Reconocimiento de voz mediante Deep Learning utilizando la STFT y la FHT

JUAN CAMILO OSPINA VILLA[1], AGUSTÍN LÓPEZ ZAPATA[1], ANA SOFÍA CALLE MUÑOZ[1], AND LAURA PATIÑO RESTREPO[1]

[*] jcospinav@eafit.edu.co, alopezz1@eafit.edu.co, ascallem1@eafit.edu.co, lpatinor@eafit.edu.co

Compiled November 8, 2023

Abstract: The audio signal processing is very important in current affairs such as security, sound, music and entertaining engineering. This document presents a project that focuses on the speech analysis using the analytical solution of the Short-Time Fourier Transform (STFT) in comparison to the Hankel Transform (FHT) to be able to generate spectrograms of different speakers. Additionally, it outlines the training process of a deep learning model through fine-tuning to be able to differentiate or classify these speakers. The accuracy of the model obtained with STFT was 94.922% and with FHT was 77.704%.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCCIÓN

La mayoría de las señales en la vida real son desconocidas y no deterministas. Si se desea hacer un análisis de estas señales, estas se deben representar en función de señales deterministas conocidas por medio de combinaciones lineales de una base ortogonal [1]. El método más común para lograrlo es en términos de la Transformada de Fourier (FT) que la descompone en sinusoidales [1], sin embargo, existen otras formas alternativas como la transformada de Fourier-Bessel (FBT), también llamada transformada de Hankel junto con sus respectivas series de Fourier -Bessel, que descompone la señal como una combinación lineal de las funciones de Bessel, usualmente $J_0(t)$ (orden-0) y $J_1(t)$ (orden-1) [1]. Siendo esta una alternativa que se favorece de su comprtamiento no estacionario, al igual que la mayoría de señales en la naturaleza, de su representación real que contiene únicamente frecuencias positivas haciendo más sencilla la descomposición de la señal y de su comportamiento amortiguado que resulta en una menor distorsión de la señal analizada[1].

El procesamiento de señales de audio específicamente es una línea de investigación altamente abordada recientemente, pues tiene aplicaciones importantes como clasificación de audio, reconocimiento de voz simulando como lo haría el oído humano, verificación del locutor aprovechando las diferencias de tono y timbre de voz de cada uno para términos de seguridad biométrica, recuperación de información musical, eliminación del ruido, entre otras. Además, este procesamiento de audio puede hacerse mediante la aplicación de modelos de aprendizaje entrenados, utilizando las transformadas menciondas anteriormente, como se abordará durante este proyecto.

2. MARCO TEÓRICO

A. Descriptores de audio en el dominio temporal

Usualmente podemos representar un sonido complejo mediante un diagrama de presión atmosférica a través del tiempo que se ilustra por medio de la supersposición de formas de onda sinusoidales con puntos de compresión, donde existe una mayor densidad de partículas del aire colisionando entre sí, y puntos de rarefacción donde existe una menor densidad de colisiones como se puede observar en la Figura 1 . Estos diagramas proporcionan información de la frecuencia, la intensidad y la fase de los sonidos de la señal, en términos de su longitud de onda λ, su amplitud A y su periodo T.

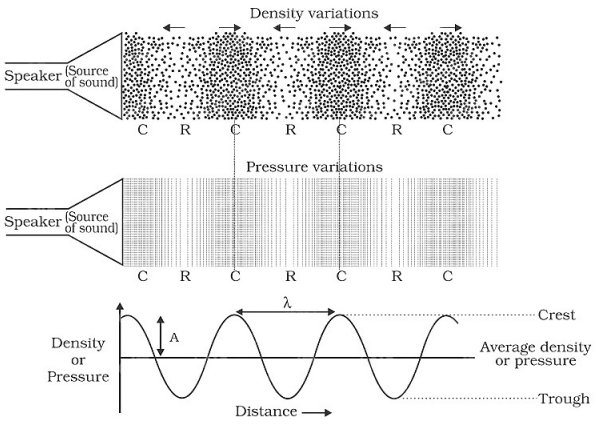

Fig. 1. Diagrama de onda sonora[2].

La interpretación del concepto de tono se relaciona con la frecuencia fundamental de forma logarítmica, permitiendo conocer si el sonido es agudo o grave. Por otro lado, el concepto de timbre permite hacer la diferenciación de las fuentes de sonido aún si tienen la misma intensidad, duración y frecuencia fundamental(la frecuencia más baja de la señal) por medio de: i) los armónicos que son múltiplos de la frecuencia fundamental

pero no todos se manifiestan siempre en todas las fuentes de sonido ni con la misma intensidad ii) la envolvente de la amplitud une todos los pico máximos de amplitud de cada una de las sinusoidales y que hace referencia a la evolución de la intensidad en el tiempo durante las fases de ataque A, decaimiento D, sostenimiento S y liberación R iii) modulación de frecuencia y amplitud mediante ondas portadoras y ondas moduladoras utilizadas para expresiones musicales [3]. Los factores mencionados que influyen en el timbre pueden verse en la Figura 2.

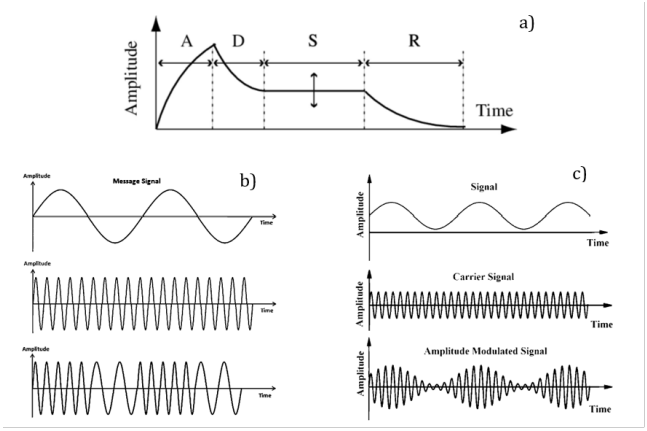

Fig. 2. a) envolvente de amplitud b) modulación de frecuencia c) modulación de amplitud [3].

B. Acondicionamiento de señales de audio

Se tiene conocimiento por lo mencionado en la sección anterior que los sonidos son ondas mecánicas expresadas gráficamente por su cambio en la presión atmosférica en el tiempo, por lo tanto, los sonidos son señales analógicas, es decir, que son continuas y reales con probabilidad de ser infinitas en valores del tiempo y en valores de la amplitud. Para ser interpretadas y manipuladas por un computador, estos valores deben volverse discretas mediante un procedimiento llamado conversión analógica-digital que contiene a su vez los procesos de muestreo y cuantización que se obsesrvan en la Figura 3. El proceso de muestreo hace referencia a la toma muestras de una señal en intervalos periodicos de tiempo siguiendo el teorema de Nyquist que indica que la frecuencia de muestreo (1/T) debe ser al menos el doble de la mayor frecuencia de la señal. El proceso de cuantización es la asignción de valores discretos de amplitud a las muestras tomadas en términos de bits.

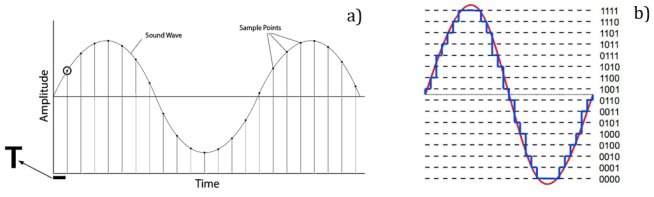

Fig. 3. a) proceso de muestreo b) proceso de cuantización [3].

Una vez se tiene la discretización o señal digital, se debe realizar el framing que se refiere a la separación de la señal discretizada en segmentos llamados "frames" cuyo tamaño será el número de muestras discretas que abarca. Este proceso se realiza debido a que las frecuencia de las señales cambian con el tiempo, por lo que la trasnformada de Fourier para hacer el paso al dominio frecuencial no debe hacerse sobre toda la señal. Estos frames se deben traslapan debido a que de no hacerlo se pierde información de la señal porque las señales procesadas no tienen un número entero de periodos y los funtos finales de cada ventana presentan descontinuidades que son mal interpretadas como frecuencias altas.

Para el dominio de la frecuencia, posterior a este paso, se debe llevar acabo el windowing o segmentación por ventanas, que serán abordadas ampliamente en el documeno, antes de aplicar la transformada de Fourier de tiempo reducido (STFT). Finalmente, con el resultado de los pasos aneriores, se puede hacer el cálculo o gráfica de características del sonido deseadas.

C. Espectrogramas y Espectrogramas de Mel

Un espectrograma es una representación gráfica de la frecuencia en función del tiempo donde se observa la intensidad o contribución de diferentes componenetes frecuenciales a lo largo del tiempo y que se obtiene aplicando la tranformada de Fourier de tiempo reducido (STFT), que será abordada más a profundidad en la sección 3A, a la señal de sonido en el dominio del tiempo. Es decir, que se se aplica la trasformada discreta de Fourier (DFT para los espectrogramas de Mel) u otra transformada para pasar al dominio de la frecuencia pero no sobre la señal completa sino por segmentos mediante la aplicación de funciones ventana que multiplican a la señal original.

