Acta Cybernetica **25** (2021) 257–269.

Speech De-identification with Deep Neural Networks*

Ádám Fodor[a], László Kopácsi[a], Zoltán Á. Milacski[b], and András Lőrincz[c]

Abstract

Cloud-based speech services are powerful practical tools but the privacy of the speakers raises important legal concerns when exposed to the Internet. We propose a deep neural network solution that removes personal characteristics from human speech by converting it to the voice of a Text-to-Speech (TTS) system before sending the utterance to the cloud. The network learns to transcode sequences of vocoder parameters, delta and delta-delta features of human speech to those of the TTS engine. We evaluated several TTS systems, vocoders and audio alignment techniques. We measured the performance of our method by (i) comparing the result of speech recognition on the de-identified utterances with the original texts, (ii) computing the Mel-Cepstral Distortion of the aligned TTS and the transcoded sequences, and (iii) questioning human participants in A-not-B, 2AFC and 6AFC (Alternative Forced-Choice) tasks. Our approach achieves the level required by diverse applications.

Keywords: speech processing, voice conversion, deep neural network, text-to-speech, speaker privacy

*The research has been supported by the European Union, the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program and by the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme through the "Application Domain Specific Highly Reliable IT Solutions" project and co-financed by the European Social Fund (EFOP-3.6.3-16-2017-00002, EFOP-3.6.3-VEKOP-16-2017-00001).

[a] Equal contributions. Department of Artificial Intelligence, Eötvös Loránd University, Budapest, Hungary, E-mail: {foauaai, kopacsi}@inf.elte.hu, ORCIDs: 0000-0001-7370-930X and 0000-0003-2387-2015

[b] Department of Artificial Intelligence, Eötvös Loránd University, Budapest, Hungary, E-mail: miztaai@inf.elte.hu, ORCID: 0000-0002-3135-2936

[c] Corresponding author, Department of Artificial Intelligence, Eötvös Loránd University, Budapest, Hungary, E-mail: lorincz@inf.elte.hu, ORCID: 0000-0002-1280-3447

DOI: 10.14232/actacyb.288282

1 Introduction

Cloud-based speech services have improved recently due to the large amount of voice data that is exploited by deep learning technology [1, 3], giving rise to superhuman performance in several tasks. Consequently, it seems reasonable to use such utilities in practice.

Unfortunately, many speech applications involve legal concerns regarding privacy. Several methods have been proposed to eliminate personal information from samples without spoiling the linguistic content before uploading. We should also mention, that in many cases the private information is carried by the linguistic content and not by the voice of the speaker. For example, when a doctor dictates medical records, the private content is the medical content and not the identity of the doctor. But in the case of diagnostic sessions with autistic people, it is the speaker whose identity should remain hidden. If an external ASR is used on the transformed speech of a patient, the identity will remain concealed, and the linguistic content can be generated safely.

Voice conversion (VC) operates by altering certain features of human speech [31]. Voice transformation (VT) converts the signal as if it was uttered by a target speaker [23]. De-identification is the process that intends to remove any personal information from the data that could be associated with identity. VC and VT may be applied to solve de-identification, but the papers in the literature suffer from several flaws: the VC algorithm in [22] is approximately invertible and relies on a good voice transformer, while VT [23, 29] requires data from pairs of speakers and is unable to anonymize the target speaker.

Our contributions are as follows. For de-identification, we propose to transform utterances to a generic voice of a Text-to-Speech (TTS) engine, by taking advantage of utterance-text sample pairs. We use an end-to-end trainable Deep Neural Network (DNN) to learn the many-to-one VT task. We suggest to learn the mapping at vocoder level. We show that the trained network gives rise to tolerable distortions at utterance level by conducting two experiments: comparing the outputs of Google's Automatic Speech Recognition (ASR) system for the original TTS output and the de-identified utterance and measuring the Mel-Cepstral Distortion (MCD) [19]. To confirm de-identification success, we further performed three kind of perceptual listening studies with human subjects (A-not-B test: distinguishing transformed utterances of different speakers, 2-Alternative Forced-Choice (2AFC) test: classifying utterances from female/male speakers, and 6-Alternative Forced-Choice (6AFC) test: estimating the number of speakers). Our proposal is irreversible and it requires only speech-transcript sample pairs for training, which are readily accessible in the literature. We argue that our method performs favorably compared to several baseline methods.

