INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Voice Conversion Using Speech-to-Speech Neuro-Style Transfer

Ehab A. AlBadawy, Siwei Lyu

University at Albany, SUNY, USA
{ealbadawy,slyu}@albany.edu

Abstract

An impressionist is the one who tries to mimic other people's voices and their style of speech. Humans have mastered such a task throughout the years. In this work, we introduce a deep learning-based approach to do voice conversion with speech style transfer across different speakers. In our work, we use a combination of Variational Auto-Encoder (VAE) and Generative Adversarial Network (GAN) as the main components of our proposed model followed by a WaveNet-based vocoder. We use three objective metrics to evaluate our model using the ASVspoof 2019 for measuring the difficulty of differentiating between human and synthesized samples, content verification for transcription accuracy, and speaker encoding for identity verification. Our results show the efficacy of our proposed model in producing a high quality synthesized speech on Flickr8k audio corpus.

Index Terms: Speech Synthesis and Spoken Language Generation, voice conversion, Speech-to-Speech model

1. Introduction

Recently, deep neural networks have been widely used for speech synthesis using different techniques such as text-to-speech (TTS) [1–5] and speech-to-speech based approaches [6–8]. Despite the significant improvement in the synthesized audio quality introduced by [1], TTS based approaches tend to miss the emotional characteristics in the speech sample for a given speaker. We argue that using a similar approach as in speech-to-speech systems can overcome such a problem and improve the synthesized speech quality.

Speech style transfer is the process of synthesising speech sample from one source speaker to a different target speaker while keeping the linguistic and speech style the same. In this work, we introduce a speech-to-speech neural network that is able to transfer the speech style across different speakers. Our approach consists of two primary steps. Firstly, given a mel-spectrogram speech utterance input, we train VAE-GAN model to reconstruct the input sample using L1-loss for the target speaker and GAN loss for other different speakers. To further refine the model performance, we introduce the latent space loss on the VAE encoder features embedding as well as cycle consistency loss [9]. The training is performed end-to-end in an unsupervised manner without any alignments between the input samples. Secondly, we train WaveNet-based vocoder on the VAE-GAN mel-spectrogram outputs to generate the synthesized speech in the time domain. Our method is inspired by a recent image-to-image style transfer model [10] applied to mel-spectrograms. Our method uses a single encoder for all input speakers to make it more feasible to generalize to multiple target speakers. Moreover, we introduce the latent loss to further constrain the encoded features in eliminating input speaker identity. This allows us to generate natural human-like synthesized speech with a unique style for each speaker.

We use three different objective metrics to evaluate our model, namely the ASVSpoof [11] for measuring the difficulty of distinguishing between the synthesized and real human samples, content verification for evaluating integrity in transferring the linguistic information between the source and the target speakers, and speaker encoding [7] for validating the speaker identity in the synthesized speech samples. Experimental evaluations on the Flickr8k audio corpus [12] show the effectiveness of our method in generating human-like speech samples while capturing the linguistics and speech style of the input speaker.

2. Related Work

2.1. Speech Synthesis

We focus our discussion on neural network based speech synthesis methods that are relevant to our current work. Hasegawa-Johnson et al. [13] proposed a sequence-to-sequence model to generate spoken description from the input image in the image2speech problem. They used both Flickr8k [12] and SPEECH-COCO [14] corpora to show the intelligibility of their model in generating relevant words and sequence them in a meaningful sentence. Jia et al. [7] proposed a neural network model to tackle TTS problem to generate synthesized speech for a given speaker. Their model contains three independent components; Speaker encoder, Synthesizer, and a neural vocoder based on Tacotron 2 [2]. They showed that their model is capable of synthesize speech for unseen speakers based on the features embedding coming from the speaker encoder model. Biadsy et al [6] introduced a speech-to-speech model named Parrotron where it is trained end-to-end. They used their model for speech normalization where they map input spectrogram of different speakers to an output spectrogram of a single target speaker. Their model is trained to transfer the linguistic content to the target speaker while ignoring non-linguistic content. Our proposed model defers from Biadsy et al [6] work by preserving both the linguistic content as well as the speech style of the input speaker and transfer them to the target speaker.