Como se ha mencionado anteriormente, el tono es percibido por el oído humano de forma logarítmica mediante la relación de la frecuencia con la frecuencia de una escala de Mel que simula esta percepción humana; la convención de esta realción se puede observar en la Figura 4. La escala de Mel está diseñada pra reconocer diferencias más fácilemnte en frecuencias bajas que en frecuencias altas.

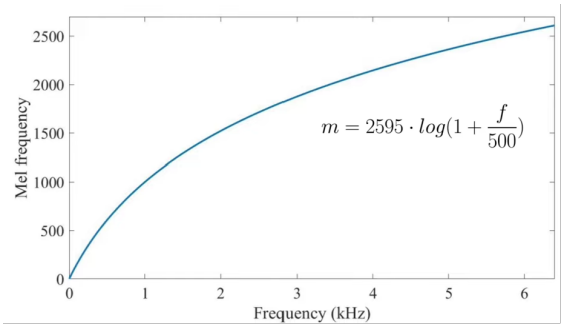

Fig. 4. Escala de Mel y ecuación de conversión [3].

Entonces, un espectrograma de Mel es un espectrograma cuyo eje -y- se ecuentra en la escala de Mel. A nivel computacional, esto se logra mediante el siguiente procedimiento: i) se hace partición del dominio o escala de frecuencia (en Hz) en la cantidad de bandas de frecuencia deseadas ii) se transforman las bandas de frecuencia a sus bandas de frecuencia de Mel correspondientes mediante la fórmula que se visualiza en la Figura 4 iii) se aplican filtros triangulares llamados filtros de Mel que capturan la energía de cada banda como se observa en la Figura 5 . A medida que las frecuencias son más altas, los filtros adquieren un mayor ancho de modo que los picos estén distanciados de acuerdo con la escala de Mel. Estos filtros se traslapan, es decir, no comienzan en el punto final del anterior, pero cubren todo

el rango de frecuencias de interés, de modo que cada uno de los filtros se multiplica por cada uno de los puntos discretos del resultado de la aplicación de las ventanas con STFT. Entre mayor número de filtros de Mel se obtiene una maypr resolución en el dominio de la frecuencia, sin embargo, entre mayor número de filtros de mel también aumenta el costo computacional, por lo que se debe hacer un balance de estas variables.

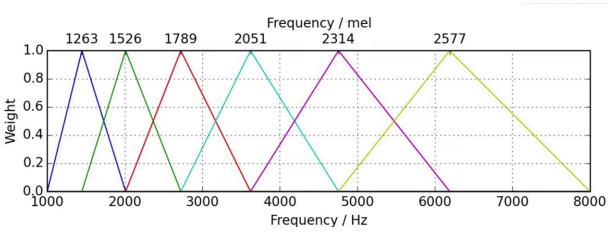

Fig. 5. Filtros triangulares de Mel [3].

D. Redes Neuronales Convolucionales

En primer lugar, las redes neuronales artificiales (ANN) comunes son funciones inspiradas en el comportamiento del cerebro humano, que se utilizan en el procesamiento de datos y aprendizaje automático. En las redes neuronales, cada una de los nodos computacionales o neuronas están interconectados a traves de varias capas (entrada, intermedias u ocultas y salida), los nodos procesan cada una de las entradas y las convierten en una salida optimizada por medio de la aplicación de una función de activación o una no-linearidad que resulta en la determinación y evluación de los pesos que tendrá cada interconexión de neuronas, es decir, qué tanta influencia representa una neurona en otra. Un esquema de modelo de resdes neuronales se puede observar en la Figura 6.

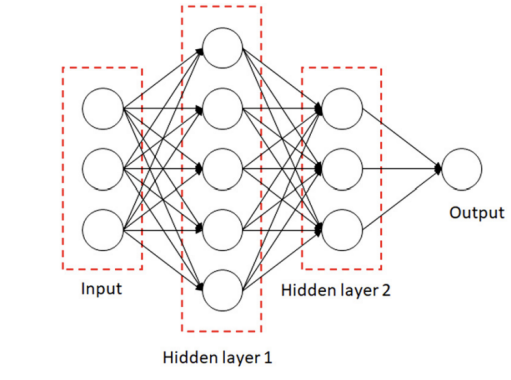

Fig. 6. Esquema de redes neuronales [4].

Las redes neuronales convolucionales (CNN) son redes neuronales que se especializan en reconocimiento de patrones y por lo tanto son mayormente utilizadas para el procesamiento de imágenes y señales de audio. Para el caso de este proyecto, los patrones que se identifican se encuentran en los espectrogramas que se encuentran en formato `.jpg`. En las redes neuronales de convolución, la trasnformación que se lleva a cabo en cada una de las neuronas de las capas ocultas (capas de convolución con un determinado número de filtros o kernels), es una operación de convolución que captura la invariancia de traslación, es decir, que los filtros son independientes de la ubicación de entrada de las carcterísticas que son evaluadas debido a que se tiene el mismo peso compartido en los elementos de la capa de convolución [4].

A su vez, en las redes neuronales convolucionales existen varias arquitecturas posibles, el presente proyecto utiliza la aquitectura ResNet 18 en la cual, en cada una de las 18 capas, se tiene un procedimiento de convolusión, una normalización del batch para que sus valores se encuentren entre 0 y 1 y se acelere el aprendizaje, y finalmente la capa ReLU que es una función de activación que verifica la linearidad, es decir, toma solo los valores positivos.El proceso de fine-tuning consiste en tomar esta arquitectura de red neuronal pre entrenada y modificar la capa de entrada y la capa de salida que en el presente proyecto son categorías con cada uno de los autores de voz que se quiere identificar.

3. SOLUCIÓN ANALÍTICA

A. Series de Fourier y transformada de fourier de tiempo reducido

Las series de fourier se utilizan para representar funciones periódicas en términos de senos y cosenos. Tienen una gran utilidad a la hora de resolver ecuacuiones diferenciales ordinarias y de varias variables. Las series de Fourier permiten trabajar con funciones que tienen discontinuidades, como el diente de sierra, valor absoluto, funciones cuadradas entre otras. Es relevante desarrollar la teoría de las series de Fourier para tener una base para comprender técnicas más avanzadas que analizan señales complejas que varían en el tiempo.

Se puede representar una onda periódica no armónica en sus componentes de frecuencia individuales armónicos. Cada una de las frecuencias contribuye en la forma de la onda original, en donde se deduce que se debe hacer una sumatoria de dichas funciones de onda. Una vez hecha la sumatoria, esta se puede plasmar en el dominio temporal en donde se visualiza su periodo, amplitud y comportamiento:

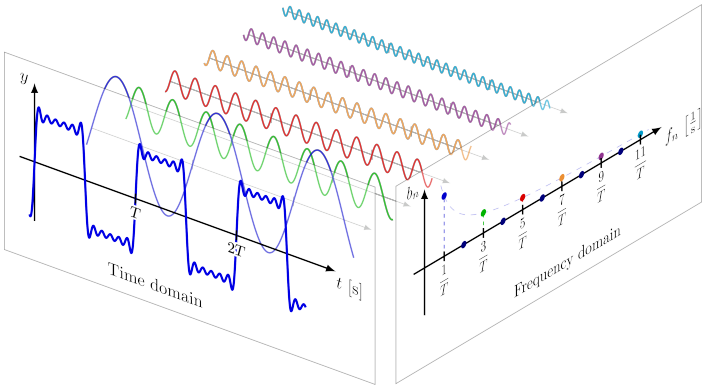

Fig. 7. Series de Fourier en el dominio temporal y frecuencial [5]-

La función de la onda compuesta se expresa como la siguiente sumatoria:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n cos(nt) + \sum_{n=1}^{\infty} b_n sin(nt) \qquad \textbf{(1)}$$

En donde los coeficientes a_0, a_n y b_n estan relacionados con la función $f(x)$ y se calculan como integrales definidas.