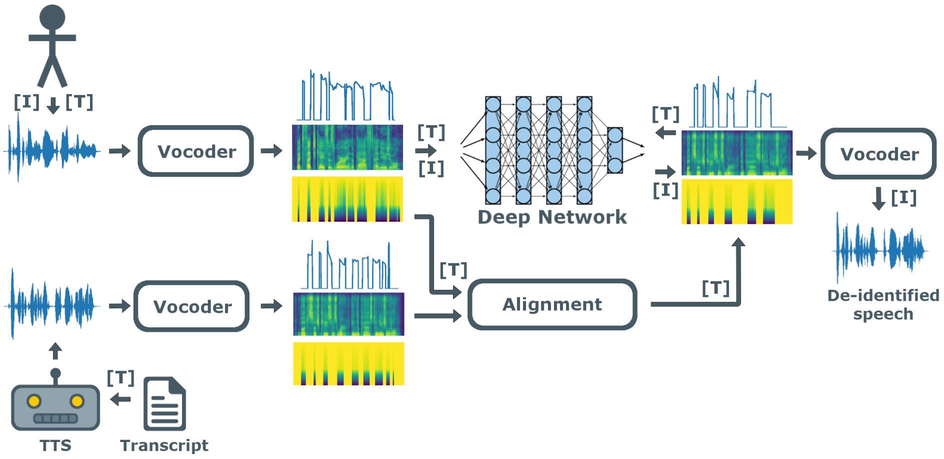

Figure 1: Schematic diagram of our proposed method. Training [T]: vocoded human voice is input to a Deep Neural Network (DNN) that is trained to approximate the aligned TTS output. Inference [I]: vocoded human voice is de-identified by the DNN and transformed back to utterances by the vocoder.

2 Related work

De-identification can be solved by either VC or VT methods. A subset of the literature focuses on classical algorithms instead of leveraging the potential of DNN architectures, and hence fail to produce state-of-the-art speech quality. The so-called transterpolation VC technique gave rise to significant improvements over diverse VC methods as reviewed in [16]. Another VC approach exploits the two-step procedure of an ASR system followed by a TTS [17]. However, due to the method of the conversion, the latter cannot take advantage of the superior performance of cloud-based ASR systems. A potential problem of VT methods is that the target speaker is not generic and hence it cannot be anonymized. Generic is meant here as monotonic and "robotic", which does not contain any prosody. Neural TTS systems nowadays are realistic enough, that it may include unintended prosody, which makes the transformation harder. The problem can be resolved by converting to an average voice, however, we decided to go with a TTS system instead of generating an average voice. In addition, VT methods need speech corpora of the original speaker and the target speaker, too. To avoid the need for a parallel corpora an approach that used a pool of pre-trained transformations between a set of speakers was put forth in [22]. A transformation function was applied to the source and the target speakers based on speaker similarity and dissimilarity, respectively. By applying several sound distortion algorithms de-identification was achieved and the transformation could be reversed, offering several advantages at the cost of vulnerability.

In contrast, several recent works propose DNNs for many-to-one VC and VT.

