Universidad de Zaragoza
Laboratorio de señal y comunicaciones

Aplicación práctica sobre HW con recursos limitados: Transformación de voz con procesado de señal no lineal

Javier Antorán (698802@unizar.es)
Alberto Mur (565825@unizar.es)

1. Propuesta

Nuestra propuesta consiste en un suplantador de voz. Dicha herramienta correrá lo más cerca posible a tiempo real. Este tipo de sistema ha sido implementado a gran escala en proyectos como WaveNet [2], sin embargo, no conocemos de ninguna implementación en entornos de recursos limitados como la propuesta de este documento. El hardware elegido será una Raspberry Pi 3 y el desarrollo se hará en Python 2.7. El objetivo del trabajo será la creación del software necesario para la captación de voz y la reproducción de la voz transformada. Un diagrama de bloques del sistema propuesto se puede ver en [Figure 1]

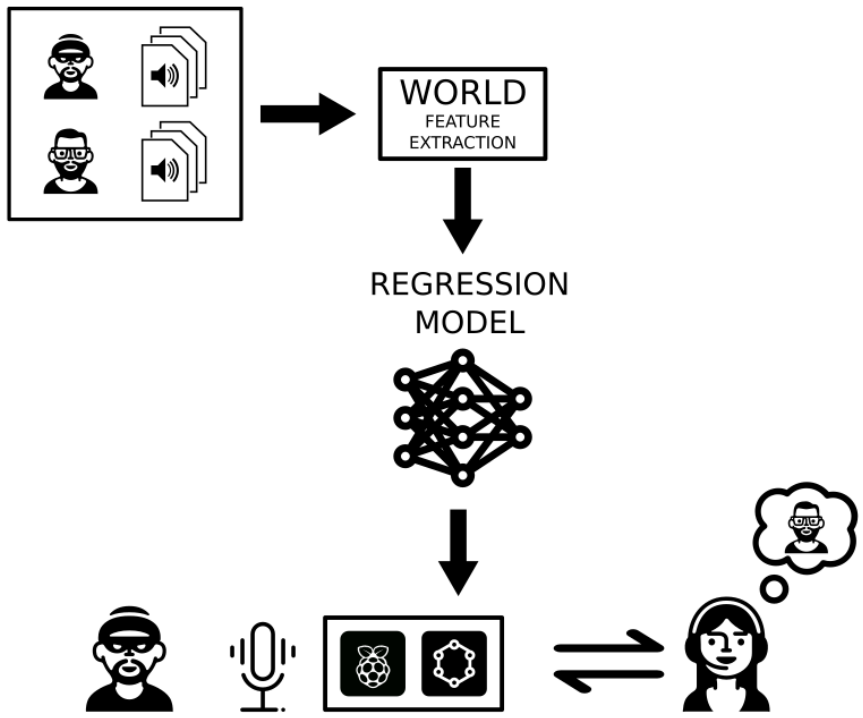

Figura 1: Esquema conceptual de la aplicación

2. Arquitectura

El sistema consistirá de una Raspberry Pi 3 la cual ejecutará nuestra aplicación. A su vez, esta consistirá de tres bloques principales: un codificador de voz, un modelo de regresión y un decodificador.

El vocoder se encargará de la extracción de características relevantes de la voz por ventanas. Su diseño se basará en un sistema de análisis, manipulación y síntesis de audio existente: WORLD[1]. El codec WORLD es adecuado para esta tarea ya que ha sido diseñado para sintetizar audio de alta fidelidad en contextos de tiempo real. Un ejemplo de su uso es la herramienta Merlin[3]. Además, está ampliamente documentado e integrado con Python a través de la librería pyWORLD. Dicha herramienta estima la frecuencia fundamental (F0), la aperiodicidad y la envolvente espectral de la señal incidente. También sintetiza la voz de forma artificial con dichos parámetros. El esquema de funcionamiento de world se puede ver en [Figure 2]

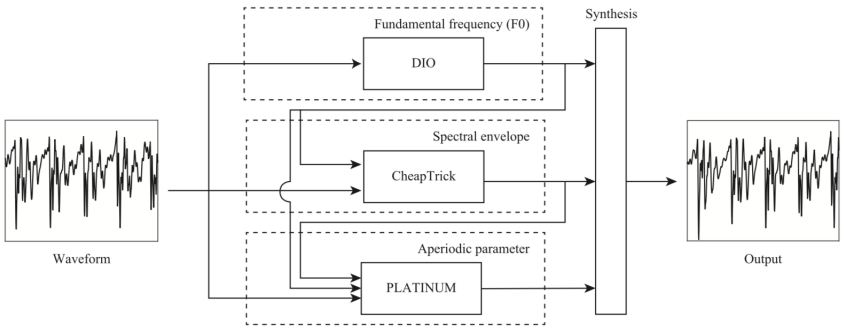

Figura 2: Esquema conceptual de world. Adaptado de [1]

Para modificar las características de la voz sintetizada efectuaremos una regresión sobre los parámetros dados por el encoder WORLD desde la voz de entrada a la voz objetivo. Utilizaremos el decoder para generar la voz modificada.

