Separación de hablantes en tiempo real usando técnicas de Deep Learning

Jose Alberto Arango Sánchez

Trabajo de grado presentado para optar al título de
Ingeniero de Sistemas

Asesor
Julián David Arias Londoño, Doctor (PhD)

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería de Sistemas
Medellín, Antioquia, Colombia
2022

Separación de hablantes en tiempo real usando técnicas de Deep Learning

Jose Alberto Arango Sánchez

Departamento de ingeniería de sistemas
Marzo 1, 2022,
Trabajo de grado para optar por el titulo de Ingeniero de Sistemas

Resumen

La separación de hablantes (Speaker Separation/Multitalker separation), es un tarea que consiste en separar en diferentes audios, las intervenciones individuales de los hablantes involucrados a partir de una mezcla auditiva. Tarea que permitiría mejorar la interacción entre humanos y sistemas, a través del habla, ya que serviría como filtro de información.

Durante este trabajo de grado, exploramos el comportamiento de 3 aproximaciones del estado del arte (DPRNN [6] , SepFormer[13], Conv-TasNet[7]), usando un corpus de grabaciones de llamadas sobre canal telefónico en el idioma español[9], con hablantes de diferentes partes de América latina. Se seleccionó Conv-TasNet como la arquitectura base por su desempeño, ya que logró una relación señal distorsión invariante en la escala (SI-SDR) de 6.9 dB, luego realizamos múltiples experimentos con esta arquitectura, con el objetivo de obtener mejores resultados, consiguiendo así un modelo con un SI-SDR de 9.9 dB. Luego de manera experimental, se identifico una relación entre la similitud entre hablantes y el desempeño del modelo, por lo tanto se planteó una mejora a la arquitectura ConvTasNet, introduciendo un término en la función de costo de la arquitectura original. Dicho término esta relacionado con la similitud entre hablantes y utiliza un Speech embedding para el cálculo de dicha similitud. Con esta mejora se logró un SI-SDR de 10.6 dB. Finalmente el modelo ConvTasNet modificado, se desplegó en una infraestructura que permitió su ejecución en tiempo real, sin embargo para garantizar el concepto de tiempo real, utilizamos segmentos de audio de 1 segundo, tiempo en el cual, por lo general solo 1 hablante interviene, lo cual es una condición distante de la realidad conocida por el modelo entrenado (longitud de las muestras de entrenamiento y validación).

Asesor interno: Julián David Arias Londoño
Profesor Titular Universidad de Antioquia

Agradecimientos

Sin duda alguna, es un gran logro para mí llegar hasta este punto; agradezco enormemente a la Universidad de Antioquia por cada uno de los espacios y momentos que me brindó durante esta experiencia. Trabajar en el grupo de investigación In2Lab fue una experiencia enriquecedora. Participar de múltiples competencias y eventos, donde sin duda alguna, demostramos el potencial de un Ingeniero de Sistemas de la UdeA, fue lo que me permitió darme a conocer al mundo profesional. Por estas y muchas cosas más, gracias y mil gracias UdeA.

De la misma forma, agradezco a mi familia y compañera de vida, por todo el apoyo brindado durante estos años de estudio. Siempre han estado en todo este proceso y han sido mi combustible durante este largo camino.

Agradezco enormemente a los profesores Julián David Arias Londoño y Raúl Ramos Pollan, por ser esa luz que iluminó mi camino y me permitió determinar la clase de profesional que quería ser.