For an overview, see [24]. A VC method using Mel-Cepstral inputs for deep autoencoders was introduced in [23]. Speaker-dependent Conditional Restricted Boltzmann Machine (CRBM) was applied using Mel-Frequency Cepstral Coefficients (MFCCs) and deltas for solving the VC task for each speaker pair in [29]. An autoencoder-based VT approach was proposed to reduce the required size of the data sets and to shorten conversion time in [32]. A VT method that generates a one-to-one speaker-dependent DNN using the weights of a speaker-independent DNN was suggested in [21]. Spectral envelope, fundamental frequency ($F0$), intensity trajectory and phone duration were converted in [30] subject to an ℓ_1 norm constraint during pre-training. Nevertheless, all of these methods restrict themselves to the case of VC and VT, without using transcript data. In this paper, we directly tackle de-identification and propose to use textual data as well besides the original speech for training, which are largely available online.

3 Proposed method

We describe the features set, the pre-processing steps and the DNN architectures in detail here. The de-identification pipeline can be seen in Fig. 1. To differentiate processes, training and inference are marked with “T” and “I”, respectively.

3.1 Measures

In speech recognition, the standard measure is the word error rate (WER), defined as the edit distance between the true word sequence and the most probable word sequence emitted by the transcriber. However, it is not ideal for short sequences. In cases of 1-word sequences, where the transcriber recognizes the expected word, but mistakes 1-2 letters, the measure is 0. This is not representative enough, that is why the Levenshtein distance is used instead.

Levenshtein distance is a measure of the similarity between two strings, which we will refer to as the S source string and the T target string. The distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to transform S into T. This measure is also called character error rate (CER) or letter error rate (LER). Letter accuracy rate (LAR) will be used in this document later on:

$$LAR(S,T) = 1 - LER(S,T) \tag{1}$$

3.2 Feature extraction

In order to find the best feature set, we compared multiple vocoder systems in terms of speech synthesis quality (Ahocoder [7], MagPhase [8], PulseModel [6], WORLD [28] and STRAIGHT [18]) using the TIMIT [35] test set. We took each sample and extracted the vocoder parameters from them. We also experimented with converting the spectral parameters into a mel-cepstral representation. Following this, we re-synthesized the samples, and measured the mean Letter Accuracy Rate (LAR) values between the predicted transcript of Google Cloud Speech-to-Text system and the transcripts provided with the TIMIT corpus.

During our experiments, we observed that using mel-cepstral representation during encoding produced more favorable results.

The LAR of the test set was 97%. By applying vocoder systems the LAR was barely affected, in every case, the relative degradation was less than 2%. In subsequent sections, we used the python wrapper of WORLD vocoder called PyWorld, because of its low computational requirements, continuous support and easy usability. The extracted features are the estimation of the fundamental frequency ($F0$), spectral envelope and aperiodicity.

$F0$ is the fundamental frequency of the vibration of our vocal folds. We perceive it as pitch. The $F0$ contour is estimated with DIO [27]. To improve the noise robustness of DIO, we also applied StoneMask pitch refinement algorithm.

Let us introduce x_n the subsampled audio signal, and X_k the frequency spectrum [15], which is the discrete Fourier transform (DFT) of a signal defined for $k = 0, 1, ..., N-1$.

We can compute the *magnitude spectrum* M_k, the *phase spectrum* Φ_k and the *power spectrum* P_k using the following equations:

$$M_k = |X_k| \tag{2}$$

$$\Phi_k = \arctan \left| \frac{Re(X_k)}{Im(X_k)} \right| \tag{3}$$

$$P_k = Re(X_k)^2 + Im(X_k)^2. \tag{4}$$

To convert the values of k into actual frequencies we can use the following formula:

$$f = \frac{k \cdot f_s}{N}, \tag{5}$$

where f_s is the sampling frequency and N is the number of samples.

The spectral envelope [15] is the contour of the magnitude spectrum, which is estimated with CheapTrick [25]. The shape of this curve approximates the frequency response of the vocal tract.

The aperiodicity is defined as the power ratio between the speech signal and the aperiodic component of the signal. It is extracted by D4C algorithm [26].

The cepstrum is the inverse discrete Fourier transform (IDFT) of the logarithm of the audio signal's P_k power spectrum:

$$C_n = IDFT(\log(P_k)), \tag{6}$$

where $k = 0, 1, ..., N-1$. It gives us a more compact, low dimensional, decorrelated representation.