2.2. Neural Style Transfer

A notable amount of work has been introduced to tackle the style transfer problem mostly in the image domain. Liu et al. [10] proposed the UNIT model for image-to-image translation from one domain to another in an unsupervised manner, which is the major inspiration of the current work. This model consists of one encoder, generator, and a discriminator for each input/target domain [15]. In the audio domain, Mor et al. [16] presented a multi-domain WaveNet autoencoder to translate an input music record to different musical instruments and styles. Their model consists of one encoder and different target decoders each for target instruments. They used a domain confusion network to constrain the encoder not to memorize the input signal and produce a semantic encoding instead. In our work, we tackle the style transfer problem in the frequency do-

http://dx.doi.org/10.21437/Interspeech.2020-3056

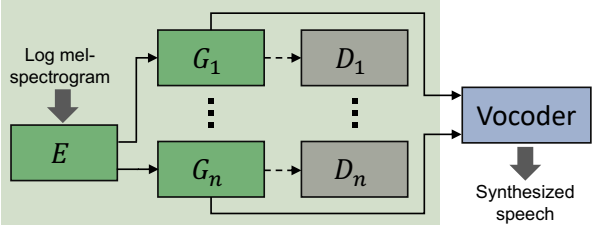

Figure 1: *The overall pipeline of our proposed model. The VAE-GAN network is trained independently from the vocoder model.*

main using mel-spectrogram input/output instead of the time domain with waveforms. This helps us to have a more stable and faster training procedure while having simpler model architecture compared to [16] where they used a WaveNet model for both their encoder and decoders subnetworks.

3. Methods

In this section, we describe in detail the model architecture and the training procedure of our method. Fig 1 shows the overall pipeline of our method. The input to the system is a speech signal of one speaker, which is converted to the target speaker's voice while keeping the content and style of the original speaker. The input speech signal is first converted to the mel-spectrogram representation (details in Sec 4.1). We then employ a neural style transfer model similar to [10], treating the input mel-spectrogram as a gray-scale image, to create an output mel-spectrogram with the style of the target speaker. The generated mel-spectrogram is then fed to the vocoder to reconstruct the speech signal in the time domain. We use the WaveNet vocoder [17] based on the open-source implementation [18].

3.1. Framework

The core components of the neural style transfer model are a pair of convolutional neural networks, corresponding to the encoder and generator (decoder), Fig.1. The encoder preserves the linguistic information in the input speech while removes identity-related information, the generator combines the style and the content of the input speech signal to create the mel-spectrogram of a new speech signal. To ensure the encoder capture identity-independent attributes such as speech volume and tempo, there is one single encoder regardless of the identities of the input speakers. We assume that using a shared encoder for the different speakers will imply a shared-latent space z that contains each sample content while removing the original speaker identity.

The architecture of the encoder E has three main parts. The first part contains the initial convolutional layer with 7×7 kernel size and no stride. The second part has two down-sampling convolutional layers with 4×4 kernel size and stride of 2. Each convolutional layer is followed by a batch normalization [19] and a LeakyReLU non-linear activation function. The third part consists of three residual blocks [20] as a final feature extractor.

Each of the different generators (G_1 to G_n where n is the number of target speakers) consists of the same parts as the encoder E but in reverse order with two exceptions. First, instead of convolutional layers, we use transposed convolutional layer for up-sampling. Second, because of the shared latent space assumption coming from the single encoder E, all of the generators share the first residual block as a pre-processing step of the latent code for each of the generators [10].

To transfer the style from one speaker to another, we exchange the latent space for both decoders as shown in Fig 2. More specifically, for speaker S_i there is a latent code $z_i = E(S_i)$ where i is the i-th speaker. To get the same sentence spoken by different speakers, we feed the latent code z_i to the corresponding speaker Generator. Where $G_i(z_i)$ will reconstruct the same input $\tilde{S}_i$, and $G_j(z_i)$ will generate $S_{i\to j}$ that has the same content i being said by speaker j (i.e. style transfer between speakers i and j).