Finalmente agradezco a la Asociación Universitaria de Antioquia (AUDEA), por brindarme un espacio para vivir, madurar y aprender a convivir en sociedad.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

El habla es una forma natural de interacción entre humanos y sistemas (ordenadores, celulares, asistentes de voz, etc). Sin embargo, el habla de interés muchas veces es opacada por varios sonidos del entorno real, como lo son: ruido de fondo, otros hablantes que interfieren, la reverberación del entorno, etc., lo cual dificulta la interacción entre humanos y sistemas. Para casos en los que el audio a procesar corresponde a la mezcla de diferentes hablantes, se han propuesto sistemas de separación de hablantes (en inglés *speech separation*), útiles para aislar las intervenciones individuales de los hablantes a partir de una mezcla auditiva de múltiples hablantes [10], tarea que para los seres humanos y en especial para el sistema auditivo es de mucha facilidad, sin embargo, se ha demostrado que es muy difícil construir un sistema automático que iguale o supere al sistema auditivo humano en esta tarea [15].

Con una amplia gama de aplicaciones, en diversos campos como la medicina con las prótesis auditivas, diseño de audífonos comerciales, las telecomunicaciones móviles y el reconocimiento automático y robusto del habla y los hablantes [15], la separación de hablantes es un campo de interés y en continuo desarrollo. Sin embargo, como en la gran mayoría de tareas relacionadas con el procesamiento de voz y habla, la mayoría de las aproximaciones existentes fueron desarrolladas y entrenadas con conjuntos de datos en el idioma inglés y bajo condiciones controladas en relación con los niveles y tipos de ruidos, por lo tanto durante en este trabajo de grado exploramos el comportamiento de algunos modelos con bases de datos de grabaciones de llamadas sobre

canal telefónico en el idioma español, con hablantes de diferentes partes de América latina. Contando así con una rica diversidad de acentos. Esto plantea un reto ya que las grabaciones sobre canal telefónico están limitadas a un filtro paso bajo de 3KHz.

Este problema ha sido abordado tradicionalmente por el procesamiento de señales, sin embargo con la aparición del aprendizaje profundo (Deep learning), se planteó como un problema supervisado, en el cual se tiene la grabación de la llamada en donde intervienen los dos hablantes (mezcla) y las intervenciones individuales de los mismos (fuentes originales), para un total de 3 audios (mezcla, fuentes 1 y 2), todos de la misma duración, los cuales son utilizados para entrenar arquitecturas específicas diseñadas para generar las fuentes individuales lo más parecidas posibles a las fuentes originales, a partir de la mezcla. Las medidas más usadas como funciones de costo y métricas de evaluación son la relación señal distorsión invariante de la escala (SI-SDR) o la relación señal ruido invariante de la escala (SI-SNR), como métricas que miden el desempeño de la separación en decibeles (dB) y en donde un mayor número indica un mayor nivel de separación. Todas las aproximación actuales tienen problemas con el seguimiento a largo plazo de un hablante individual, por lo tanto estos modelos tienden a fallar especialmente cuando hay una pausa larga en el audio de la mezcla, reto que junto al corpus de llamadas telefónicas, será abordado durante este trabajo.

Finalmente este trabajo se encuentra dividido en 4 fases: investigación, experimentación, despliegue y evaluación. Durante la investigación se realizará una búsqueda de las aproximaciones(modelos y/o arquitecturas) más relevantes en el estado del arte. Luego se comprenderá a detalle sus componentes y funcionamientos, además de evaluar su desempeño con modelos pre-entrenados. Posteriormente se seleccionará una aproximación, la cual pasará por la fase de experimentación y en la que se entrenará con diversas configuraciones y fuentes de datos, para intentar mejorar el desempeño del modelo actual, en caso de ser necesario. Luego de consolidar el mejor modelo, este se desplegará en una infraestructura que permita su ejecución en tiempo real, para posteriormente evaluar su desempeño y determinar los escenarios bajo los cuales se obtiene un desempeño óptimo.