Una de las propiedades de la serie es la completitud, que se refiere a la posibilidad de que un conjunto de funciones se combine para aproximar una función continua y periódica en un intervalo [6]. Observe la base trigonométrica propuesta en 1809 por Fourier:

$$\phi(t) = \left\{ \phi_{1n} = \frac{1}{\sqrt{\pi}} sin(nt), \phi_{2n} = \frac{1}{\sqrt{\pi}} cos(nt), \phi_{3n} = \frac{1}{\sqrt{2\pi}} \right\} \tag{2}$$

Con $n = 0, 1, 2, 3....$ Esta base trigonométrica son funciones propias de la siguiente EDO autoafjunta lineal [6]:

$$y'' + n^2 y = 0 \tag{3}$$

Las funciones propias ortogonales tienen un valor propio de n en el intervalo $[0, p\pi]$. p es un entero que satisface las condiciones de frontera de la teoría de Sturm-Liouville. Cuando p es el entero 2, las funciones propias para el mismo valor de n resultan ser ortogonales bajo el producto interno y se obtienen las siguientes condiciones [6]:

$$\int_0^{2\pi} sin(mt) sin(nt) dt = \left\{ \begin{array}{c} \pi\, \delta_{m,n},\ m \neq 0 \\ 0,\ m = 0 \end{array} \right\} \tag{4}$$

$$\int_0^{2\pi} cos(mt) cos(nt) dt = \left\{ \begin{array}{c} \pi\, \delta_{m,n},\ m \neq 0 \\ 2\pi,\ m = n = 0 \end{array} \right\} \tag{5}$$

$$\int_0^{2\pi} sin(mt) cos(nt) dt = 0 \tag{6}$$

La Eq. (6) siendo válida para todos los n, m enteros. En la teoría de Sturm-Liouville, es posible llegar a los coeficientes de las series de Fourier gracias a las anteriores propiedades. Al multiplicar una función cualquiera con una función propia y luego se integra sobre el intervalo, se obtiene un término no nulo si ambas funciones son ortogonales [6]. Se obtienen las siguientes expresiones:

$$a_o = \frac{1}{\pi} \int_0^{2\pi} f(t) dt \tag{7}$$

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(t) cos(nt) dt \tag{8}$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(t) sin(nt) dt \tag{9}$$

n siendo un entero mayor a 0. Para señales no periódicas se hace la asumpción de que el periodo de aquellas funciones es infinito. Al hacer el límite, las frecuencias que antes eran discretas n se vuelven continuas entonces se debe hacer una integral. Entonces, se introduce la transformada de Fourier de una función de onda no periódica $f(t)$ como:

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi f t} dt \tag{10}$$

Ahora, una de las transformadas utilizadas en este proyecto para procesar un audio de voz que varía en el tiempo es la transformada de fourier de tiempo reducido. Se tiene una señal $x(n)$ la cual está definida en todo n; luego, $X_n(e^{jw_k})$ es la transformada de tiemo reducido de la señal en un tiempo n y frecuencia w_k [7]. En este desarrollo se asume el uso de filtros de procesamiento de señal simétricos y uniformemente espaciados en el dominio frecuencial. Lo anterior, con el objetivo de utilizar solo una función de ventana $w(n)$ la cual hace el papel de un filtro pasa bajos (filtra altas frecuencias) en segmentos específicos del tiempo de la señal. Toda la información crucial del filtro (como la resolución temporal y espectral) se encuentra en $w(n)$. Observe la expresión de STFT:

$$X_n(e^{jw_k}) = \sum_{m=-\infty}^{\infty} w(n-m) x(m) e^{-jw_k m} \tag{11}$$

Observe que la ventana selecciona el segmento de la señal $x(n)$ que se quiere analizar. La Eq. (11) también se puede interpretar como un conjunto de filtros en donde $X_n(e^{jw_k})$ es una función para n con una frecuencia w_k fija [7]. La STFT se puede reescribir como una convolución lineal de la señal con una respuesta al impulso $w(n)$:

$$X_n(e^{jw_k}) = [x(n) e^{-jw_k n}] * w(n) \tag{12}$$

La modulación que porporciona $e^{-jw_k n}$ a la señal permite desplazar el espectro de frecuencias de la señal en w_k hasta 0 [7]. Es decir, el STFT se reinterpreta como una filtración del espectro de una señal a cierta frecuencia por medio de un filtro pasa bajos $w(n)$.

Algo importante en la tarea computacional es la resolución tiempo-frecuencia. Este concepto da cuenta de las propiedades espectrales y temporales de una señal que varía en el tiempo. En la técnica STFT la esta información se determina por medio de las ventanas. Cuando una ventana es estrecha, esta permite obtener mayor resolución temporal; se distinguen pequeños cambios de amplitud o tono en la señal en un corto lapso de tiempo. Con una ventana más grande, se logra reunir los componentes frecuenciales en dicho intervalo temporal [8]. Antes de desarrolar matemáticamente lo dicho anteriormente, considere que el STFT se realiza sobre un dominio continuo, entonces la expresión Eq. (11) se puede escribir:

$$F_x^{\gamma}(\tau, w) = \int_{-\infty}^{\infty} x(t) \gamma^*(t-\tau) e^{-jwt} dt \tag{13}$$

Donde $F_x^{\gamma}(\tau, w)$ es la STFT, x(t) la señal, $\gamma^*(t-\tau)$ la función ventana desplazada en el tiempo y $e^{-jwt}dt$ el modulador de la frecuencia. Si se tiene la función de la ventana desplazada en el dominio temporal τ unidades hacia la derecha y modulada en frecuencia, se puede hacer esta igualdad [8]:

$$\gamma_{\tau;w}(t) = \gamma(t-\tau) e^{jwt} \tag{14}$$

Ahora, seguimos con la aplicación del principio de desplazamiento de Fourier [8]. Se procede a hacer la transformada de Fourier a ambos lados de la igualdad:

$$\Gamma_{\tau;w}(\nu) = \Gamma[\gamma(t-\tau)] e^{jwt} \tag{15}$$

Tenga en cuenta que la transformada de Fourier de un producto en el dominio temporal se puede reescribir como una convolución en el dominio de la frecuencia:

$$\Gamma[fg] = \Gamma[f] * \Gamma[g] \tag{16}$$

$$\Gamma_{\tau;w}(\nu) = \Gamma[\gamma(t-\tau)] * \Gamma[e^{jwt}] \tag{17}$$

Se hace la transformada de Fourier para una onda compleja, la cual es la delta dirac desplazada. En sí, se utiliza el principio de modulación de la transformada de Fourier en el dominio temporal, el cual equivale a un desplazamiento frecuencial:

$$\Gamma[e^{jwt}] = 2\pi\delta(\nu - w) \tag{18}$$

Ahora, para simplificar la parte de $\Gamma[\gamma(t-\tau)]$ se utiliza el principio de desplazamiento en el dominio frecuencial. Lo anterior dicta que un desplazamiento temporal equivale a una fase introducida en el dominio de la frecuencia:

$$\Gamma[\gamma(t-\tau)] = \Gamma[\gamma(t)]e^{-j\nu\tau} \tag{19}$$

Que reemplazado en Eq. (18) y Eq. (19) en Eq. (17) queda:

$$\Gamma_{\tau;w}(\nu) = \Gamma[\gamma(t)]e^{-j\nu\tau} * 2\pi\delta(\nu - w) \tag{20}$$

Se utiliza la propiedad de sifting para la delta de dirac, que simplifica la convolución de una función con la delta y se denota:

$$f(\nu) * \delta(\nu - w) = \int_{-\infty}^{\infty} f(\xi)\delta(\xi - w)d\xi = f(\nu - w) \tag{21}$$

Al aplicarla sobre Eq. (20) se obtiene:

$$\Gamma_{\tau;w}(\nu) = \Gamma[\gamma(t)]e^{-j(\nu-w)\tau} * 2\pi\delta(\nu - w) \tag{22}$$

En donde la fase añadida no afecta la propiedad de sifting. Procedemos a hacer la propiedad de convolución de la delta, se simplifica y normaliza:
Al aplicarla sobre Eq. (20) se obtiene:

$$\Gamma_{\tau;w}(\nu) = 2\pi\Gamma[[\gamma(t)]]_{\nu=w}e^{-j(\nu-w)\tau} \tag{23}$$

$$\Gamma_{\tau;w}(\nu) = \Gamma[\gamma(t)](\nu - w)e^{-j(\nu-w)\tau} \tag{24}$$

Luego, se utiliza la relación de Perseval [8] la cual dice que la energía total de una señal en el dominio temporal es igual a la energía total en su transformada de Fourier en el dominio frecuencial. Se calcula de la siguiente manera y se aplica al resultado Eq. (24):

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(w)|^2 dw \tag{25}$$

$$\langle x, \gamma_{\tau;w}\rangle = \int_{-\infty}^{\infty} x(t)\gamma^*(t-\tau)e^{-jwt}dt \tag{26}$$

Observe que aplicando la relación de Perseval en Eq. (27) el producto interno en el tiempo es proporcional a aquel en frecuencia.

$$\langle x, \gamma_{\tau;w}\rangle = \frac{1}{2\pi}\langle X, \Gamma_{\tau;w}\rangle \tag{27}$$

Ahora, se procede a calcular en el dominio de la frecuencia:

$$\langle x, \gamma_{\tau;w}\rangle = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\nu)\Gamma^*(\nu - w)e^{j(\nu-w)\tau}d\nu \tag{28}$$

$$F_x^{\gamma}(\tau, w) = e^{-jw\tau}\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\nu)\Gamma^*(\nu - w)e^{j\nu\tau}d\nu \tag{29}$$

En la Eq. (29) se puede decir que la STFT se puede hallar integrando el producto de la transformada de Fourier de la señal y la ventana conjugada compleja que tiene un desplazamiento en frecuencia y un término de modulación. Esta ecuación dicta la base de la relación entre la resolución temporal y frecuencial debido a la función de la ventana $\gamma(t)$ y como afecta la transformada de fourier de la señal. Cuando la función $\gamma(t)$ es ancha en el tiempo, la transformada de Fourier de la señal es más estrecha en el dominio de la frecuencia lo que permite gran resolución de los componentes de frecuencia. En caso de que la ventana corte un tiempo corto, se traduce en una transformada de Fourier más amplia en el dominio frecuencial lo que da paso a distinguir frecuencias cercanas y revela estructuras armónicas propias de cada señal (¡como las voces!). El término de modulación de frecuencia en la expresión incluye una τ la cual se desplaza por la señal y que permite visualizar distintos segmentos a través del tiempo [8]. Este compromiso entre resoluciones da paso al principio de incertidumbre el cual solo permite alta resolución ya sea temporal o frecuencial, pero no ambas por la naturaleza de la ecuación [8].

