Voice impersonation is not the same as voice transformation, although the latter is an essential element of it. In voice impersonation, the resultant voice must convincingly convey the impression of having been naturally produced by the target speaker, mimicking not only the pitch and other perceivable signal qualities, but also the style of the target speaker. In this paper, we propose a novel neural-network based speech quality- and style-mimicry framework for the synthesis of impersonated voices. The framework is built upon a fast and accurate generative adversarial network model. Given spectrographic representations of source and target speakers’ voices, the model learns to mimic the target speaker’s voice quality and style, regardless of the linguistic content of either’s voice, generating a synthetic spectrogram from which the time-domain signal is reconstructed using the Griffin-Lim method. In effect, this model reframes the well-known problem of style-transfer for images as the problem of style-transfer for speech signals, while intrinsically addressing the problem of durational variability of speech sounds. Experiments demonstrate that the model can generate extremely convincing samples of impersonated speech. It is even able to impersonate voices across different genders effectively. Results are qualitatively evaluated using standard procedures for evaluating synthesized voices.

Index Terms— Voice impersonation, generative adversarial network, style transformation, style transfer
The original GAN model [7] comprises a generator \( G(z) \) and discriminator \( D(x) \). The generator \( G \) takes as input a random variable \( z \) drawn from some standard probability distribution function \( P_z \), e.g. a standard Normal distribution, and produces an output vector \( G(z) \).

The discriminator \( D() \) attempts to discriminate between samples \( x \sim P_x \) that are drawn from \( P_x \), the true (but unknown) distribution we aim to model, and samples produced by the Generator \( G \). Let \( T \) represent the event that a vector \( x \) was drawn from \( P_x \). The discriminator attempts to compute the \textit{a posteriori} probability of \( T \), i.e. \( D(x) = P(T|x) \).

To train the GAN, we attempt to learn \( G \) such that \( D(G(z)) \), the score output by the discriminator in response to productions by \( G \) is maximized (i.e. \( G \) “fools” the discriminator). At the same time we attempt to learn \( D \) such that \( D(G(z)) \) is minimized, while also maximizing \( D(x) \) for any \( x \sim P_x \). All of these objectives can be concurrently achieved through the following optimization:

\[
\min_G \max_D \mathbb{E}_{x \sim P_x} \left[ \log D(x) \right] + \mathbb{E}_{z \sim P_z} \left[ \log(1 - D(G(z))) \right]
\]

The GAN training framework is illustrated in Figure 1.

![Fig. 1. The original GAN model](image)

![Fig. 2. Style transfer by GAN](image)

### 2.2. GANs for style transfer

The basic GAN has been extended in a number of ways in the literature [9, 10, 11, 12, 13], particularly in the context of style transfer among images, e.g. as in Figure 2. The common underlying denominator in all of these models is that an input data instance (usually an image) \( x_A \) drawn from a distribution \( P_A \) is transformed to an instance \( x_{AB} \) by a generator (more aptly called a “transformer”), \( G_{AB} \). The aim of the transformer is to convert \( x_A \) into the style of the variable \( x_B \) which natively occurs with the distribution \( P_B \).

The discriminator \( D_B \) attempts to distinguish between genuine draws of \( x_B \) from \( P_B \) and instances \( x_{AB} \) obtained by transforming draws of \( x_A \) from \( P_A \). The actual optimization is achieved as follows. We define

\[
\begin{align*}
L_G &= E_{z \sim P_z} \left[ \log(1 - D_B(G(z))) \right] \\
L_D &= -E_{x_A \sim P_A} \left[ \log D_B(x_A) \right] - E_{z \sim P_z} \left[ \log(1 - D_B(G(z))) \right]
\end{align*}
\]

To train the GAN, its two components are alternately updated by minimizing the two losses in Equation 1. The generator \( G \) is updated by minimizing the “generator loss” \( L_G \), while the discriminator is updated to minimize the “discriminator loss” \( L_D \).

Our work is however more directly based on the “DiscoGAN” model [9], shown in Figure 3. The DiscoGAN is a symmetric model which attempts to transform two categories of data, \( A \) and \( B \), into each other. The DiscoGAN includes two generators (more aptly called “transformers”) \( G_{AB} \) and \( G_{BA} \). \( G_{AB} \) attempts to transform any draw \( x_A \) from the distribution \( P_A \) of \( A \) into \( x_{AB} = G_{AB}(x_A) \), such that \( x_{AB} \) is indistinguishable from draws \( x_B \) from the distribution \( P_B \) of \( B \). \( G_{BA} \) does the reverse.
since our objective is to modify specific aspects of the speech, e.g. style, we must add extra components to our model to achieve this. We call our model, which incorporates all these modifications, the VoiceGAN.

3.0.1. Retaining Linguistic Information

Linguistic information is encoded largely in the details of the spectral envelope. To ensure that this is retained, we modify our reconstruction loss as:

\[
L_{\text{CONST}} = \alpha d(x_{\text{ABA}}, x_A) + \beta d(x_{AB}, x_A)
\] (4)

Here, the term \(d(x_{AB}, x_A)\) attempts to retain the structure of \(x_A\) even after it has been converted to \(x_{AB}\). Careful choice of \(\alpha\) and \(\beta\) ensures both, accurate reconversion and retention of linguistic information, after conversion to \(x_{AB}\).