Capítulo 2

Estado del arte

Tradicionalmente, la separación de hablantes (SH) ha sido enfrentado desde el procesamiento de señales, sin embargo, durante la ultima década, con la introducción del aprendizaje profundo (Deep Learning), esta tarea ha sido abordada como un problema de aprendizaje supervisado, en donde patrones de discriminación del habla, hablantes y ruido se aprenden durante el entrenamiento, surgiendo así aproximación (algoritmos, modelos y arquitecturas), que han acelerado drásticamente el progreso y rendimiento de la SH [15]. Estas aproximaciones, han permitido el desarrollo de un amplia gama de aplicaciones, en diversos campos como la medicina, con las prótesis auditivas, la telecomunicación móvil y el reconocimiento automático y robusto del habla y los hablantes [15].

Los métodos de separación de hablantes, se clasifican dependiendo del número de sensores o micrófonos. Existiendo así, métodos de separación monoaurales (Single-microphone) y basados en matrices (Multi-microphone). Durante este trabajo, abordaremos métodos monoaurales, esto debido a la naturaleza del corpus que se empleara (grabaciones de llamadas telefónicas). La mayoría de los métodos desarrollados, han formulado el problema de la separación, a través de una representación tiempo-frecuencia (T-F) o espectrograma de la señal mixta, la cual se estima a partir de la forma de onda, utilizando la transformada de Fourier de corta duración (STFT). Estos métodos tienen como objetivo, estimar un espectrograma limpio de las fuentes individuales, a partir del espectrograma de la mezcla. Este proceso se puede realizar

mediante dos enfoques [7]:

- Método directo: el cual realiza la estimación directa de la representación del espectrograma de cada fuente de la mezcla. Utilizando para esto, técnicas de regresión no lineal, donde los espectrogramas de las fuentes limpias se utilizan como objetivo de entrenamiento.

- Estimación de máscara: estima una función de ponderación (máscara), para cada una de las fuentes, la cual al ser multiplicada por cada bin Tiempo Frecuencia (T-F) del espectrograma de la mezcla, recupera las fuentes individuales limpias.

Ambos enfoques calculan la forma de onda de cada fuente, utilizando la transformada de Fourier inversa de tiempo corto (iSTFT), a partir del espectrograma de magnitud estimado de cada fuente, junto con la fase original. Siendo el enfoque de estimación de máscara, el más común en los últimos años y en el cual, el aprendizaje profundo ha logrado los mayores incrementos a nivel de métricas de separación. Sin embargo, este enfoque tiene varias deficiencias, entre las que se destacan [7]:

1. La STFT, es un transformación de señal genérica, la cual no es necesariamente óptima para la separación del habla.

2. La representación tiempo-frecuencia necesita una descomposición de frecuencias de alta resolución de la señal de la mezcla, lo cual requiere una ventana temporal larga, que garantice el éxito de la separación. Por lo tanto, se aumenta la latencia y el costo computacional, limitando su uso en aplicaciones de tiempo real.

3. La disociación entre la magnitud y la fase genera un reto al momento de realizar la reconstrucción de la forma de onda.

Dada estas dificultades, nace una red de separación de audio en el dominio del tiempo totalmente convolucional (Conv-TasNet), con un enfoque End-to-End, la cual utiliza un codificador lineal para generar una representación de la forma de onda del habla optimizada, para luego ser usada en la separación de hablantes individuales.

El codificador lineal del Conv-TasNet sustituye la STFT, la cual es utilizada para la extracción de características, por una representación basada en datos, que se optimiza conjuntamente con un paradigma de entrenamiento End-to-End.

La separación de los hablantes se logra, aplicando un conjunto de funciones de ponderación (máscaras), a la salida del codificador. Luego las representaciones modificadas del codificador, se convierten en formas de onda mediante un decodificador lineal, el cual juega el papel de la iSTFT.

El modelo Conv-TasNet genera las máscaras, utilizando una red convolucional temporal (TCN). Dicha red esta conformada por bloques convolucionales dilatados 1-D apilados, lo cual le permite modelar dependencias a largo plazo de la señal del habla, manteniendo un tamaño de modelo pequeño. Este modelo fue evaluado, utilizando métricas objetivas de distorsión como: la relación señal-ruido invariante de la escala (SI-SNRi) y la relación señal distorsión (SDRi). También a través de medidas subjetivas de la calidad del audio como: evaluación perceptiva de la calidad subjetiva (PESQ) y la puntuación media de opinión (MOS).