En tanto a los espectrogramas, es necesario utilizar una técnica de visualización ya que como vimos en el desarrollo el STFT tiene muchos términos complejos [8]. Se puede obtener el espectrograma de un STFT con la siguiente expresión:

$$S_x(\tau, w) = \left|F_x^{\gamma}(\tau, w)\right|^2 \tag{30}$$

Que es equivalente a:

$$S_x(\tau, w) = \left|\int_{-\infty}^{\infty} x(t)\gamma^*(t-\tau)e^{-jwt}dt\right|^2 \tag{31}$$

Observe una imágen del proyecto donde se tiene la onda de voz y su respectivo espectrograma por medio del uso de esta expresión:

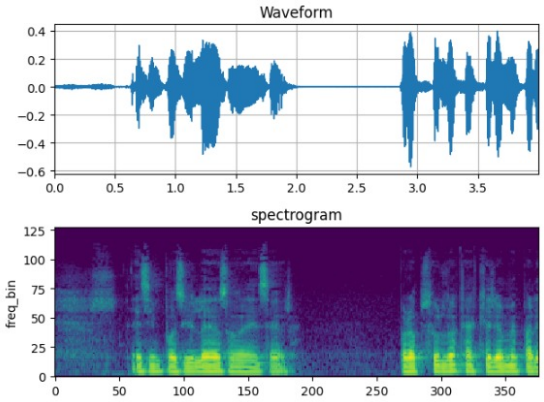

Fig. 8. Señal de voz y su espectrograma.

La visualización da a conocer como las frecuencias cambian con el tiempo, además de que los colores respresentan la distribución de la magnitud de la energía. Lo anterior debido a que la expresión $S_x(\tau, w)$ calcula la magnitud al cuadrado de una señal compleja, lo que resulta ser su potencia o la energía instantánea [8].

B. Representación Fourier Bessel y transformada de Hankel

En el proyecto se indaga acerca de la viabilidad de utilizar las funciones de Bessel como base para representar una onda variable en el tiempo. Estas funciones aparecen en las soluciones de la ecuación diferencial de segundo grado de Bessel. Ahora,

dicha ecuación diferencial surge como una forma particular de la ecuación de Sturm-Liuville debido a la aplicación de condiciones de frontera específicas [6]. Se procede a deducir la ecuación diferencial de Bessel a partir de la ecuación de Helmholtz en coordenadas cilindricas [6]:

$$\nabla^2\psi(\rho,\varphi,z)+k^2\psi(\rho,\varphi,z)=0 \tag{32}$$

Se reemplaza el laplaciano en coordenadas cilíndricas:

$$\frac{1}{\rho}\frac{\partial}{\partial\rho}(\rho\frac{\partial\psi}{\partial\rho})+\frac{1}{\rho^2}\frac{\partial^2\psi}{\partial\varphi^2}+\frac{\partial^2\psi}{\partial z^2}+k^2\psi=0 \tag{33}$$

En donde la función ψ tiene la forma:

$$\psi(\rho,\varphi,z)=P(\rho)\Phi(\varphi)Z(z) \tag{34}$$

Se reemplaza en la Eq. (33) y se obtiene una ecuación diferencial de derivadas ordinarias:

$$\frac{\Phi Z}{\rho}\frac{\mathrm{d}}{\mathrm{d}\rho}(\rho\frac{\mathrm{d}P}{\mathrm{d}\rho})+\frac{PZ}{\rho^2}\frac{\mathrm{d}^2\Phi}{\mathrm{d}\varphi^2}+P\Phi\frac{\mathrm{d}^2Z}{\mathrm{d}z^2}+k^2P\Phi Z=0 \tag{35}$$

Se procede a dividir sobre el factor $P\Phi Z$:

$$\frac{1}{\rho P}\frac{\mathrm{d}}{\mathrm{d}\rho}(\rho\frac{\mathrm{d}P}{\mathrm{d}\rho})+\frac{1}{\rho^2\Phi}\frac{\mathrm{d}^2\Phi}{\mathrm{d}\varphi^2}+k^2=-\frac{1}{Z}\frac{\mathrm{d}^2Z}{\mathrm{d}z^2} \tag{36}$$

Se asumen las siguientes equivalencias debido a convención histórica las cuales permiten una simplificación sencilla [6]:

$$\frac{\mathrm{d}^2Z}{\mathrm{d}z^2}=l^2Z \tag{37}$$

$$k^2+l^2=n^2 \tag{38}$$

Se reemplazan en la Eq. (36) y se multiplica a ambos lados por ρ^2 para obtener:

$$\frac{\rho}{P}\frac{\mathrm{d}}{\mathrm{d}\rho}(\rho\frac{\mathrm{d}P}{\mathrm{d}\rho})+n^2\rho^2=-\frac{1}{\Phi}\frac{\mathrm{d}^2\Phi}{\mathrm{d}\varphi^2} \tag{39}$$

Se hace la siguiente equivalencia al lado derecho de la ecuación y se reemplaza en Eq. (39):

$$\frac{\mathrm{d}^2\Phi}{\mathrm{d}\varphi^2}=-m^2\Phi \tag{40}$$

$$\rho\frac{\mathrm{d}}{\mathrm{d}\rho}(\rho\frac{\mathrm{d}P}{\mathrm{d}\rho})+(n^2\rho^2-m^2)P=0 \tag{41}$$

La expresión anterior es la E.D. de Bessel, la cual se puede reescribir de la siguiente manera [1]:

$$x^2\frac{d^2}{dx^2}+x\frac{dy}{dx}+(x^2-\nu^2)y=0 \tag{42}$$

La solución a la anterior ecuación se encuentra por medio del método de Frobenius [1]. La solución también es llamada la ecuación de Bessel, que tiene la forma:

$$y=C_1J_v(x)+C_2Y_v(x) \tag{43}$$

En donde J_v y Y_v son las funciones de Bessel de primera y segunda especie de órden ν [1]. La función de Bessel de primera especie se caracteriza por ser finita y no singular en el origen, la cual se suele usar en problemas de calor. Por el otro lado, la función de Bessel de segunda especie es infinita y singular en el origen y se usan para describir fenómenos ópticos como la difracción. Estas ecuaciones de Bessel están presentes en la segunda transformada utilizada en el proyecto, que es la transfomrada de Fourier-Bessel. Esta transformada permite descomponer una función $f(t)$ que dependa de una distancia radial cilíndrica, integrable y continua en infinitas sumas de funciones de Bessel de primera especie [1].

$$F(\eta)=\int_0^\infty tf(t)J_\nu(\eta t)dt \tag{44}$$

En donde η es una variable de transformación que se interpreta como una frecuencia radial si $f(t)$ dependiera de una distancia radial cilíndrica. Las funciones de Bessel de primera especie de órden ν obedecen la siguiente propiedad de ortogonalidad [1]:

$$\int_0^\infty tJ_\nu(\eta t)J_\nu(\eta' t)dt=\frac{\delta(\eta-\eta')}{\eta} \tag{45}$$

Observe que dentro de la integral t es la función de peso y es fundamental para que la propiedad de ortogonalidad de mantenga. La ortogonalidad se definirá de acuerdo al resultado del producto punto de las dos funciones de Bessel y la función de peso. Ahora, al lado derecho de la propiedad está la delta de Dirac la cual adquiere un valor diferente de 0 cuando $\eta=\eta'$ y la integral es igual a 1 [1]. Por último, las series de Fourier-Bessel es una sumatoria infinita de funciones Bessel que multiplican un coeficiente a_n. Además, se considera que una variable de transformación discreta es ς_k y es la raiz m-ésima positiva de la función de Bessel de órden ν. Observe la serie:

$$f(t)=\sum_{m=1}^{\infty}a_nJ_\nu(\varsigma_m t) \tag{46}$$

Observe que a_n son los coeficientes que están relacionados con la energía de los modos vibracionales de un sistema físico. Es decir, cada voz contiene distintos coeficientes que contribuyen a la expansión de series y caracterizan la forma de su señal. Para hallar los coeficientes, se asume que Eq. (46) existe y converge en $[0,1]$ de fomra homogénea. Se procede a multiplicar ambos lados de Eq. (46) por la siguiente expresión:

$$\int_0^1 tJ_\nu(\varsigma_k t)dt \tag{47}$$

$$\int_0^1 tf(t)J_\nu(\varsigma_k t)dt=\sum_{m=1}^{\infty}a_m\int_0^1 tJ_\nu(\varsigma_m t)J_\nu(\varsigma_k t)dt \tag{48}$$