With mel-cepstral analysis [10, 33] we can warp the frequency scale and compress the frequency coefficients. With the following formula we can calculate the

M-th order mel-cepstral coefficients:

$$\log\left(X(e^{-i\omega})\right) = \sum_{m=0}^{M} \tilde{c}_m e^{-i\tilde{\omega}}, \tag{7}$$

where $X\left(e^{-i\omega}\right)$ is the discrete Fourier transform of x_n, and $\tilde{c}_m$ is the m-th order mel-cepstral coefficients.

$$\tilde{\omega} = \tan^{-1} \frac{\left(1-\alpha^2\right)\sin\omega}{(1+\alpha^2)\cos\omega - 2\alpha}. \tag{8}$$

is the phase response of an all-pass filter. It gives us the warped frequency scale. The $\alpha \in [-1, 1]$ is the all-pass constant, which gives the warping characteristic. With the right α value, which is chosen to be 0.58 based on Merlin [34] suggestion, the mel-scale becomes a good approximation to the human auditory frequency scale.

The following features are used as inputs and targets to several CNN and ConvLSTM architectures: Mel-Cepstral Coefficients (MCEP) and band aperiodicity (BAP) were calculated using Eq. (7) from the spectral envelope and aperiodicity, respectively. Linear interpolation of $\log F0$ was calculated from $F0$. We also applied a thresholded binary voiced/unvoiced (V/UV) mask. Dynamic features (delta and delta-delta) were determined using MCEP and BAP.

3.3 Data sets

We employed the following benchmark corpora in our voice conversion evaluations.

TIMIT [35] is used frequently for comparing different machine learning methods. This database is attractive for verification and parameter tuning of the algorithms since it is relatively small, but still has phonetically diverse samples. The training set has 462 speakers, 8 utterances/speaker. The validation set consists of 50 speakers, totally 400 utterances, and the test set contains 192 sentences from 24 speakers. Each utterance is approximately 3.5 seconds long on average. The speakers also represent 8 major dialect regions of the United States.

NTIMIT [9] is a multi-speaker speech database with phone bandwidth that is derived from TIMIT by adding noise to the samples.

3.4 Pre-processing

The target TTS voices are generated with the Festival Speech Synthesis System [4] using the transcripts of the datasets. The choice of Festival was motivated by comparing several TTS systems and supported generic voices. The TTS generated sound files were aligned to match with the corresponding sound files produced by the speaker. We used Dynamic Time Warping (DTW) for the alignments.

In case of the TIMIT [35] and NTIMIT [9] data sets, audio normalization was unnecessary. The train-dev-test speakers are carefully separated. We also augmented data by applying speed warping factors to enlarge the TIMIT dataset.

Vocoder features were extracted from both the original speakers' and the TTS voice. The interpolated $\log F0$, the V/UV mask vector and delta and delta-delta features are calculated. Multiple combinations of these features are tested as inputs for different network architectures. Z-score normalization was applied to all of the calculated features, resulting in zero mean and unit variance.

3.5 Modeling feature transformation

For feature transformation, various deep learning architectures were applied and compared: (1) we experimented with an architecture, which we refer to as Dense, having four 1,024 unit dense layers. (2) we used a Convolutional Neural Network (ConvNet) with two 1D convolutional layers of 512 units and kernel width 7 and stride 1, and two 1,024 unit dense layers. (3) tried a model, which we call C-BLSTM, having three batch normalized 256 unit 1D convolutional layers with kernel width 3, one 128 unit BLSTM layer and two 512 unit dense layers was also tested, where the first dense layer was batch normalized. Finally, two state-of-the-art architectures based on (4) Residual Networks (ResNet) [11] and (5) Wav2Letter [20] were also evaluated.

Within all networks, we used ReLU activation functions and dropout layers with probability between 0.2 and 0.3, in addition to adding a final dense output layer on top with linear activation.