Para la separación de hablantes, el modelo Conv-TasNet empleo los conjuntos de datos WSJ0-2mix (dos hablantes) y WSJ03mix (3 hablantes). Estos conjuntos de datos son utilizados comúnmente por la mayoría de las aproximaciones presentes en el estado del arte de la separación de hablantes. Estos son generados mediante la mezclas de audios en ingles, seleccionados aleatoriamente de diferentes hablantes extraídos del conjunto de datos de Wall Stree Journal (WSJ0) [4], los cuales, posteriormente, son mezclados con ruido aleatorio entre -5 dB y 5dB. El desempeño de esta aproximación con el conjunto de datos WSJ0-2mix, fue de 15.3 dB para SI-SNRi, 15.6 dB en SDRi, MOS de 4.03 y un PESQ de 3.22. Finalmente, al tener un tamaño significativamente pequeño (5.1 Millones de parámetros), lo convierte en una solución adecuada para aplicaciones de separación de voz tanto offline como en tiempo real. Sin embargo, dado que Conv-TasNet utiliza una longitud de contexto temporal fija, el seguimiento a largo plazo de un hablante individual puede fallar, en especial cuando se presentan pausas de larga duración en el audio a separar (mezcla) [7].

Los sistemas de separación de hablantes basados en el dominio del tiempo, como Conv-TasNet, reciben secuencias de entrada con un gran número de pasos en el tiempo, lo cual introduce desafíos al momento de modelar estas secuencias extremadamente largas. Si se utilizarán redes neuronales recurrentes (RNN) convencionales,

se presentarían dificultades en su optimización. Por otro lado, si se utilizan redes convolucionales unidimensionales (CNN 1-D), no se podría realizar un modelado de secuencias a nivel de enunciado, cuando el campo receptivo es menor que la longitud de la secuencia[6]. Por lo tanto surge como solución a estas dificultades, una red neuronal recurrente de doble recorrido (DPRNN), el cual se denomina así mismo como un método sencillo, que organiza las capas de una RNN en una estructura profunda que permite modelar secuencias extremadamente largas.

La aproximación DPRNN, divide la secuencia de entrada en trozos más cortos e intercala dos RNNs, una RNN intra-chunk y una RNN inter-chunk, las cuales realizan un modelado local y uno global respectivamente. En un bloque DPRNN, la RNN intra-chunk, procesa los trozos locales de forma independiente y luego la RNN inter-chunk, agrega información de todos los trozos, logrando así un procesamiento a nivel de enunciado. Finalmente, este modelo utiliza la misma base de datos de la aproximación anterior (WSJ0-2mix), obteniendo 18.8 dB en la métrica de SI-SNRi y 19.0 dB en SDRi [6]. Sin embargo, las RNN son modelos inherentemente secuenciales, por lo tanto, no es posible la paralelización de sus cálculos y tienden a generar un cuello de botella cuando se procesan grandes conjuntos de datos con secuencias largas. Aquí entran en acción los transformers, una alternativa natural a las RNN estándar, sustituyendo los cálculos recurrentes por mecanismos multi-head. Naciendo así SepFormer, una red neuronal basada en transformers, que aprende dependencias a corto y largo plazo, con un enfoque multi-escala para la separación de hablantes [13].

La red SepFormer, al igual que las dos anteriores aproximaciones, trabaja bajo el dominio del tiempo, adopta el framework propuesto en DPRNN. Sustituyendo las RNN por un pipeline multi-scale, compuesto por transformers. Esta aproximación alcanza el mayor rendimiento entre las dos aproximaciones anteriores, usando el mismo conjunto de datos (WSJ0-2mix). Obteniendo así 22,3 dB en SI-SNRi. A pesar de contar con 26 millones de parámetros, consigue un rendimiento competitivo, siendo más rápido y exigiendo menos memoria que los métodos anteriores. Todo esto gracias a la introducción de los transformers en su arquitectura.

