lower bound, and its linear/non-linear power difference. The $512$-*dim* utterance-level device quality x-vectors were averaged within speakers to obtain per-speaker Device Quality representations.

#### *4.1.3. Deep Spectrum Vectors*

The previous two representations are derived from supervised techniques aiming to capture only speaker identity, or only recording quality, respectively. An interesting alternative to these would be a problem-agnostic representation. One such representation is the Deep Spectrum[4], which is derived from the activations of a specific layer in a very deep image CNN [15, 16]. In our work, they come from layer *fc2* in a VGG-19 model that was pre-trained on a variety of non-speech images. The spectrogram of each utterance is scaled down to a fixed $227 \times 227$ image and passed forward through the network to obtain a Deep Spectrum vector of $4096$-*dims*. These were averaged within speakers to obtain per-speaker Deep Spectrum representations. We hypothesize that the Deep Spectrum captures noise-like artifacts at varying times and frequencies in the spectrogram, which are lost or deliberately discarded by the other representations.

[2] `https://kaldi-asr.org/models/m7`

[3] `https://github.com/rhoposit/MOS_Estimation`

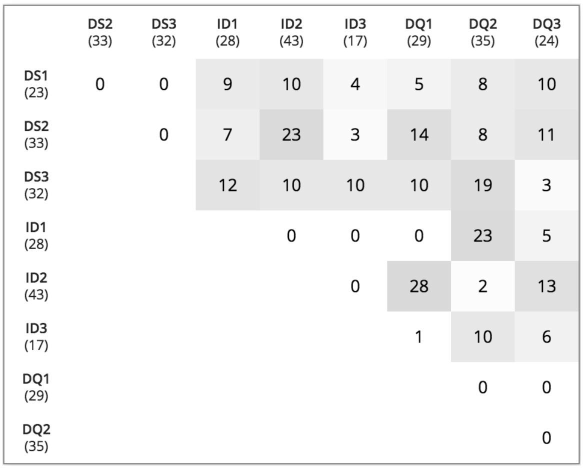
[4] `https://github.com/DeepSpectrum/DeepSpectrum`

### 4.2. Clustering

Each of the above acoustic representations was extracted for each file in the dataset, then averaged across all files from the same speaker to obtain per-speaker representations. For each representation in turn, the speakers were clustered using scikit-learn's implementation of k-means [17].

A key design choice is how many clusters to obtain. We certainly want a minimum of 3 clusters, to avoid a 2-way split purely on gender. A larger number of clusters implies less data per cluster, and – in practical terms – a less manageable number of systems to build and compare. We ran the clustering multiple times with different random seeds and for 3 to 5 clusters. We compared the results using the clustering performance evaluation metrics (Calinski-Harabasz Index and Silhouette Coefficient[5]) that the same library provides for when the ground truth labels are not known and chose k=3.

Since we will be comparing TTS models each trained on the data of a single cluster, reasonably equally-sized clusters are desirable. For the Deep Spectrum and Device Quality features, the resulting clusters came out balanced. For the Speaker Identity vectors, across all runs with different random seeds, one of the clusters was consistently larger, so we selected the run that gave the best balance. Figure 2 shows the size of each cluster and the overlap between speakers. There is no ordering to the clusters: e.g., cluster 1 for Deep Spectrum has no relationship to cluster 1 for Device Quality.



| | DS2<br>(33) | DS3<br>(32) | ID1<br>(28) | ID2<br>(43) | ID3<br>(17) | DQ1<br>(29) | DQ2<br>(35) | DQ3<br>(24) |
|---|---|---|---|---|---|---|---|---|
| DS1<br>(23) | 0 | 0 | 9 | 10 | 4 | 5 | 8 | 10 |
| DS2<br>(33) | | 0 | 7 | 23 | 3 | 14 | 8 | 11 |
| DS3<br>(32) | | | 12 | 10 | 10 | 10 | 19 | 3 |
| ID1<br>(28) | | | | 0 | 0 | 0 | 23 | 5 |
| ID2<br>(43) | | | | | 0 | 28 | 2 | 13 |
| ID3<br>(17) | | | | | | 1 | 10 | 6 |
| DQ1<br>(29) | | | | | | | 0 | 0 |
| DQ2<br>(35) | | | | | | | | 0 |