4 Results

Here, we present our results. We note that before training on larger data sets, a sanity check was carried out by varying the size of TIMIT using the Dense, the ConvNet and the C-BLSTM architectures, confirming that augmentation improves the results.

4.1 Experimental setup

Festival 2.5 with "voice_cmu_us_rms_cg" was used to generate the target TTS utterances in the experiments. We applied PyWorld [13] and SPTK 3.9 [14] for feature extraction. We implemented the neural networks in Python using the Keras [5] deep learning framework backed by Tensorflow [2]. We used early stopping with patience 10 and Adam optimizer with its default parameters. The loss function is Mean Squared Error (MSE), however, if interpolated $\log F0$ with V/UV mask is used as input, it is not used in loss calculation. We trained the models on TIMIT [35] dataset.

Regardless of the model, the network took the whole sequence as input, and its target was the DTW transformed TTS utterances.

Our implementation is available on GitHub[1].

[1] `https://github.com/lkopi/deidentification`

4.2 Objective evaluation

To show that the trained network producing quality outputs at utterance level, we conducted two quantitative experiments.

The first experiment concerned ASR accuracy. We uploaded the de-identified network outputs to Google Cloud Speech-to-Text system and quantified the difference between the true and the predicted transcript.

Precision values are given in Table 1. First, we measured the TTS voices (without applying DTW). It reached 97% ASR accuracy. The DTW aligned TTS reached 93%. ASR accuracy, i.e., a 4% drop was found. The tested DNNs performed well on the TIMIT data set, producing well understandable speech after synthesization. Intriguingly, they mostly achieved 80-85% ASR accuracy, only, a relatively large drop compared to the 93% goal.

Cloud-based ASR services improve constantly. One can expect that the performance will increase over time. We found that the ResNet and Dense architectures marginally outperformed the others.

In the second experiment, we evaluated Mel-Cepstral Distortion (MCD) [19] between de-identified Dense network outputs and the aligned TTS signals. MCD is given by the following equation:

$$MCD[dB] = \frac{1}{N}\sum_{n=1}^{N}\frac{10}{\log 10}\sqrt{2\sum_{d=1}^{D}(c_{n,d} - \hat{c}_{n,d})^2}, \tag{9}$$

where N is the number of frames in the analysis, D is the number of coefficients, $c_{n,d}$ and $\hat{c}_{n,d}$ are the dth coefficient of the nth frame of the target and the predicted MCEP vector, respectively.

The final MCD values were obtained by averaging over all 1,680 test sample pairs of the TIMIT corpora using the Dense architecture. The mean and standard deviation values are presented in Fig. 2 for our method (last column) together with values available in the literature. Our method seemingly outperformed its baselines, however other methods listed in the figure were trained on different datasets, so

Table 1: Details of the objective evaluations on the TIMIT database. The average and the standard deviation of the Letter Accuracy Rate (LAR) measured with Google Cloud Speech-to-Text system is presented for the proposed architectures and input features. Notation: "il$F0$" means interpolated $\log F0$.

Method	Architecture	ASR (LAR) precision using the following features:		
		$\log F0$ + MCEP + BAP	il$F0$ + MCEP + BAP	il$F0$ + MCEP + BAP + deltas
Dense	5 dense	0.77 ±0.17	**0.85 ±0.15**	**0.84 ±0.16**
ConvNet	2 conv + 3 dense	0.76 ±0.17	0.82 ±0.17	0.82 ±0.15
C-BLSTM	3 conv + blstm + 3 dense	0.77 ±0.14	0.79 ±0.15	0.79 ±0.15
ResNet	4 residual + 3 dense	**0.79 ±0.15**	0.82 ±0.16	0.82 ±0.14
Wav2Letter	9 conv + 2 dense	0.76 ±0.16	0.79 ±0.17	0.77 ±0.16