Finalmente, todas las aproximaciones encontradas en el estado del arte, abordan el problema de la separación de hablantes como un problema de aprendizaje supervisado, usando métodos de separación monoaurales (Single-microphone). Cada uno de las propuestas encontradas, trata de solucionar una deficiencia presente en el mismo estado del arte: Conv-TasNet sustituye la STFT, la cual es utilizada para la extracción de características, por una representación basada en datos, la DPRNN introduce las RNN ya que las (CNN 1-D) del Conv-TasNet, no realizan un modelado de secuencias a nivel de enunciado, por otro lado el SepFormer al utilizar transformers permite la paralelización de cálculos y evita cuellos de botella generados por las RNN.

Capítulo 3

Objetivos

3.1. Objetivo general

Realizar un análisis comparativo de diferentes arquitecturas para la separación de hablantes en el idioma español, a partir de métricas de calidad de la señal generada y latencia, que permita seleccionar la mejor arquitectura para realizar un despliegue en tiempo real.

3.2. Objetivo específico

1. Realizar una revisión y evaluación en términos de tamaño de los modelos y costo computacional, de aproximaciones para la separación de hablantes disponibles en el estado del arte, principalmente enfocadas en técnicas de Deep Learning.

2. Identificar y recolectar una base de datos de audios de llamadas en el idioma español, que pueda ser usada para el entrenamiento y la validación de las arquitecturas seleccionadas.

3. Realizar pruebas de entrenamiento y validación de las arquitecturas seleccionadas, usando los corpus en español, que permita identificar la aproximación con mejor relación señal distorsión invariante a la escala.

4. Diseñar un arquitectura de software para el despliegue del mejor modelo encontrado, en un infraestructura que permita su ejecución en tiempo real.

Capítulo 4

Marco Teórico

Antes de entrar en materia sobre la separación de hablantes, debemos aclarar una cantidad de conceptos presentes en el campo del procesamiento del audio: las diferentes formas de representación, escalas, etc. Luego definiremos algunos conceptos propios del dominio de la separación de hablantes; las métricas usadas en la evaluación de estos sistemas, funciones de costo y formas de entrenamiento. Finalmente, hablaremos sobre la similitud entre hablantes, ya que es un aspecto que experimentalmente demostró tener una gran influencia en el proceso de separación de fuentes, para los diferentes modelos y/o arquitecturas evaluadas, siendo este el elemento que permitió introducir una mejora al modelo ConvTasNet.

4.1. Representación del audio

La representación fundamental del audio es en forma de onda. Algunos enfoques de separación de hablantes operan directamente con esta, aunque otros, realizan algún tipo de procesamiento previo. En esta sección, hablaremos sobre los diferentes tipos de representaciones de entrada y salida, que se emplean comúnmente en las arquitecturas de separación de hablantes.

4.1.1. Representaciones en el tiempo

Forma de onda

La forma de onda, es la representación fundamental del audio, siendo esta, la abreviatura de una señal de audio digitalizada, es la representación más parecida del sonido físicamente. De manera general y omitiendo varios detalles en el ámbito de la física, la acústica y el procesamiento de señales, aquello que denominamos forma de onda, inicia con la presión del aire que cambia con el tiempo y es registrada por un micrófono, que convierte los cambios en la presión del aire en una señal eléctrica. El voltaje de esta señal, se muestrea en un intervalo de tiempo regular, se cuantifica y se convierte en una matriz digital en una computadora. Esta matriz digital es lo que llamamos forma de onda [10]. Como se muestra en la figura 4-1, esta es una señal en el dominio del tiempo, donde se muestra la intensidad (amplitud), de una onda a través del tiempo, pero no como es el comportamiento de la frecuencias. En este caso en concreto la amplitud igual a cero indica silencio.