3.0.2. Variable-length Input Generator and Discriminator

To account for the fact that unlike images, speech signals are of variable length that cannot be scaled up or down, we must make modifications to the generators and discriminators. The modified structures are shown in Figure 4. Figure 4 (a) shows the structure of the original generator in DiscoGAN. Based on its fully convolutional structure, it can handle variable length inputs. Figure 4 (b), we shows the architectural details for our proposed discriminator in VoiceGAN. In this, an adaptive pooling layer is added after the CNN layers, and before the fully connected layer. It includes channel-wise pooling in which each channel’s feature map is pooled into a single element. This converts any variable-sized feature map into a vector of a fixed number of dimensions, with as many components as the number of channels.

3.0.3. Style Embedding Model (\(D_S\))

In addition to the discriminator that distinguishes between the generated and real data, we add a second type of discriminator to our model to further extract the target style information from input data and to make sure that the generated data still has this style information embedded in it. To achieve this, we include a discriminator \(D_S\) that is similar in architecture to that in Figure 5.

The discriminator \(D_S\) determines if the original and transformed signals match the desired style. To do so we introduce the following style loss:

\[
L_{\text{STYLE-A}}(x_A) = d(D_S(x_A), label_A) + d(D_S(x_{AB}), label_B) + d(D_S(x_{ABA}), label_A)
\] (5)
4. EXPERIMENTS AND RESULTS

We use the TIDIGITS [14] dataset. This dataset comprises a total of 326 speakers: 111 men, 114 women, 50 boys and 51 girls. Each speaker reads 77 digit sentences. The sampling rate of the audio is 16000 Hz. We chose to use this database due to its relatively simple linguistic content. For the purpose of demonstration, we choose an unquantifiable, but identifiable characteristic: gender. Our goal then is to show that these data can be used to learn to convert the gender of a speaker’s voice. In the discussion below, therefore, “style” refers to gender. We note that any other characteristic may have been similarly chosen.

4.1. Model implementation

The model architecture is that of the VoiceGAN described above. The generator network in the model comprises a 6-layer CNN encoder and a 6-layer transposed CNN decoder. The discriminator network comprises a 7-layer CNN with adaptive pooling. We employ batch normalization [15] and leaky ReLU activations [16] in both the networks. The number of filters in each layer is an increasing power of 2 (32, 64, 128). When training the networks, a smoothness constraint, comprising the cumulative first order difference between adjacent columns in the spectrogram, is added to the loss to enhance the temporal continuity of the generated spectrogram. Results are available at [17].

4.2. Quality evaluation of generated results

4.2.1. Style Classification Test

We use an independently-trained CNN-based classifier to predict the style of our generated data. The classifier was trained on 800 utterances from speakers of both genders. The results show that 100% of the generated data are classified as the target speaker’s style, which indicates that our VoiceGAN network achieves good style transfer performance.

4.2.2. Speech Signal to Noise Ratio (SNR) Test

To evaluate the quality of our generated speech signal [18], we also conduct a signal-to-noise (SNR) ratio test using the standard NIST STNR method and the WADA SNR method [19]. The results are shown in Table 2. For each data class, we randomly select 40 samples from our test dataset (20 for each speaker) and compute the mean and variance of the generated results. The WADA test results are all around 100 dBJ since our generated noise is not well-modeled by Gaussian noise. The STNR test results show that our generated data is of good quality. For evaluation, the time-domain signal is reconstructed from the generated spectrogram using the Griffin-Lim method, which is based on an iterative procedure that minimizes the mean square error between the modified magnitude spectrogram and the actual signals spectrogram. Details of this method are explained in [20]. We find that the Griffin-Lim method does not reduce the voice quality to any significant degree.

5. CONCLUSIONS

The VoiceGAN model is observably able to transfer style from one speaker to another. As proposed however, this model remains vanilla and many extensions are possible. The method is easily extended to other stylistic features that may be identified. In principle, while longer-term prosodic-level style features may also be transferred, simple binary discriminators may no longer be useful for such characteristics. More continuous-valued discrimination may be required. We have not verified if multiple style aspects may be modified if we so choose; however doing so in a measurable and controlled manner is a challenge that remains to be addressed. Future versions of the VoiceGAN model will continue to incorporate the most relevant innovations in the area of adversarial modeling.

Table 1. NIST STNR TEST

<table>
<thead>
<tr>
<th>Data (use GL-method)</th>
<th>A (dB)</th>
<th>B (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original signal</td>
<td>55.60±4.97</td>
<td>52.91±3.58</td>
</tr>
<tr>
<td>X_A and X_B</td>
<td>54.97±6.28</td>
<td>52.15±3.70</td>
</tr>
<tr>
<td>X_AB and X_BA</td>
<td>49.64±1.80</td>
<td>49.92±4.36</td>
</tr>
<tr>
<td>X_ABA and X_BAB</td>
<td>53.58±2.69</td>
<td>50.05±2.12</td>
</tr>
</tbody>
</table>
6. REFERENCES