Y por propiedad ortogonal, se tiene que la integral del lado derecho equivale a la siguiente expresión cuando $\varsigma_m=\varsigma_k$:

$$\int_0^1 tJ_\nu^2(\varsigma_m t)dt=\frac{J_\nu{}'^2(\varsigma_m)}{2} \tag{49}$$

Reemplazando en Eq. (48):

$$\int_0^1 tf(t)J_\nu(\varsigma_k t)dt=a_k\frac{J_\nu{}'^2(\varsigma_k)}{2} \tag{50}$$

Se encuentra la ecuación de los coeficientes de la expansión de Fourier-Bessel.

$$a_k=\frac{2}{[J_\nu{}'(\varsigma_k)]^2}\int_0^1 tf(t)J_\nu(\varsigma_k t)dt \tag{51}$$

El espectrograma se hace de una manera similar a la Eq. (31), que es calculando el modulo cuadrado de la transformada de Fourier-Bessel.

4. SIMULACIÓN COMPUTACIONAL

El núcleo de la simulación de este proyecto es el ánalisis de los espectrogramas de diferentes voces utilizando la transformada de Fourier en tiempo reducido y la transformada rápida de Hankel. Teniendo en cuenta que un espectrograma es una representación visual de la energía de una señal en función del tiempo y la frecuencia [8], la implementación computaciuonal consiste entonces en la obtención de un Dataset de espectrogramas de diferentes audios de varias personas para entrenar un modelo de Deep Learning que categorice las voces de las mismas utilizando el software Python.
En esta sección se describe el procesamiento de los datos utilizados, así como la implementación de diferentes librerías que permiten la realización de estos espectrogramas acompañados de diferentes pruebas para varias personas. Luego, se expone el modelo de DL utilizado para realizar el reconocimiento de voz, tanto con los espectrogramas derivados de la transformada de Fourier como con los de la transformada de Hankel y los resultados obtenidos en cada una de las partes del proyecto.

A. Metodología

Como se mencionó, para realizar el entrenamiendo del modelo de DL se utilizan espectrogramas de diferentes voces. Inicialmente, se presenta que datos serán procesados y como se extraen, se filtran y utilizan los atributos de los mismos. Posterior a esto, se muestra como se obtienen los espectrogramas mediante ambas transformadas y su pre-procesamiento para generar un Dataset que pueda ser utilizado para el entrenamiento. Luego, se muestra la estructura del modelo y los paramétros escogidos para implementar el reconocimiento de voz.

A.1. Datos y su procesamiento

Los datos que se utilizan para el procesamiento son archivos de audio. Estos archivos son de tipo WAV (Waveform Audio File Format), que contienen la forma de onda de la señal de audio, es decir, la energía (magnitud) del sonido a través del tiempo. Con el fin de procesar los datos se utilizó la librería `librosa`. Inicialmente, se cargan los archivos de audio (tipo WAV) de la carpeta utilizando la función `librosa.load`, esta función crea un array de Numpy con la amplitud de la onda de audio a través del tiempo. Véase la representación de esta forma de onda en la figura 9.

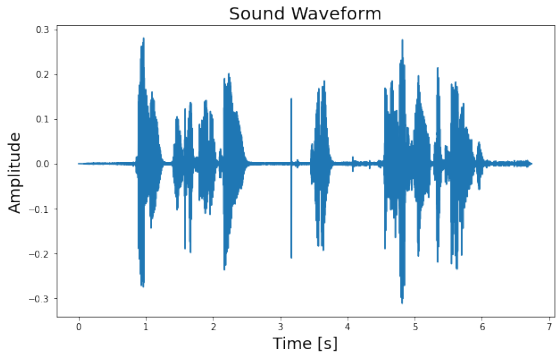

Fig. 9. Forma de onda de un audio

La forma de onda mostrada es para un audio que decía "Hola, esta es la primera prueba para realizar los espectrogramas". Con esta frase se realizan los espectrogramas demostrativos de la siguiente sección con diferentes voces.
Para generar el Dataset de entrenamiento del modelo se grabaron 14 minutos de audio de las voces de 6 personas. Los primeros 7 minutos de audio corresponden a la lectura del libro El Principito [9], y los otros 7 minutos a alguna otra obra literaria elegida por el locutor. Todos los audios se registraron el formator `.mp3` para garantizar homogeneidad el su procesamiento y representación.

Utilizando la librería `pydub` se dividieron todas las grabaciones en fragmentos de 4000 milisegundos y se convirtieron al formato `.wav`. Este último representa el sonido como una onda, tal como se aprecia en 9. Esta representación es óptima para poder realizarle a la señal las transformadas de Hankel y Fourier pertinentes para generar los espectrogramas cuya implementación se explica en el siguiente inciso.

Teniendo ya un total de 1321 fragmentos de audio, pasamos a realizar los espectrogramas respectivos. A cada dato se le aplica una transforamada MelSpectrogram perteneciente la librería `torchaudio.transforms.MelSpectrogram` [10] que utiliza internamente los métodos de procesamiento de audio de `librosa`. Los parámetros de la transformada se especifican en el siguiente inciso y su explicación en el inciso C del marco teórico. Si se desea ahondar en su implementación computacional se recomienda ver [10]. Cada espectrograma se guarda como una imagen `.jpg`. Posteriormente se repite el procemiento con espectrogramas a partir de la transformada de Hankel detallado también el el siguiente inciso y se guardan las imagenes en un dataset aparte con el mismo formato. Es importante recalcar que todos los spectrogramas se guardan sin barra de color, título, u información de los ejes, puesto que todos tienen la misma escala, y estos elementos no le brindan información adicional a la red.

A.2. Espectrogramas

Ya teniendo la forma de onda que se mostró en el anterior inciso es posible realizar los espectrogramas. Un espectrograma muestra cómo la energía de la señal está distribuida en diferentes componentes de frecuencia a lo largo del tiempo. Generalmente, se representa en un gráfico 2D con el eje horizontal que representa el tiempo y el eje vertical que representa la frecuencia [8]. Los colores del espectrograma representan la intensidad (magnitud relativa) de las diferentes frecuencias a lo largo del tiempo. Con el fin de obtener esta representación frecuencial se realiza la transformada de Fourier, que como se explicó en el análisis teórico brinda una descomposición en frecuencias de una función, en este caso, de una forma de onda de audio. Para realizar los espectrogramas se utilizó la transformada de fourier en tiempo reducido, expresada en la ecuación 31. Esta transformada está implementada en la librería `librosa` y la función es `librosa.stft` que tiene como paramétros la forma de onda, la longitud de la señal de ventana después de rellenar con ceros (1024 por lo general) y el número de muestras por columna de la transformada (512, que es la mitad del paramétro anterior). Como esta transformada tiene componentes de amplitud y fase se tiene que sacar la magnitud cuadrada de la salida de la transformación. Para visualizar el espectrograma se utilizó la función `librosa.display.specshow`, cambiando la escala de los datos ya que estos son lineales y para permitir una visualización adecuada debe someterse a la escala logaritimica en base 10. En el siguiente inciso se muestra el espectrograma correspondiente a la forma de onda mostrada en la figura 9.

La transformada de Fourier de tiempo reducido es por excelencia la transformada que permite visualizar y categorizar señales de audio de forma eficiente. Lo que se realiza ahora

es utilizar la transformada rápida de Hankel a modo de comparación, que como se mostró en el análisis teórico, utiliza las funciones de Bessel como base ortogonal y está representada por medio de una transformación radial. La función, en este caso de la librería `scipy`, que permite realizar la transformada es `scipy.fft.fht`. Esta función tiene como paramétros la forma de onda, el espaciamiento de los datos y el orden ν de la función de Bessel de primera especie. Al igual que con la transformada de Fourier se debe extraer la magnitud cuadrada de la amplitud y realizar el escalamiento logaritmico para permitir la visualización con la misma librería `librosa.display.specshow`. Se muestra el espectrograma correspondiente a la forma de onda 9 en el siguiente inciso.

Como se mencionó anteriormente en el artículo, los datos procesados para la red se realizaron directamente con la implementación de `torchaudio` [10] para espectrogramas de mel. A conitunuación se muestran los parámetros utilizados en caso de que se quieran replicar los resultados obtenidos:

```
n_fft = 1024
win_length = n_fft/2
hop_length = 512
n_mels = 128

mel_spectrogram = T.MelSpectrogram(
    sample_rate=sample_rate,
    n_fft=n_fft,
    win_length=win_length,
    hop_length=hop_length,
    center=True,
    pad_mode="reflect",
    power=2.0,
    norm="slaney",
    n_mels=n_mels,
    mel_scale="htk",
)
```

A.3. Modelo y entrenamiento

Para el proceso de entrenamiento utilizamos principalmente la librería `fastai`, que brinda una API de alto nivel construida sobre `pytorch` para implementar algoritmos de Deep Learning con técnicas a la vanguardia [11]. También se utilizaron algunas herramientas brindadas directamente por `pytorch` [12].