Figure 2: *Overlap of speakers between clusters. "DS" are clusters using Deep Spectrum features; "ID", Speaker Identity x-vectors, and "DQ", Device Quality x-vectors. The number in parentheses below each system indicates the size of the cluster, while the squares show the number of speakers that overlap for each pair of clusters.*

## 5. System Descriptions

### 5.1. Architecture

Figure 1 shows that we trained one model for each cluster of each representation, plus a baseline model using all the data. All models used the Ophelia[6] version of Deep Convolutional TTS (DCTTS) [18], an S2S architecture using only convolutional layers. This architecture was selected for its fast speed of training compared to, e.g., Tacotron 2 [19]. Since our speaker subset selection method is extrinsic to the model, we expect our findings to generalise to other architectures.

DCTTS has two main components: (1) the Text2Mel (T2M) model maps a phoneme sequence to an 80-band mel-scale spectrogram at a low time resolution of 20 frames per second; it comprises a text encoder, an audio encoder, an audio decoder and an attention module; (2) the Spectrogram Super Resolution Network (SSRN) upsamples that mel-scale spectrogram to a 1025-band linear spectrogram at 80 frames per second. Waveform generation was performed using the Griffin-Lim algorithm [20]. All models were trained with phonemic transcriptions produced from the input text by Festival [21] using the CMU lexicon. All T2M models were multi-speaker, we provide speaker code embeddings to every layer of T2M. We used the same SSRN model for all systems, trained on the whole dataset without speaker information.

### 5.2. Cluster-dependent TTS Systems

We trained one system for each of the resulting 9 clusters of speakers. The speaker representations used to form clusters of speakers are *not* used by the TTS models. Figure 2 shows the number of speakers used to train each of the models.

All the cluster-dependent models were trained for 1500 epochs. Although this means a different number of weight updates per model (as the total number of samples differs) we observed that the output of the models is stable even when 'over-trained'. For each speaker, one sentence (the same across all systems) was randomly held out for the listening test.

### 5.3. Baseline TTS System

The baseline system was trained on the full dataset from all 88 speakers remaining after the data preparation in Section 3. To ensure fair comparison with the above systems, it was trained until perceived quality converged, which took 4000 epochs.

## 6. Evaluation

Recall that our goal is to achieve the best quality synthetic speech. Since we have no particular target speaker in mind, and to reduce the number of systems to be evaluated in the subsequent listening test, we identified the best cluster-dependent model per representation, through expert listening to random sentences whilst varying the speaker code input to the T2M model, identifying the most stable model. For Deep Spectrum the best system was DS2, for Speaker Identity it was ID2 and for Device Quality it was DQ1.

In the following listening test, we evaluated: (1) Copy Synthesis by passing ground-truth 80-band mel-scale spectrograms at a low time resolution of 20 frames per second (Section 5.1) through SSRN followed by the Griffin-Lim algorithm, (2) Baseline, (3) DS2, (4) ID2 and (5) DQ1.

To control for listeners' preferences for some speakers over others, the listening test used synthetic speech generated using the speaker codes of the 14 speakers in the intersection of all the above systems, of the single held-out sentence mentioned earlier. We implemented a MUSHRA-like listening test.[7] Copy Synthesis was provided as a reference sample, not to be rated. Copy Synthesis was also included as the hidden refer-

[5] `https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation`

[6] `https://github.com/CSTR-Edinburgh/ophelia`

[7] Listen to the samples at: `https://pilarog.github.io/`

ence, among the other four samples options in a random order, presented to listeners for side-by-side rating. Listeners were instructed to rate each sample from 0 to 100 according to its quality given the reference, and in doing so also to find the hidden reference and give it a score of 100.