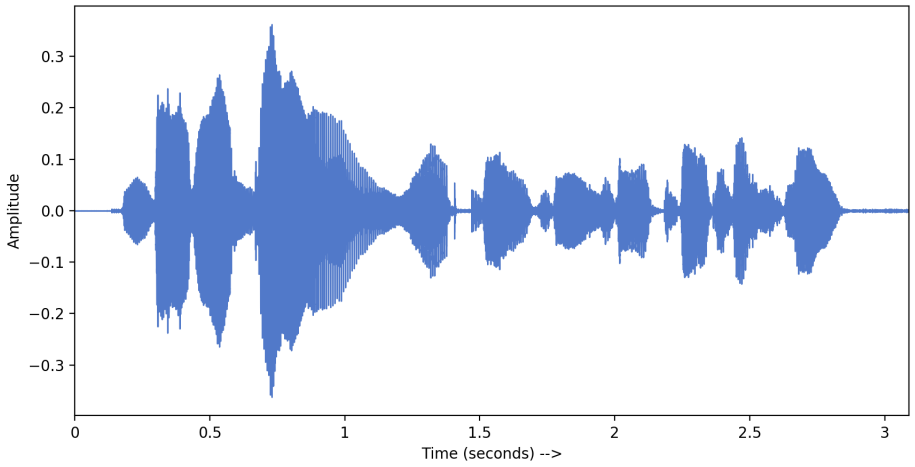


Figura 4-1: Visualización de una señal en el dominio del tiempo

De manera general, las señales se pueden clasificar en: monofónica y estereofónicas. Decimos que una señal es monofónica o mono, si esta señal ha sido grabada con un solo micrófono o canal de audio, por lo tanto la forma de su matriz es: $x \in \mathbb{R}^{t \times 1}$, mientras que una señal es estereofónica, o estéreo, si la matriz tiene dos canales, es decir, la matriz tiene la forma : $x \in \mathbb{R}^{t \times 2}$.

Uno de los principales aspectos de la forma de onda, es la frecuencia de muestreo (sample rate), la cual describe, cuántas mediciones o muestras se generan por segundo y se mide en hercios (Hertz) o Hz. Es importantes resaltar que para un señal con una frecuencia de muestreo sr, la frecuencia máxima que se puede representar de manera confiable es: $f_N = \frac{sr}{2}$, la cual se llama la frecuencia de Nyquist. Por ejemplo, si una señal tiene una frecuencia de muestreo de 44,1 kHz, la frecuencia más alta posible es 22,05 kHz.

Es de vital importancia conocer el valor de la frecuencia de muestreo, ya que hay diversas aproximaciones de separación de hablantes, que reducen la frecuencia de muestreo de la señales de entrada, realizando un proceso de submuestreo (down-sampling), para reducir la carga computacional durante el proceso de entrenamiento. Dicha reducción, elimina la información de alta frecuencia de una señal, lo que se considera un mal necesario cuando se crean prototipos de modelos basados en audio [10]. Todos los enfoques de separación de hablantes, suponen una frecuencia de muestreo igual para los conjuntos de datos de entrenamientos, validación y prueba, si esta suposición no se cumple, no se garantiza el funcionamiento óptimo del modelo.

Dado que las amplitudes no proveen mucha información, ya que sólo hablan de la intensidad de la grabación de audio, es necesario transformar la representación del audio al dominio de la frecuencia. Una representación en el dominio de la frecuencia, nos permitirá conocer las diferentes frecuencias presentes en una señal. Dicha representación es tradicionalmente empleada en tareas relacionadas con: reconocimiento de voz, clasificación de audio, separación y segmentación de audio, clasificación de género musical, reconocimiento de voz, etc.