Primero preparamos los datos para la red con un objeto `DataLoaders`. La función de este es realizar todas las transformaciones pertinentes a los datos antes de ser ingresados a la red. En partícular se separó un 20% del dataset original para validación de forma completamnete aleatoria y se definieron batches de entrenamiento de 64 datos. Esto significa que se ingresan las imagenes al modelo en grupos de 64 imagenes, antes de aplicar backpropagation [11] y actualizar los parametros de la red. El proceso se repite hasta que el modelo vea todo el set de entrenamiento, lo cual se denomina como una época. El proceso de entrenamiento consiste en varias épocas. El target de cada dato, esto es el valor que buscamos predecir, se toma del nombre de la carpeta donde se encuentra ese dato en específico y corresponde al nombre de la persona a la cual pertenecen los espectrogramas en dicha carpeta.

Después de cargar los datos, se procede a instanciar un `Learner` de `pytorch` [12]. Este objeto cuenta con toda la información necesaria para llevar a cabo el proceso de aprendizaje. En particular se utiliza un `vision_learner` [11], una subclase que está optimizada para entrenar modelos de imagenes. El objeto recibe el `DataLoader` mencionado anteriormente, un modelo, en este caso tt XResnet18 [13], y una métrica para que podamos entender el desempeño del modelo, en este caso se utiliza sencillamente el ratio de error de la red. Cabe resaltar que la función de pérdida a optimizar la determina automáticamente el `vision_learner` con un proceso de mock training [11], la función determinada en este caso fue `CrossEntropyLossFlat` [12].

Antes de pasar al proceso de aprendizaje, es importante mencionar que se utilizó un proceso de transfer learning [11], en vez de entrenar la red desde 0, modificamos los afinamos los pesos actuales de la red para optimizarla al tipo de datos que queramos procesar. También se hace una pequeña modificación a la red, ya que tt XResnet18 fue entrenada originalmente para clasificar más de 20000 categorías de imágenes [13]. Para daptarla a nuestras necesidades removemos la última capa de activaciones que naturalmente es densa, pues tiene una neurona por cada categoría original, y la reemplzamos por un bloque de categorización de solo 6 elementos, correspondiente con las 6 voces presentes en el set de datos. También reemplazamos la cabeza para recibir un tamaño de imagen estándar de 465x308 pixeles.

Para realizar el entrenamiento se hace un último paso previo que es encontrar un learning rate (ratio de aprendizaje) óptimo para esta coponfiguración de la red. `Fastai` proporciona un método en `vision_learner` denominado `lr_find` [11] que encuentra determina un valor recomendado para este hiperparámetro. Esto se logra realizando varias secuencias de 'mock' training, iterando sobre el parámetro para encontrar el punto donde se comporta mejor. A continuación se muestra una gráfica del proceso en cuestión.

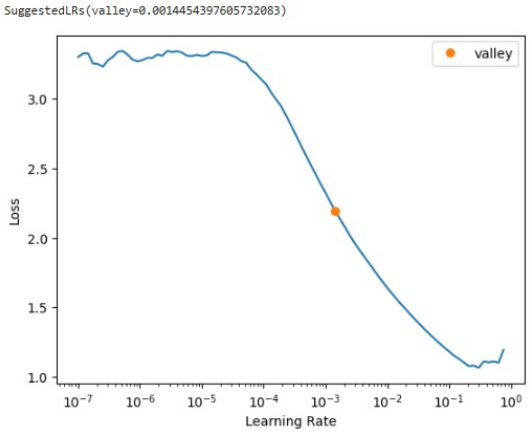

Fig. 10. Optimización del Ratio de Aprendizaje [11]

La recomendación oficial de Fastai [11] es utilizar el valor marcado en naranja en 10 dividido entre 10.

Con todo listo procedemos a entrenar la red con el método `fine_tune` por un total de 12 épocas (este es un valor experimental y puede modificarse a criterio propio). Los resultados del proceso de entrenamiento se presentan en la siguiente sesión.

Finalmente se hacen algunas evaluaciones al modelo. En primera instancia se saca la matriz de confuisión para el set de validación para estudiar cuales voces fueron más difíciles de reconocer y diferenciar. Luego se mide la precisión real de la red, o accuracy. Esto se logra tomando datos nuevos (que la red no ha visto nunca y no hacen parte ni del set de entrenamiento ni del set de validación). En este caso utilizamos espectrogramas correspondientes a segmentos de 4 segundos de audio provenientes de un total de 5 minutos de audio nuevo proporcionado por cada uno de los participantes con contenido a libre elección. Se utiliza el modelo para hacer inferencia en estos datos y se calcula la precisión como el número de prediciones correctas entre el

total de los datos a inferir en porcentaje. Por último se realiza una nueva matriz de confusión utilizando dichos resultados.

Si se quiere indagar mas en el proceso de aprendizaje puede consultar el proyecto en github Voice_Recognizer [14].

B. Resultados

En esta sección se muestran los resultados de cada una de las secciones mencionadas con anterioridad, detallando los aspectos relevantes para el entrenamiento y el comportamiento del modelo implementado.

B.1. Espectrogramas

A continuación se muestra el espectrograma correspondiente a la transformada de Fourier de tiempo reducido implementada a la muestra de audio evidenciada en la figura 9.

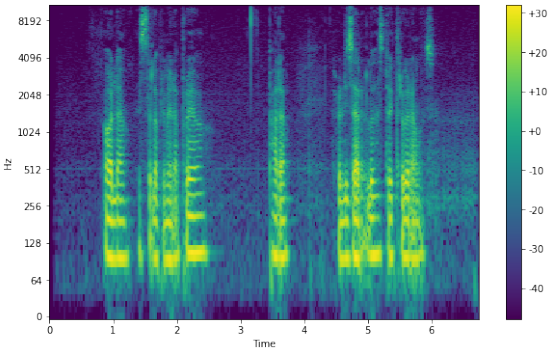

Fig. 11. Espectrograma de la fft a una forma de onda

Es posible evidenciar como el espectrograma, por medio del mapa de color, muestra la intensidad en decibeles de la frecuencia a través del tiempo. Específicamente en este espectrograma se nota que es proveniente de una voz grave dada la mayor concentración de intensidades en las frecuencias bajas.
Ahora se muestra la comparación entre el espectrograma anterior y otro proveniente de un audio diciendo la misma frase ("Hola, esta es la primera prueba para realizar los espectrogramas") dicho por una persona diferente. La figura 12 muestra ambos espectrogramas.

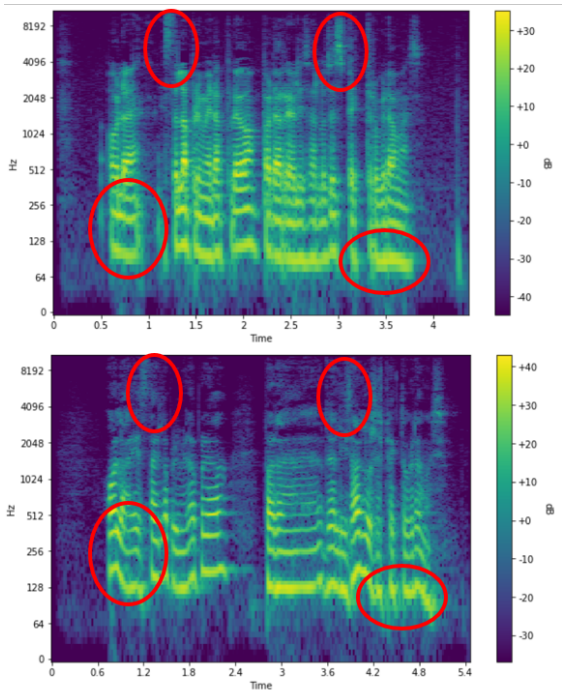

Fig. 12. Comparación de espectrogramas de la fft

En la figura 12 se logra evidenciar como cambian las distribuciones de intensidad frecuencial a través del tiempo. Los principales cambios se notan en las frecuencias bajas, alrededor de 128 *Hz* y 512 *Hz*, viéndose como la segunda persona presenta oscilaciones notorias en estas frecuencias, mientras que los cambios de frecuencia de la voz del primero son más suaves y con mayor densidad.
Ahora, se muestra el espectrograma correspondiente a la transformada de Hankel realizada a la forma de onda que se muestra en la figura 9.