The test was implemented in Qualtrics[8] and participants who self-identified as US citizens and native speakers of US English were recruited via Prolific Academic[9]. Any participant who scored any of the references lower than 50 was discarded and replaced, other participants were retained, until we reached a pre-set target of 20 retained participants.

## 7. Results

Results are shown in Figure 3. We tested the scores for normality with the Shapiro-Wilk test. As scores were not normally distributed, we compared their averages with the Wilcoxon signed-rank test, and tested for statistical significance after Bonferroni correction (alpha=0.05).

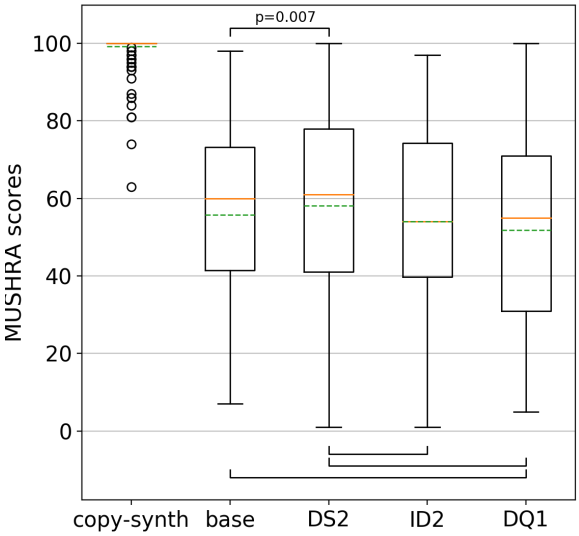


Figure 3: *Horizontal lines show significant pairs. The p-value given is for the system that is significantly better than baseline. Inside each box, the solid line indicates the median and the dashed line is the mean.*

The best Deep Spectrum system, DS2, was significantly better than baseline at $p \approx 0.007$. The other two cluster-dependent systems were significantly inferior to the baseline.

The results demonstrate speaker selection has the potential to improve quality. Using data from 33 speakers selected using Deep Spectrum speaker representations proved to be significantly better than a baseline system using data from 88 speakers, and better than a system trained on data from 43 speakers selected using Speaker Identity representations, and one trained on data from 29 speakers selected using Device Quality representations.

## 8. Discussion

One explanation for the good result using the Deep Spectrum representation is its ability to capture more speaker characteristics and other information from the spectrogram than Speaker Identity x-vectors. Specifically, we speculate it may capture speaking rate and recording conditions.

Clustering using the Deep Spectrum representation is only one way to identify speaker subsets, and was tested only on one dataset. It is not clear how much the result depends on the characteristics of the data, such as the nature and distribution of outliers, which will differ from one dataset to another. However, Deep Spectrum features are generic and not task-specific, which suggests they should also work on other datasets.

We also trained other systems that we didn't include in our formal evaluation as they were evidently worse than the baseline. These systems: random selection of 27 speakers (average number of speakers in the cluster-dependent models) to control for data size; single gender subsets; T2M models that encoded the cluster label and distance from centroid instead of accepting a speaker label; fine-tuning both the baseline model and the cluster-dependent models to each of the speakers used in evaluation. We chose $k = 3$ clusters, but found informally that $k = 5$ also appears to work well.

Evaluating the output of multi-speaker models, especially when no particular target speaker or use case is in mind, presents some challenges. It is more common to use multi-speaker datasets to provide data additional to a limited quantity of data from the desired target speaker; in that case, evaluation can be for that target speaker only.

During the design of our evaluation we considered several ways to compare our models to the baseline. We could have compared speech generated for all the speakers that could be produced from each and every cluster-dependent model against speech generated by the baseline for the same speaker. However, this would be a costly evaluation. Instead, we limited the evaluation to only our best models, thus limiting the number of speakers generated. By using the same speakers across all models, we aimed to avoid two problems: 1) listeners' preferences for some speakers over others; 2) variations in quality from a single model as the speaker is varied. These factors could be important in other evaluations.