3.2. Training

The encoder and generator are trained in tandem using uncorresponded sets of speech signals of multiple subjects in an unsupervised manner. To facilitate the subsequent description, we will use the following notations:

- i and j are speaker indices where $i, j \in [1, n]$ and $i \neq j$
- S_i is a data point for speaker i drawn from distribution P_{S_i}
- $S_{i\to j}$ represents the translated speech from speaker i to speaker j where $S_{i\to j} = G_j(E(S_i))$
- $q(z_i|S_i)$ is probabilistic encoder produces distribution z_i given speaker sample S_i
- $p_{G_i}(S_i|z_i)$ is probabilistic generator for speaker i that produces distribution S_i given latent code z_i

The overall training loss of the neural style-transfer model for mel-spectrogram is defined as follows:

$$L = \lambda_1 L_{\text{VAE}} + \lambda_2 L_{\text{GAN}} + \lambda_1 L_{\text{CC}} + \lambda_3 L_{\text{latent}} \quad (1)$$

The **VAE loss (L_{VAE})** is defined as follows

$$L_{\text{VAE}} = \lambda_4 \sum_i D_{\text{KL}}(q(z_i|S_i)||p(z)) - \sum_i \mathbb{E}_{z_i\sim q(z_i|S_i)}[\log p_{G_i}(S_i|z_i)] \quad (2)$$

where the first term is the KL divergence (KLD) of the approximated posterior and the prior of the latent space and the second term is calculated through the Monte Carlo method, which can be understood in terms of the reconstruction of the input from the posterior distribution and the likelihood. For the KL-divergence we use prior distribution $p(z)$ as a zero mean Gaussian $\mathcal{N}(z|0, I)$ [15].

For each speaker there is GAN subnetwork that use the VAE subnetwork for the generation step followed by D_k as a discriminator where k is the index of a given speaker and $k \in [1, n]$. For example, in speaker 1 we have GAN_1 that consists of G_1 and D_1. Positive samples for GAN_1 are sampled from S_1, while negative samples are G_1 outputs for input speaker i where $i \in [2, n]$. Then, the **GAN Loss (L_{GAN})** aims to penalize the VAE network for the translated samples between speakers i and j ($i \neq j$)

$$L_{\text{GAN}} = \sum_i \mathbb{E}_{S_i\sim P_{S_i}}[\log D_i(S_i)] + \sum_{i,j} \mathbb{E}_{S_{j\to i}\sim p_{G_i}(S_{j\to i}|z_j)}[\log(1 - D_i(S_{j\to i})] \quad (3)$$

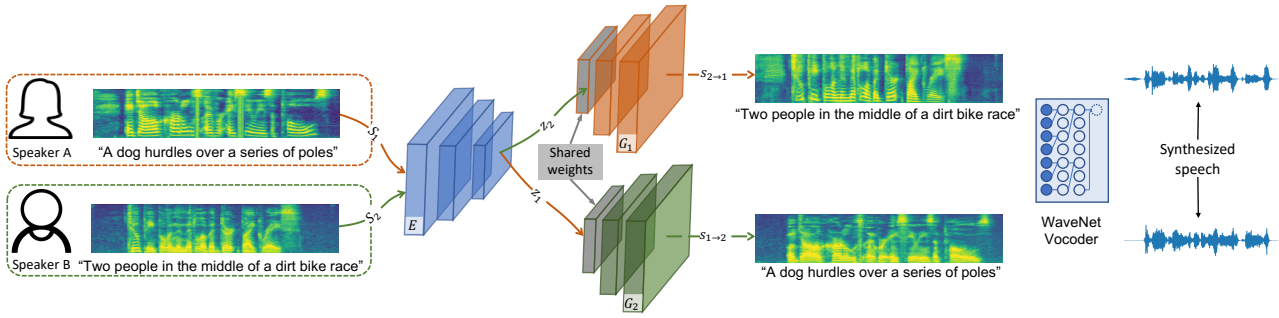

Figure 2: *The overall procedure of voice conversion with our method for style transfer between two different speakers during evaluation. The Encoder E generates features embedding for a given input speaker which then based on one of the Generators (G_1 or G_2) to produce a mel-spectrogram output for the target speaker. We use WaveNet-based vocoder to generate the speech in the time domain.*