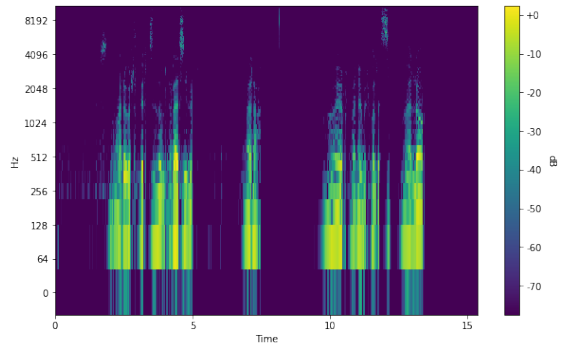

Fig. 13. Forma de onda de un audio

El espectrograma anterior muestra como según la transformada de Hankel, la magnitud de las frecuencias a través del tiempo para esta transformación es menor al nivel de referencia, representando atenuaciones para estas frecuencias. Cabe resaltar que esta transformada cuenta con un factor de transformación de frecuencia radial, por lo que la transformación presenta estas formas lineales según el aumento de la frecuencia y el paso del tiempo. Caso contrario a lo que ocurre con la transformada de fourier de tiempo reducido, donde se logra distinguir que el fac-

tor de transformación tiene simetría periódica, correspondiente al tipo de forma de onda que se está procesando.
A continuación, en la figura 14 se muestran dos espectrogramas de la transformación de hankel de dos formas de onda correspondientes a dos personas diciendo la misma frase.

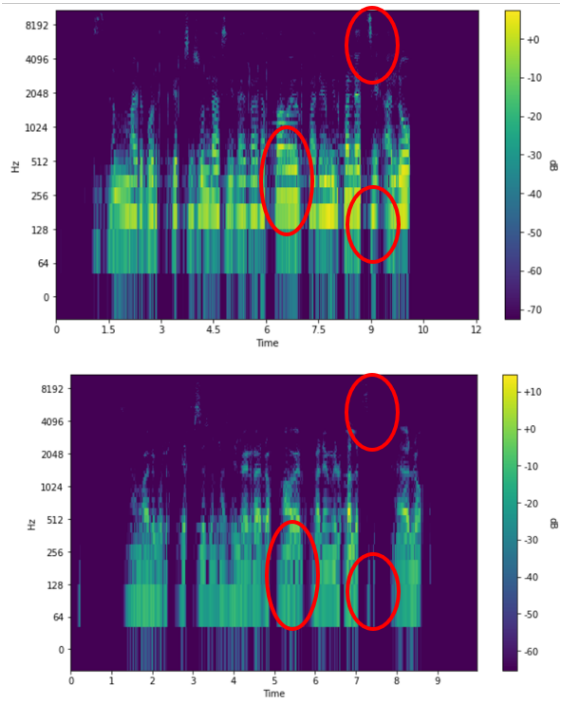

Fig. 14. Comparación de espectrogramas de la fht

De entrada es posible notar que para el modelo será complicado distinguir entre los espectrogramas generados por la transformación de Hankel debido a que las distribuciones de transformación de freuencia radial toman valores muy similares a través del tiempo. Notandose en el espectrograma como las mismas lineas rectas en los instantes del tiempo en una distribución de valor de frecuencia casi similar. Cabe resaltar que las intensidades de la frecuencia si son claramente distinguibles y como se concentran en zonas específicas del espectro, por lo que esta transformación permite analizar muy bien esta componente de la señal.

B.2. Modelo y entrenamiento

Con el Dataset generado, se genera un batch para visualizar los MelSpectrograms y los espectrogramas resultantes de la fht en el intervalo de separación de 4 segundos elegido aleatoriamente para todas las figuras entre los 7 minutos de grabación para todos los participantes. (Ver figura 15)

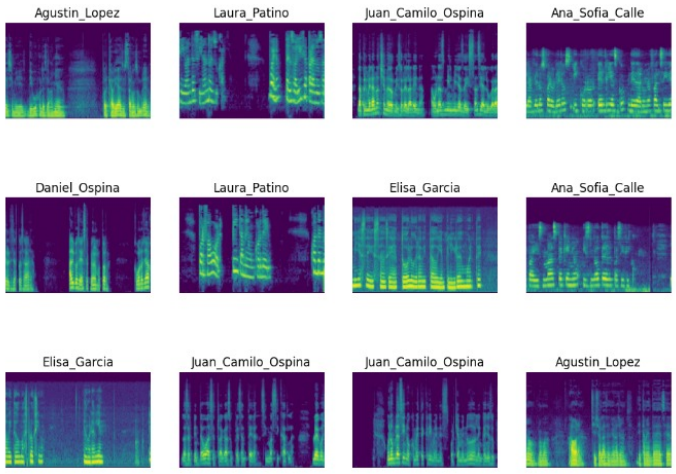

Fig. 15. Batch de MelSpectograms del Dataset

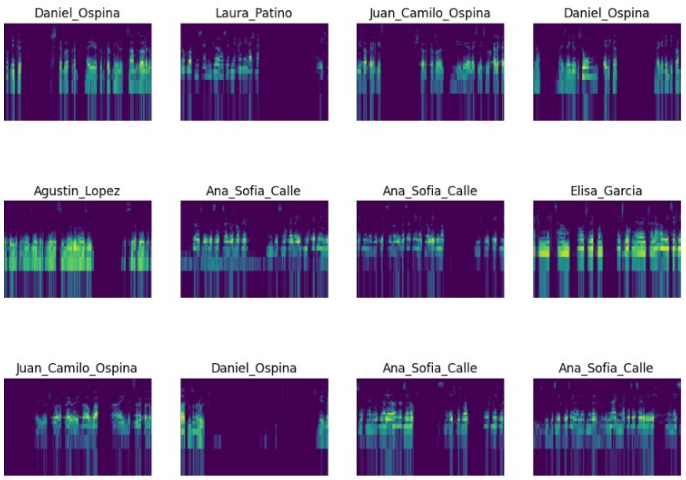

Fig. 16. Batch de espectrogramas de la fht del Dataset

Se logra evidenciar como entre los espectrogramas es posible diferenciar entre las magnitudes relativas de las frecuencias, denotando diferencias como las mencionadas en el ánalisis de los espectrogramas anteriores (especialmente el realizado con la transformada de Fourier).
Ahora se muestran los resultados del entrenamiento. Para este fin se ponen las matrices de confusión que presentan las categorías (en este caso personas) y presenta cuantas veces la red hizo bien la predicción o si se confundió. La primera matriz de confusión mostrada es la de las pruebas de validación (ver figura 17), la de arriba siendo la del modelo con la implementación de la STFT y la segunda con la FHT.

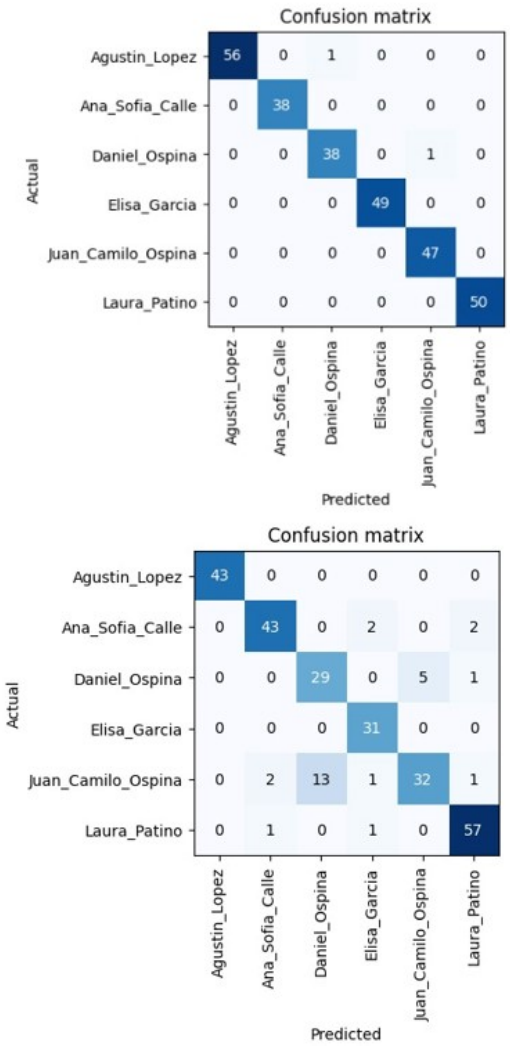

Fig. 17. Matrices de confusión de pruebas de validación

Es posible ver que el modelo implementado con la stft presentó una menor confusión respecto a los datos con los que el modelo es entrenado. Presentando una confusión entre Juan Camilo y Daniel dado a que ellos son hermanos y tienen una voz muy parecida, esto se lograba evidenciar cualitativamente en los espectrogramas que fueron generados. También hubo una confusión entre Daniel y Agustín dado a que ambos tienen una voz grave y el algoritmo confundió en una instancia los espectrogramas correspondientes. Ahora bien, para el modelo entrenado con los espectrogramas de la transformada rápida de Fourier, se evidencia como el algoritmo confundió muchas más veces las voces de Daniel y Juan Camilo, esto por lo que se mencionó en la sección de los espectrogramas, donde la forma de la distribución de frecuencias a través del tiempo no cambiaba significativamente de aspecto. Se nota como con Agustín la red no se confunde una sola vez, esto se dá porque su voz es grave y dada la naturaleza de la transformada este aspecto se distingue de los espectrogramas, no en términos de la distribución de la frecuencia a través del tiempo sino en términos de la intensidad de la misma concentrada en las frecuencias bajas.
Las siguientes matrices de confusión (ver figura 18) son cuando se prueba el modelo con datos que nunca había procesado, es decir, con el segundo set de datos que no hacen parte del entrenamiento.