Para obtener el mejor resultado posible, compararemos las capacidades de varios modelos de regresión para la tarea dada. Los principales criterios que tendremos en cuenta son la calidad de audio obtenida y la complejidad computacional. Esta última resultará ser un factor condicionante dadas las limitadas capacidades de nuestra plataforma hardware. El modelo será entrenado con las características de la voz obtenidas por WORLD en un ordenador distinto a la Raspberry Pi.

Los modelos actualmente en consideración son los siguientes: Regresión lineal con una capa densa o 'fully connected', Regresión logística, técnicas de kernel como Gaussian Process y una red neuronal sencilla, con una o dos capas ocultas.

Utilizar una red recurrente LTSM o una red convolucional permitiría al modelo tener en cuenta varias ventanas de audio simultáneamente y su posicionamiento en el tiempo. Esta información le permitiría aprender sobre el contexto de la voz y potencialmente podría producir resultados más fieles. La limitación de estos modelos para nuestra tarea está en su demanda computacional, la cual probablemente sea inviable para nuestro hardware. También, requieren un dataset más amplio que el que podemos recolectar en el tiempo asignado a este proyecto. Esto último se podría solucionar utilizando un autoencoder como modelo de regresión y realizando una primera fase de aprendizaje no supervisado sobre muestras de audio sin clasificar.

3. Metodología

3.1. Estudio de viabilidad

Se comprueba la viabilidad de la propuesta instalando el software que será necesario para ejecutar la aplicación en el dispositivo objetivo. El software verificado ha sido: Python2.7, PyTorch, Numpy, PyWorld. Este ha sido instalado con éxito en la Raspberry Pi.

Se comprueba que la herramienta World produce resultados suficientemente buenos para nuestro propósito. Esto se hace tanto en equipos de usuario como en las Raspberry Pi mediante unas muestras de audio. Con los experimentos realizados hasta el momento no se puede llegar a una conclusión sobre si será posible la ejecución en tiempo real.

3.2. Desarrollo

Se decide utilizar Google Colaboratory como entorno de desarrollo. Se realizará un estudio de la herramienta World para elegir cuales de sus funciones desplegar y optimizar sus parámetros a nuestra tarea. También se preparán funciones de Dynamic Time Warping para el alineamiento temporal de las muestras de entrenamiento. El modelo inicial con el que trabajaremos será una capa lineal densa cuyos pesos se obtendrán por mínimos cuadrados. Cuando se obtenga un conjunto de entrenamiento lo suficientemente grande se realizará el entrenamiento.

Trabajaremos con un repertorio de muestras de audio acotadas. Primero intentaremos aplicar nuestro sistema sobre voces diciendo una única palabra: 'uno'. Después extenderemos el dataset a las palabras correspondientes a los 10 dígitos del sistema decimal.

3.3. Despliegue

Una vez obtenidos unos resultados aceptables en el entorno de desarrollo se desplegará el sistema en la Raspberry Pi y se verificará su funcionamiento.

4. Trabajo futuro

Como propuestas de ampliación se plantea el uso de potenciómetros que permiten variar a voluntad del usuario algunos parámetros para afinar el dispositivo. Estos parámetros se obtendrían utilizando métodos que nos permitan extraer variables latentes relevantes de la información: Una red en forma de embudo (autoencoder), PCA, etc. También se plantea el entrenamiento con secuencias largas de audio, que contengan gran variaedad de fonemas, para poder utilizar nuestro sistema para cualquier secuencia de palabras pronunciadas por el interlocutor.

Referencias

[1] Masanori MORISE, Fumiya YOKOMORI, and Kenji OZAWA. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, E99.D(7):1877–1884, 2016.

[2] 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. *ArXiv e-prints*, September 2016.

[3] Zhizheng Wu, Oliver Watts, and Simon King. *Merlin: An Open Source Neural Network Speech Synthesis System*. 9 2016.