The **Cycle Consistency (CC) Loss (L_{CC})** helps to enforce the speaker independent shared-latent space assumption by having a cycle-reconstruction stream [9]

$$L_{\text{CC}} = \lambda_4 \sum_{i,j} D_{\text{KL}}(q(z_j|S_{i\to j})||p(z)) - \sum_{i,j} \mathbb{E}_{z_j\sim q(z_j|S_{i\to j})}[\log p_{G_i}(S_i|z_j)] \quad (4)$$

Similar to VAE loss (L_{VAE}), we use KL divergence and negative log-likelihood for L_{CC} computation. The KL divergence penalizes the network on the latent codes of both the original and translated samples from the prior distribution. While the negative likelihood term ensures the reconstruction of the original S_i from the translated one $S_{i\to j}$.

The **Latent Loss (L_{latent})** is the L1-distance between the codes' centroids of all speakers i and j

$$L_{\text{latent}} = |C_i - C_j| \quad (5)$$

where C_i and C_j are the centroids of speakers i and j distributions respectively defined as follows:

$$C_i = \frac{1}{|P_{S_i}|} \sum_{S_i \in P_{S_i}} E(S_i). \quad (6)$$

We implemented our neural speech style transfer model using PyTorch framework, and train it on a Titan Xp GPU for approximately 8 hours. We use the Adam optimizer with $1e-4$ learning rate with a batch size of 4 samples for each speaker. The training algorithm is run for 100 epochs. For the regularization parameters in the objective functions, we use $\lambda_1 = 100$, $\lambda_2 = 10$, $\lambda_3 = 10$, and $\lambda_4 = 1e-3$. We choose these values of the regularization parameters to give more weight to the reconstruction loss in L_{VAE} compared to other loss terms.
In addition, we also train the WaveNet vocoder [17] separately using the mel-spectrograms from both generators (G_1 and G_2) where the target ground truth is the original waveform for each sample.

4. Experiments

4.1. Dataset and Feature Extraction

In our work, we use the Flickr8k Audio Caption Corpus [12], which contains $40,000$ spoken captions generated from $8,000$ photographs from Flickr.com. All utterances have a sampling rate of 16 kHz and are 1.9 seconds in length. To increase the diversity in the training/testing sets, we pick two speakers with opposite genders and have the most number of utterances. The total number of utterances for both training and testing are $4,668$ (60% male and 40% female). We strictly divide the dataset to training and testing sets with an approximate ratio of 70% and 30%, respectively. We use the training set to train both the VAE-GAN network and the WaveNet vocoder [17], while the testing set is used to evaluate the model performance.

We compute the log mel-spectrogram with 0.05 seconds window length and quarter-window overlap; this produces n windows and 128 frequency bands where n depends on each utterance length. For each training step, we randomly crop 128 consecutive frames from the log mel-spectrogram of each utterance to generate a 128×128 input sample.

We train the mel-spectrogram based neural style transfer model using the procedure in Sec 3.2. We then perform voice conversion with a style transfer experiment between the two selected speakers as described in Section 3.1.

4.2. Evaluation

To date, there has not been universally agreed objective metrics for the quality of the synthesized utterance. To this end, we use three objective methods to evaluate the naturalness of synthesized voices using our model. For fairness, we train each the evaluation methods on its original dataset it proposed with to eliminate any possible bias in our results. For testing, we report the final results on the held out test set using both the original and synthesized samples. For the synthesized samples, we only pick the ones with style transfer between two different speakers ($S_{i\to j}$). In the following, we provide more details about each evaluation method and the train set for each of them.