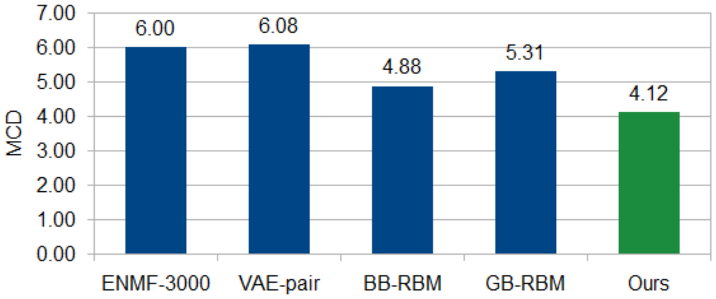

Figure 2: Mean Mel-Cepstral Distortion (MCD) values of various schemes: Exemplar-based Nonnegative Matrix Factorizations (ENMF) [12] using 3000 randomly selected source-target pair frames, VAE-pair [12], Bernoulli-Bernoulli RBM (BB-RBM) [29], Gaussian-Bernoulli RBM (GB-RBM) [29] and our proposed method.

a direct comparison is not possible. ENMF-3000 and VAE-pair were evaluated on the VCC2016 Speech Corpus, and the BB-RBM and GB-RBM were ran on ATR Japanese speech database. The figure only allows to give an impression about the performance of our method.

4.3 Subjective evaluation

To confirm that our Dense network can properly de-identify human speech, we conducted four qualitative experiments with human participants in an isolated environment. The synthesized outputs are intelligible and successfully de-identified. The collection of the subjective tests is available online[2].

In all four tests, the results were convincing, subjects performed like *random choice*, see Table 2. Participants were unable to sense any of the relevant aspects of the speakers. In both 6AFC tests, the task was to guess the number of speakers (between 1 and 6). In the first 6AFC test, none of the subjects inferred accurately. In the second 6AFC test, 4 subjects out of 22 predicted correctly, which matches random guessing within tolerance.

5 Summary

We presented a deep neural network based speech de-identification method that can map vocoder features of human speech to those of a generic TTS engine with little or minimal loss of sound quality using the TIMIT data set. The novelty of our scheme is that de-identification is based on speech-text sample pairs, which are widely available in the speech processing community. In the resulting signal,

[2] `https://people.inf.elte.hu/foauaai/deidentification`

Table 2: Results of the perceptual listening experiments. We report the average and the standard deviation of the identification accuracy.

Task	# of subj.	# of samp.	Accuracy mean ±std	Random choice
A-not-B	22	20	0.56 ±0.15	0.5
Female/Male (2AFC)		15	0.51 ±0.15	0.5
# of Speakers (6AFC)		6	0	0.16
		6	0.18	0.16

the identity of the speaker is concealed, as confirmed by our perceptual listening experiments.

A limitation of our technique is that the dynamics of the original speaker are inherited due to the application of DTW. We hypothesize that this problem may be alleviated by applying DTW in the loss function of the deep network. We leave such studies to future work.

Our technique enables privacy-aware speech recognition for adults. The proposed method is lightweight and can be used for collecting de-identified databases when the privacy of the user is important, for example in cloud-based speech services or in medical records. The fact that our method requires only speech-transcript sample pairs is a very promising aspect for deep learning, which requires large and high quality databases.

References

[1] Aaron van den, Oord, Dieleman, Sander, Zen, Heiga, Simonyan, Karen, Vinyals, Oriol, Graves, Alex, Kalchbrenner, Nal, Senior, Andrew, and Kavukcuoglu, Koray. Wavenet: A generative model for raw audio. *arXiv:1609.03499*, 2016.

[2] Abadi, Martín, Agarwal, Ashish, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from `tensorflow.org`.

[3] Amodei, Dario, Ananthanarayanan, Sundaram, et al. Deep Speech 2: End-to-end speech recognition in English and Mandarin. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 173–182, 2016.

[4] Black, Alan. The Festival Speech Synthesis System: System Documentation (1.1.1). Technical Report HCRC/TR-83, Human Communication Research Center, 1997.