## 9. Conclusions and Future Work

We have demonstrated that training a system on a selected subset of speakers improves synthetic speech quality compared to a baseline trained with a larger dataset: more data was not better.

We proposed a simple unsupervised method to find this speaker subset by clustering per-speaker representations, and found that Deep Spectrum features worked well.

In future work, we would like to automate the only step of the process that requires manual evaluation: finding the best cluster-dependent model out of the $k$ such models per speaker representation. We also plan to replicate the experiments using another S2S architecture such as Tacotron 2, and to employ other large multi-speaker datasets.

**Acknowledgements:** this work was supported in part by: ANID, Becas Chile, nº 72190135; the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh; and a PhD studentship from the DataLab Innovation Centre, Ericsson Media Services, and Quorate Technology.

[8] `https://www.qualtrics.com/`

[9] `https://www.prolific.co/`

# 10. References

[1] P. Baljekar and A. W. Black, “Utterance selection techniques for tts systems using found speech.” in *SSW*, 2016, pp. 184–189.

[2] K.-Z. Lee and E. Cooper, “A comparison of speaker-based and utterance-based data selection for text-to-speech synthesis,” *Interspeech 2018*, vol. 12873, 2018.

[3] J. Williams, J. Rownicka, P. Oplustil, and S. King, “Comparison of Speech Representations for Automatic Quality Estimation in Multi-Speaker Text-to-Speech Synthesis,” *Speaker Odyssey*, 2020.

[4] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, “Libritts: A corpus derived from librispeech for text-to-speech,” *arXiv preprint arXiv:1904.02882*, 2019.

[5] E. Cooper, Y. Levitan, and J. Hirschberg, “Data selection for naturalness in hmm-based speech synthesis,” in *Speech Prosody*, 2016.

[6] E. Cooper and X. Wang, “Utterance selection for optimizing intelligibility of tts voices trained on asr data,” *Interspeech 2017*, vol. 1, 2017.

[7] F.-Y. Kuo, S. Aryal, G. Degottex, S. Kang, P. Lanchantin, and I. Ouyang, “Data selection for improving naturalness of tts voices trained on small found corpuses,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 319–324.

[8] R. Dall, C. Veaux, J. Yamagishi, and S. King, “Analysis of speaker clustering strategies for hmm-based speech synthesis,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[9] A. W. Black and T. Schultz, “Speaker clustering for multilingual synthesis,” in *Multilingual Speech and Language Processing*, 2006.

[10] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and V. Klimkov, “Effect of data reduction on sequence-to-sequence neural tts,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7075–7079.

[11] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldi speech recognition toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[13] J. Williams and J. Rownicka, “Speech replay detection with x-vector attack embeddings and spectral features,” *Proc. Interspeech 2019*, pp. 1053–1057, 2019.

[14] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, “Asvspoof 2019: Future horizons in spoofed and fake audio detection,” 2019.

[15] N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. W. Schuller, “An Image-based Deep Spectrum Feature Representation for the Recognition of Emotional Speech,” in *Proceedings of the 2017 ACM on Multimedia Conference - MM ’17*. Mountain View, California, USA: ACM Press, 2017, pp. 478–484. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3123266.3123371

[16] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, and B. Schuller, “Snore Sound Classification Using Image-Based Deep Spectrum Features,” in *Interspeech 2017*. ISCA, Aug. 2017, pp. 3512–3516. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0434.html

[17] 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.

[18] H. Tachibana, K. Uenoyama, and S. Aihara, “Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4784–4788.

[19] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[20] 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.

[21] R. A. Clark, K. Richmond, and S. King, “Multisyn: Open-domain unit selection for the festival speech synthesis system,” *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.