ASVspoof 2019 Baseline. The AVSspoof 2019 Challenge [11] provides a Gaussian Mixture Model (GMM)-based model as their main classifier with linear frequency cepstral coefficients [21] (LFCC) features. For training, we use the original ASVspoof 2019 dataset to train the GMM model with. We use the Equal Error Rate (EER) to evaluate the classifier performance in the test split. While the ASVspoof baseline does not measure the quality of the synthesized speech, we use it as a quantitative evaluation to measure the difficulty of distinguishing between real human voices and synthesized ones.

Content Verification Metric. To ensure that the synthesized speeches contain the same linguistic content as the original speech, we use the *word error rate* (WER) between the original transcript and the predicted one from the syn-

thesized speeches to measure the intelligibility. We use the `SpeechRecognition` [22] open-source library to get the transcript of each sample.

Speaker Encoding Metric [7]. We use the `Speaker Encoder` model from [7] to verify if the speaker identity is preserved in the synthesized speech. This model uses a `long-short term memory` (LSTM) based RNN model for speaker encoding. The input to the model is the mel-spectrogram frames translated to a 265-dim vector for each speech sample. In training this model, a generalized end-to-end speaker verification loss [23] is minimized, where the samples with the same speaker preserve high cosine similarity, while the samples from different speakers are far apart in the embedding space. We use their pre-trained model without any fine-tuning and test whether the original and synthesized samples from the same speaker are in the same cluster. To classify each sample for one of the two speakers, we use the centroids of original samples for each of the two speakers using Eq 6, and then get the probability for each class using the following equation:

$$p(y = k|x) = \frac{\exp -d(f_\theta(x), c_k)}{\sum_{i=1}^{2} \exp -d(f_\theta(x), c_i)} \tag{7}$$

where k is the class number and $k \in 1, 2$, x is the input sample, $f_\theta x$ is the speaker encoder model, c_k is the centroid of speaker k, and d is the euclidean distance function. We use EER to evaluate the model performance.

4.3. Results

Table 1: *Equal Error Rate (EER) [%] using ASVSpoof and Speaker encoding metrics, and Word Error Rate (WER) [%] using Content Verification metric*

Metric	Data	EER/WER
ASVspoof 2019 [11]	Evaluation set in [11]	9.57
	Flickr8k [12] test split	38.89
Content Verification	Original samples	2.01
	Synthesised samples	10.36
Speaker Encoding [7]	Flickr8k [12] test split	0.001

Table 1 summarizes the performance of our model with regards to the three evaluation methods in Sec 4.2. To test the difficulty of distinguishing between real and fake samples, we use the ASVspoof baseline [11]. From the held-out test set, we construct a balanced number of real and synthesized samples. The real samples come from the original speech of each speaker, where the synthesized samples are the ones with style transfer between two different speakers. The ASVspoof 2019 baseline method has a 38.89% EER on the Flickr8k test set while its performance on the original ASVspoof 2019 dataset is 9.57 EER [11]. This indicates it is more difficult to differentiate between real and synthesized samples from our method than those from the original ASVspoof baseline dataset.

For the second evaluation method, we use WER to see if the model preserves the original linguistic content in the synthesized samples. We first compute the WER between the original and the predicted transcripts from the speech samples using the open-source `SpeechRecognition` Library [22]. As shown in Table 1, we get 2.01% WER on the original samples. We use this value as an upper bound for the content verification