[5] Chollet, Francois et al. Keras. `https://keras.io`, 2015.

[6] Degottex, Gilles, Lanchantin, Pierre, and Gales, Mark. A log domain pulse model for parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):57–70, 2018. DOI: `10.1109/taslp.2017.2761546`.

[7] Erro, Daniel, Sainz, Inaki, Navas, Eva, and Hernaez, Inma. Harmonics plus noise model based vocoder for statistical parametric speech synthesis. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):184–194, 2014. DOI: `10.1109/jstsp.2013.2283471`.

[8] Espic, Felipe, Botinhao, Cassia Valentini, and King, Simon. Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis. *Proc. Interspeech*, 2017. DOI: `10.21437/interspeech.2017-1647`.

[9] Fisher, William M., Doddington, George R., Goudie-Marshall, Kathleen M., Jankowski, Charles, Kalyanswamy, Ashok, Basson, Sara, and Spitz, Judith. NTIMIT: A phonetically balanced, continuous speech, telephone bandwidth speech database. In *Proc. IEEE ICASSP*, pages 109–112, 1990. DOI: `10.1109/icassp.1990.115550`.

[10] Fukada, Toshiaki, Tokuda, Keiichi, Kobayashi, Takao, and Imai, Satoshi. An adaptive algorithm for mel-cepstral analysis of speech. In *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, volume 1, pages 137–140. IEEE, 1992. DOI: `10.1109/icassp.1992.225953`.

[11] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] Hsu, Chin-Cheng, Hwang, Hsin-Te, Wu, Yi-Chiao, Tsao, Yu, and Wang, Hsin-Min. Voice conversion from non-parallel corpora using variational autoencoder. In *APSIPA, Asia-Pacific*, pages 1–6. IEEE, 2016. DOI: `10.1109/apsipa.2016.7820786`.

[13] Hsu, Jeremy et al. PyWorldVocoder: A Python wrapper for World Vocoder. `https://github.com/JeremyCCHsu/` Python-Wrapper-for-World-Vocoder, 2016.

[14] Imai, Satoshi, Kobayashi, Takao, et al. Speech signal processing toolkit (SPTK), 2009. `http://sp-tk.sourceforge.net`.

[15] Iser, Bernd, Minker, Wolfgang, and Schmidt, Gerhard. Broadband spectral envelope estimation. *Bandwidth Extension of Speech Signals. Lecture Notes in Electrical Engineering*, 13:67–95, 2008. DOI: `10.1007/978-0-387-68899-2_5`.

[16] Jin, Qin, Toth, Arthur R., Schultz, Tanja, and Black, Alan W. Speaker de-identification via voice transformation. *IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 529–533, 2009. DOI: `10.1109/ASRU.2009.5373356`.

[17] Justin, Tadej, Struc, Vitomir, Dobrisek, Simon, Vesnicer, Bostjan, Ipsic, Ivo, and Mihelic, France. Speaker de-identification using diphone recognition and speech synthesis. *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7, 2015. DOI: `10.1109/FG.2015.7285021`.

[18] Kawahara, Hideki. STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds. *Acoustical Science and Technology*, 27(6):349–353, 2006. DOI: `10.1250/ast.27.349`.

[19] Kominek, John, Schultz, Tanja, and Black, Alan W. Synthesizer voice quality of new languages calibrated with mean mel cepstral distortion. *First International Workshop on Spoken Languages Technologies for Under-Resourced Languages (SLTU-2008)*, pages 63–68, 2008.

[20] Liptchinsky, Vitaliy, Synnaeve, Gabriel, and Collobert, Ronan. Letter-based speech recognition with Gated ConvNets. *CoRR*, abs/1712.09444, 2017.

[21] Liu, Li-Juan, Chen, Ling-Hui, Ling, Zhen-Hua, and Dai, Li-Rong. Spectral conversion using deep neural networks trained with multi-source speakers. *Proc. IEEE ICASSP*, pages 4849–4853, 2015. DOI: `10.1109/ICASSP.2015.7178892`.

