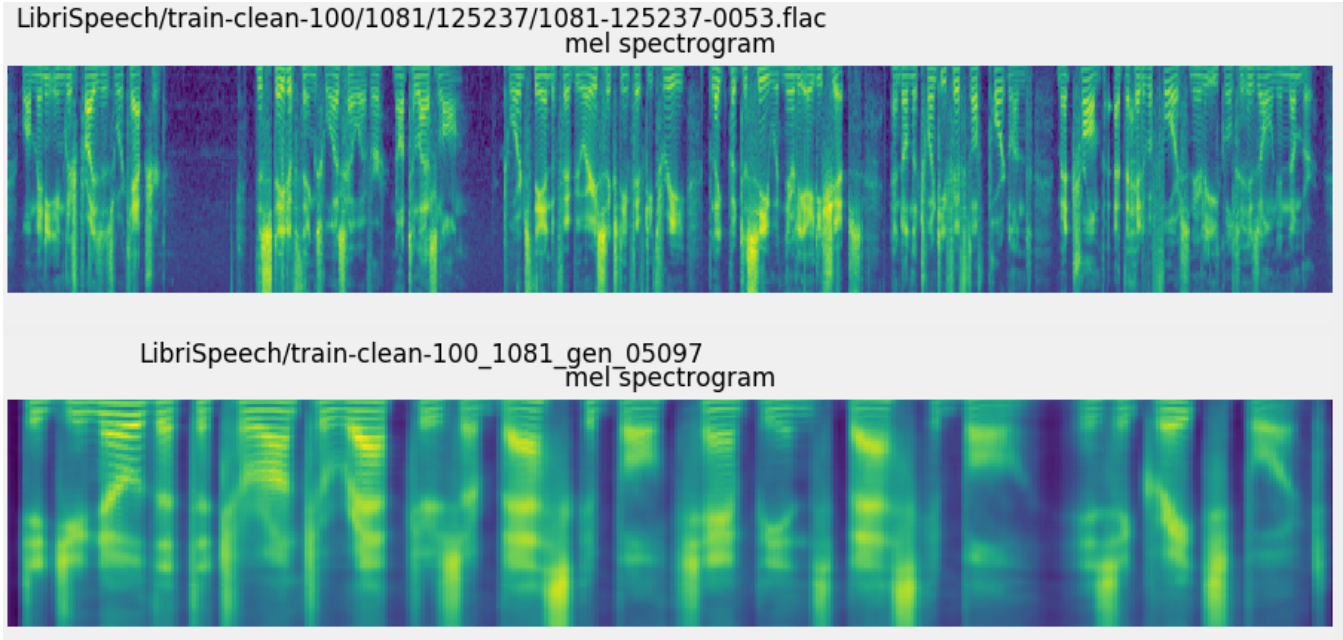
You can now speak using someone else's voice with Deep Learning

George Seif
Jul 2, 2019 · 4 min read ★



Text-to-Speech (TTS) Synthesis refers to the artificial transformation of text to audio. A human performs this task simply by reading. The goal of a good TTS system is to have a computer do it automatically.

One very interesting choice that one makes when creating such a system is the selection of which *voice* to use for the generated audio. Should it be a man or a woman? A loud voice or a soft one?

This used to present a restriction when doing TTS with Deep Learning. You'd have to collect a dataset of text-speech pairs. The set of speakers who recorded that speech is

How Voice Cloning Works

It's clear that in order for a computer to be able to read out-loud with any voice, it needs to somehow understand 2 things: what it's reading and how it reads it.

Thus, Google researchers designed the voice cloning system to have 2 inputs: the text we want to be read and a sample of the voice which we want to read the text.

For example, if we wanted Batman to read the phrase "I love pizza", then we'd give the system two things: text that says "I love pizza" and a short sample of Batman's voice so it knows what Batman should sound like. The output should then be an audio of Batman's voice saying the words "I love pizza"!

From a technical view, the system is then broken down into 3 sequential components:

(1) Given a small audio sample of the voice we wish to use, encode the voice waveform into a fixed dimensional vector representation

(2) Given a piece of text, also encode it into a vector representation. Combine the two vectors of speech and text, and decode them into a Spectrogram

(3) Use a Vocoder to transform the spectrogram into an audio waveform that we can listen to.

Simplified version of the system. Original source

In the paper, the three components are trained independently.

Text-to-speech systems have gotten a lot of research attention in the Deep Learning

After being separately encoded, the speech and the text are **combined** in a **common embedding space**, and then **decoded together** to create the final output waveform.

Code to clone voices

Thanks to the beauty of the open source mindset in the AI community, there is a publicly available implementation of this voice cloning right here! Here's how you can use it.

First clone the repository.

```
git clone https://github.com/CorentinJ/Real-Time-Voice-Cloning.git
```

Install the required libraries. Be sure to use Python 3:

```
pip3 install -r requirements.txt
```

In the README file you'll also find links to download pre-trained models and datasets to try out some of the samples.

Finally, you can open the GUI by running the following command:

```
python demo_toolbox.py -d <datasets_root>
```

There's a picture of how mine looks like down below.

As you can see, I've set the text I want the computer to read on the right side as: "Did you know that the Toronto Raptors are Basketball champions? Basketball is a great sport."

You can click on the "Random" buttons under each section to randomise the voice input, then click "Load" in load the voice input into the system.

Dataset selects the dataset from which you will select voice samples, *Speaker* selects the person who is talking, and *Utterance* selects the phrase which is spoken by the input voice. To hear how the input voice sounds, simply click "Play".

Once you press the button "Synthesize and vocode" the algorithm will run. Once it's finished you'll here the input Speaker reading your text out-loud.

You can even record your own voice as an input but clicking on the "Record one" button, which is quite fun to play around with!

Further Reading

If you'd like to learn more about how the algorithm works, you can read Google's official NIPS paper. There are some further audio sample results over here. I'd highly cloning the repository and giving this awesome system a try!

. . .

Like to learn?

Follow me on twitter where I post all about the latest and greatest AI, Technology, and