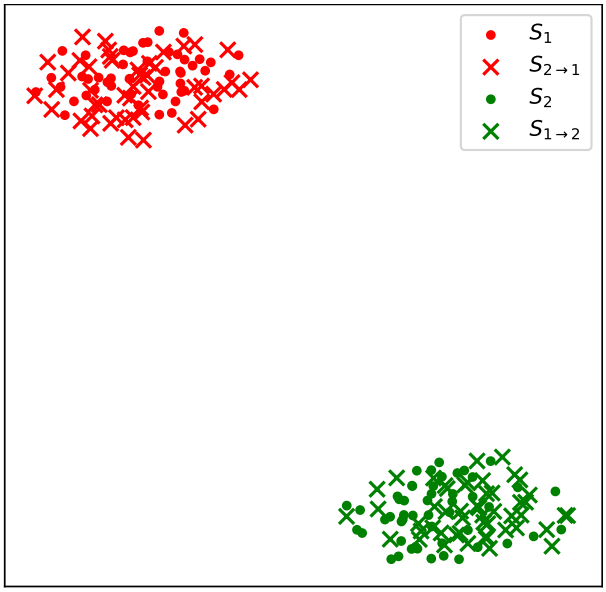

Figure 3: *tSNE [24] visualization of the features embedding for each speaker on the original (S_1, S_2) and synthesized samples ($S_{2\to1}, S_{1\to2}$) using speaker encoding evaluation method [7].*

method's performance. Computing the WER on the synthesized samples we achieve relatively close WER to the upper bound with 10.36% WER. This indicates the intelligence of the model to preserve the original content in the synthesized samples.

As shown in Fig 3, both the original and synthesized samples of the same speaker occupy the same cluster while there is a distinct separation between the two speakers' clusters. We compute EER on the predicted probabilities using Eq 7 where we achieve 0.001%EER (Table 1). These results demonstrate the efficacy of our proposed model to preserve the speaker's identity on the synthesized samples.

5. Conclusion and Future Work

In this work, we present a new voice conversion method based on a neural style transfer model of the mel-spectrograms. Our method takes advantage of the recent developments in neural network models for image style transfer. Experimental results show that our method can faithfully transfer styles across different speakers while preserving the content of the original speech. In future work, we will further explore the possible modification of our proposed model to generalize to broad samples with noise in the background as well as cross-linguistic speech style transfer.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No (IIS-1816227). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work is also supported by a Google Faculty Research Award.

7. References

[1] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] 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.

[3] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *arXiv preprint arXiv:1710.07654*, 2017.

[4] Z. Kons, S. Shechtman, A. Sorin, C. Rabinovitz, and R. Hoory, "High quality, lightweight and adaptable tts using lpcnet," *Proc. Interspeech 2019*, pp. 176–180, 2019.

[5] Y. Wang, R. Skerry-Ryan, Y. Xiao, D. Stanton, J. Shor, E. Battenberg, R. Clark, and R. A. Saurous, "Uncovering latent style factors for expressive speech synthesis," *ML4Audio Workshop, NIPS*, 2017b.

[6] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia, "Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," *Proc. Interspeech 2019*, pp. 4115–4119, 2019.

[7] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in neural information processing systems*, 2018, pp. 4480–4490.

[8] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, "Direct speech-to-speech translation with a sequence-to-sequence model," *arXiv preprint arXiv:1904.06037*, 2019.

[9] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[10] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in neural information processing systems*, 2017, pp. 700–708.

[11] 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," *arXiv preprint arXiv:1904.05441*, 2019.

[12] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using amazon's mechanical turk," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*. Association for Computational Linguistics, 2010, pp. 139–147.

[13] M. Hasegawa-Johnson, A. Black, L. Ondel, O. Scharenborg, and F. Ciannella, "Image2speech: Automatically generating audio descriptions of images," *Proceedings of ICNLSSP, Casablanca, Morocco*, 2017.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[16] N. Mor, L. Wolf, A. Polyak, and Y. Taigman, "A universal music translation network," in *International Conference on Learning Representations*, 2018.

[17] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, pp. 125–125.

[18] R. Yamamoto, M. Andrews, M. Petrochuk, W. Hy, cbrom, O. Vishnepolski, M. Cooper, K. Chen, and A. Pielikis, "r9y9/wavenet_vocoder: v0.1.1 release," Oct. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1472609

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[20] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, 2016, pp. 469–477.

[21] M. Sahidullah, T. Kinnunen, and C. Hanilçi, "A comparison of features for synthetic speech detection," 2015.

[22] A. Zhang, "Speech recognition (version 3.8) [software]," https://github.com/Uberi/speech˙recognitionreadme, 2017.

[23] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[24] L. Van Der Maaten, "Barnes-hut-sne," *arXiv preprint arXiv:1301.3342*, 2013.