[22] Magariños, Carmen, Lopez-Otero, Lopez-Otero, Paula, Docio-Fernandez, Laura, Rodriguez-Banga, Eduardo, Erro, Daniel, and Garcia-Mateo, Carmen. Reversible speaker de-identification using pre-trained transformation functions. *Computer Speech & Language*, 46:36–52, 2017. DOI: `10.1016/j.csl.2017.05.001`.

[23] Mohammadi, Seyed Hamidreza and Kain, Alexander. Voice conversion using deep neural networks with speaker-independent pre-training. *Proc. IEEE SLT Workshop*, pages 19–23, 2014. DOI: `10.1109/SLT.2014.7078543`.

[24] Mohammadi, Seyed Hamidreza and Kain, Alexander. An overview of voice conversion systems. *Speech Communication*, 88:65–82, 2017. DOI: `10.1016/j.specom.2017.01.008`.

[25] Morise, Masanori. CheapTrick, a spectral envelope estimator for high-quality speech synthesis. *Speech Communication*, 67:1–7, 2015. DOI: `10.1016/j.specom.2014.09.003`.

[26] Morise, Masanori. D4C, a band-aperiodicity estimator for high-quality speech synthesis. *Speech Communication*, 84:57–65, 2016. DOI: `10.1016/j.specom.2016.09.001`.

[27] Morise, Masanori, Kawahara, Hideki, and Nishiura, Takanobu. Rapid F0 estimation for high-SNR speech based on fundamental component extraction. *IEICE TRANSACTIONS on Information and Systems*, 93:109–117, 2010.

[28] Morise, Masanori, Yokomori, Fumiya, and Ozawa, Kenji. WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Trans. Info. Sys.*, 99(7):1877–1884, 2016. DOI: `10.1587/transinf.2015edp7457`.

[29] Nakashika, Toru, Takiguchi, Tetsuya, and Ariki, Yasuo. Voice conversion based on speaker-dependent restricted Boltzmann machines. *IEICE Transactions on Information and Systems*, E97.D(6):1403–1410, 2014. DOI: `10.1587/transinf.E97.D.1403`.

[30] Nguyen, Hy Quy, Lee, Siu Wa, Tian, Xiaohai, Dong, Minghui, and Chng, Eng Siong. High quality voice conversion using prosodic and high-resolution spectral features. *Multimedia Tools and Applications*, 75(9):5265–5285, 2016. DOI: `10.1007/s11042-015-3039-x`.

[31] Qian, Jianwei, Du, Haohua, Hou, Jiahui, Chen, Linlin, Jung, Taeho, Li, Xiang-Yang, Wang, Yu, and Deng, Yanbo. VoiceMask: Anonymize and sanitize voice input on mobile devices. *arXiv:1711.11460*, 2017.

[32] Sekii, Yusuke, Orihara, Ryohei, Kojima, Keisuke, Sei, Yuichi, Tahara, Yasuyuki, and Ohsuga, Akihiko. Fast many-to-one voice conversion using autoencoders. *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*, pages 164–174, 2017. DOI: `10.5220/0006193301640174`.

[33] Tokuda, Keiichi, Kobayashi, Takao, Masuko, Takashi, and Imai, Satoshi. Mel-generalized cepstral analysis — a unified approach to speech spectral estimation. In *Third International Conference on Spoken Language Processing*, 1994.

[34] Wu, Zhizheng, Watts, Oliver, and King, Simon. Merlin: An open source neural network speech synthesis system. In *9th ISCA Speech Synthesis Workshop*. ISCA, 2016. DOI: `10.21437/ssw.2016-33`.

[35] Zue, Victor, Seneff, Stephanie, and Glass, James. Speech database development at MIT: TIMIT and beyond. *Speech Communication*, 9:351–356, 1990. DOI: `10.1016/0167-6393(90)90010-7`.