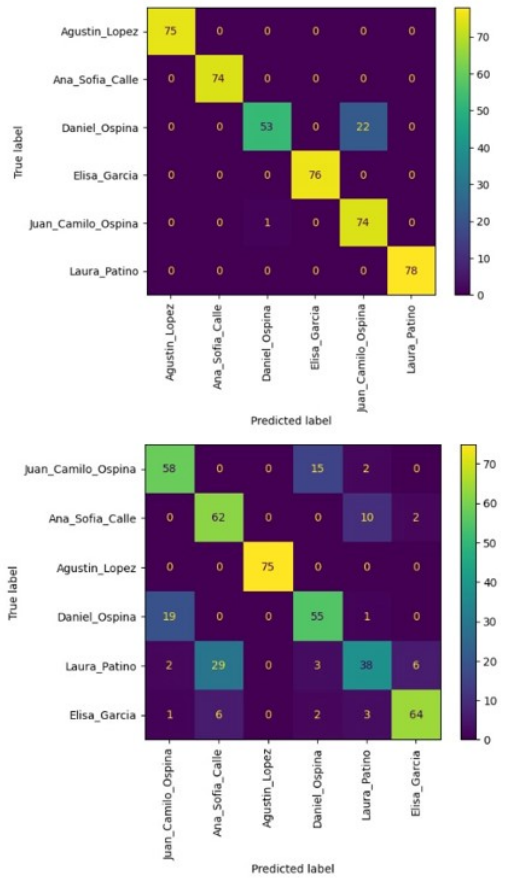

Fig. 18. Matrices de confusión con datos nuevos

Estas matrices de confusión muestran como el modelo implementado con stft presenta una menor confusión respecto a la implementación de la fht. Con respecto a la primera, se logra ver como se sigue confundiendo el algoritmo con Daniel y Juan Camilo por las mismas razones ya mencionadas. Ahora, con respecto a la implementación con la fht, se muestra como con Agustín obtuvo una precisión del 100% como en la parte del entrenamiento de validación. Aquí se vuelve a evidenciar como se confunde el algoritmo con Daniel y Juan Camilo. Para este caso, aparece una confusión del modelo aún mayor con Laura y Ana, dado a que los tonos de su voz son muy parecidos, demostrando esto que la implementación por medio de la fht es buena diferenciando entre los tonos de la voz, como se mencionó con anterioridad. Pero cuando se trata de analizar otros aspectos de la forma de onda, como las oscilaciones entre las intensidades por la "armonía" de la voz, falla constantemente y no logra hacer una categorización adecuada.

A continuación se muestran las tablas del proceso de entrenamiento del algoritmo. La primera, mostrada en la figura 19, es la de la implementación del stft. Cada fila corresponde a las épocas del entrenamiento, la primera columna indica el proceso de minimixación de la función de pérdida respecto al set de entrenamiento de los datos. Consecuentemente, la segunda columna indica ls función de pérdida set de validación, es decir, con respecto al 20% de los datos que se separó para verificar que

el modelo generalice correctamente. Podemos ver que ambas decrecen consistentemente en cada época indicando un proceso exitoso de optimización. Finalmente, la métrica que me permite evaluar el proceso de aprendizaje, es el porcentaje del error rate, que, al final del entrenamiento indica un porcentaje del 0.7%, lo que incidica que la red es capaz de predecir el locutor de un 99.3% de los datos de entrenamiento. De forma análoga, en el proceso de aprendizaje utilizando las fht 20 se alcanzo un error rate de aproximadamente un 11%, indicando que la red es capaz de reconocer un 89% de los locutores.

epoch	train_loss	valid_loss	error_rate	time
0	2.433583	1.182790	0.435714	00:10

epoch	train_loss	valid_loss	error_rate	time
0	0.727149	0.532837	0.135714	00:14
1	0.598653	0.210864	0.057143	00:13
2	0.447293	0.117972	0.021429	00:13
3	0.342888	0.091298	0.010714	00:13
4	0.273233	0.085326	0.010714	00:13
5	0.215340	0.077788	0.007143	00:13
6	0.177153	0.072724	0.007143	00:13
7	0.144629	0.074145	0.007143	00:13
8	0.122680	0.069450	0.007143	00:13
9	0.114571	0.068594	0.007143	00:14
10	0.100102	0.070611	0.007143	00:14
11	0.091522	0.071120	0.007143	00:14

Fig. 19. Proceso de entrenamiento con la stft

epoch	train_loss	valid_loss	error_rate	time
0	2.832088	1.852528	0.833333	01:54

epoch	train_loss	valid_loss	error_rate	time
0	2.014812	1.402789	0.594697	00:14
1	1.906845	1.061357	0.416667	00:13
2	1.692821	0.780444	0.306818	00:13
3	1.468800	0.598279	0.227273	00:13
4	1.284517	0.469209	0.196970	00:13
5	1.115026	0.409464	0.155303	00:13
6	0.968128	0.350011	0.128788	00:13
7	0.850107	0.323872	0.109848	00:13
8	0.755762	0.318114	0.128788	00:13
9	0.687659	0.304152	0.106061	00:14
10	0.638056	0.297816	0.106061	00:13
11	0.600260	0.299607	0.109848	00:13

Fig. 20. Proceso de entrenamiento con la fht

Como validación final, se evalúa un nuevo set de 423 datos que el algoritmo nunca "vió". Para la implementación del stft se logró una precisión del 94.92% y para del fht es del 77.7%. Es claro que se obtuvo un excelente resultado con la primera y uno decente para la transformada rápida de hankel. Todo esto validado y justificado por lo mencionado anteriormente con los ánalisis de los espectrogramas y las matrices de confusión.

5. CONCLUSIONES

- Es fundamental tener una base sólida en la teoría de las series de Fourier y Fourier-Bessel para comprender las técnicas utilizadas en el procesamiento de datos de una señal variable en el tiempo. Se logra desarrollar las relaciones de resolución en series de Fourier, las transformadas de Fourier y Fourier-Bessel así como los coeficientes para esta última; en los cálculos se presentan las funciones especiales delta de Dirac y delta de Kronecker, además de utilizar el método de separación de variables de E.D. parciales.

- Entre las principales causas de error y de difrenca de exactitus entre ambos métodos, se encuentra que el tratamiento de la señal que realiza Python de forma predeterminada con la STFT es más sofisticada al igual que la función perdeterminada de `mel_spectrogram`, mientras que el tratamiento con la FHT es menos sofisticado y requiere de un acondicionamiento adiconal de paso a simetría radial, además del paso de las frecuencias y la amplitud a la escala logrítmica (sin ser la escala de Mel) si se quiere mejorar los resultados de éste.

- Se puede obsesrvar que ambos modelos, el de FHT en mayor medida que el de STFT, tienden a hacer confusiones entre dos miembros de una familia o confusiones entre dos hombres y dos mujeres, lo que se explica debido a que los armónicos de la frecuencia y sus intensidades, entre otras características pueden ser similares entre estas categorías.

REFERENCES

1. P. K. Chaudhary, V. Gupta, and R. B. Pachori, Digit. Signal Process. **135**, 103938 (2023).
2. M. Singh, "Wavelength of sound waves," (2023).
3. V. Velardo, "Audio signal processing for machine learning," (2020).
4. J. Teuwen and N. Moriakov, *Convolutional Neural Networks* (2020), pp. 481–501.
5. I. Neutelings, "Fourier series synthesis," .
6. G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists* (Academic Press, San Diego, CA, 2005), 6th ed.
7. L. R. R. JONT B. ALLEN, PROCEEDINGS OF THE IEEE **65** (1977).
8. A. Mertins, *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications* (John Wiley Sons Ltd, 1999).
9. A. de Saint-Exupéry, *El Principito: The Little Prince* (Editorial Verbum, 2019).
10. Y.-Y. Yang, M. Hira, Z. Ni, A. Astafurov, C. Chen, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang *et al.*, "Torchaudio: Building blocks for audio and speech processing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, 2022), pp. 6982–6986.
11. J. Howard and S. Gugger, Information **11**, 108 (2020).
12. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, Adv. neural information processing systems **32** (2019).
13. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2016), pp. 770–778.
14. L. P. A. S. C. Juan Camilo Ospina, Agustin Lopez, "Reconocimiento de voz mediante deep learning utilizando la stft y la fht," https://github.com/JuanHaunted/Voice_Recognizer (2023).