4.1.2. Representaciones en la frecuencia

Las representaciones tiempo-frecuencia también son muy utilizadas en la separación de hablantes, las cuales consta de una matriz bidimensional, la cual representa el contenido de frecuencia de una señal de audio a lo largo del tiempo [12].

Cómo se logra apreciar en la figura 4-2, llamamos bin TF (tiempo frecuencia), a una entrada específica en esta matriz (franja horizontal). Comúnmente estas repre-

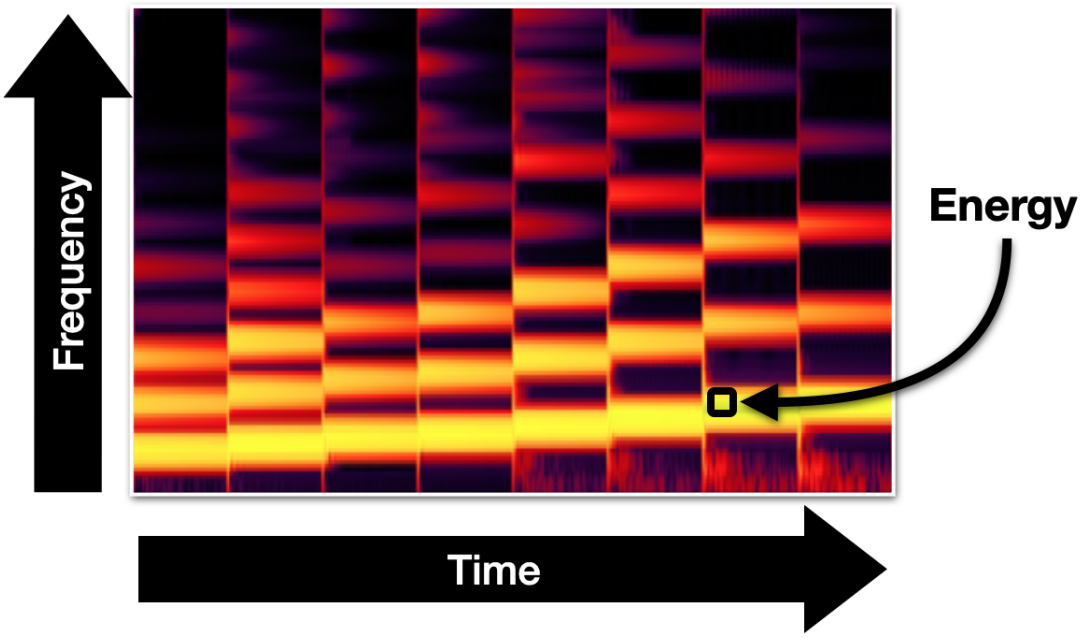


Figura 4-2: Visualización de una señal en el dominio tiempo-frecuencia.

sentaciones se visualizan usando un mapa de calor, que tiene el tiempo a lo largo del eje x y la frecuencia a lo largo del eje y. Cada intervalo de TF en el mapa de calor representa la amplitud de la señal en ese momento y frecuencia en particular. La gran mayoría de los mapas de calor, contiene un barra de colores junto a estos, la cual indica la escala de valores de amplitud y sus respectivos colores, generalmente se asume que colores más brillantes representan amplitudes más altas que los colores más oscuros [10].

A continuación, describiremos algunas de las representaciones de frecuencia de tiempo más populares. Sin embargo, cabe destacar que muchas de las representaciones de tiempo-frecuencia, emplean la transformada de Fourier de tiempo corto o STFT, por lo tanto hablaremos un poco de esta antes.

La transformada de Fourier es una operación matemática que permite convertir una señal continua en el dominio del tiempo, al dominio de la frecuencia. Por otro lado, una STFT se calcula a partir de una representación de forma de onda, calculando una transformada discreta de Fourier (DFT), de una pequeña ventana móvil, a lo largo de la duración de la ventana [10]. Este proceso se describe en la figura 4-3.

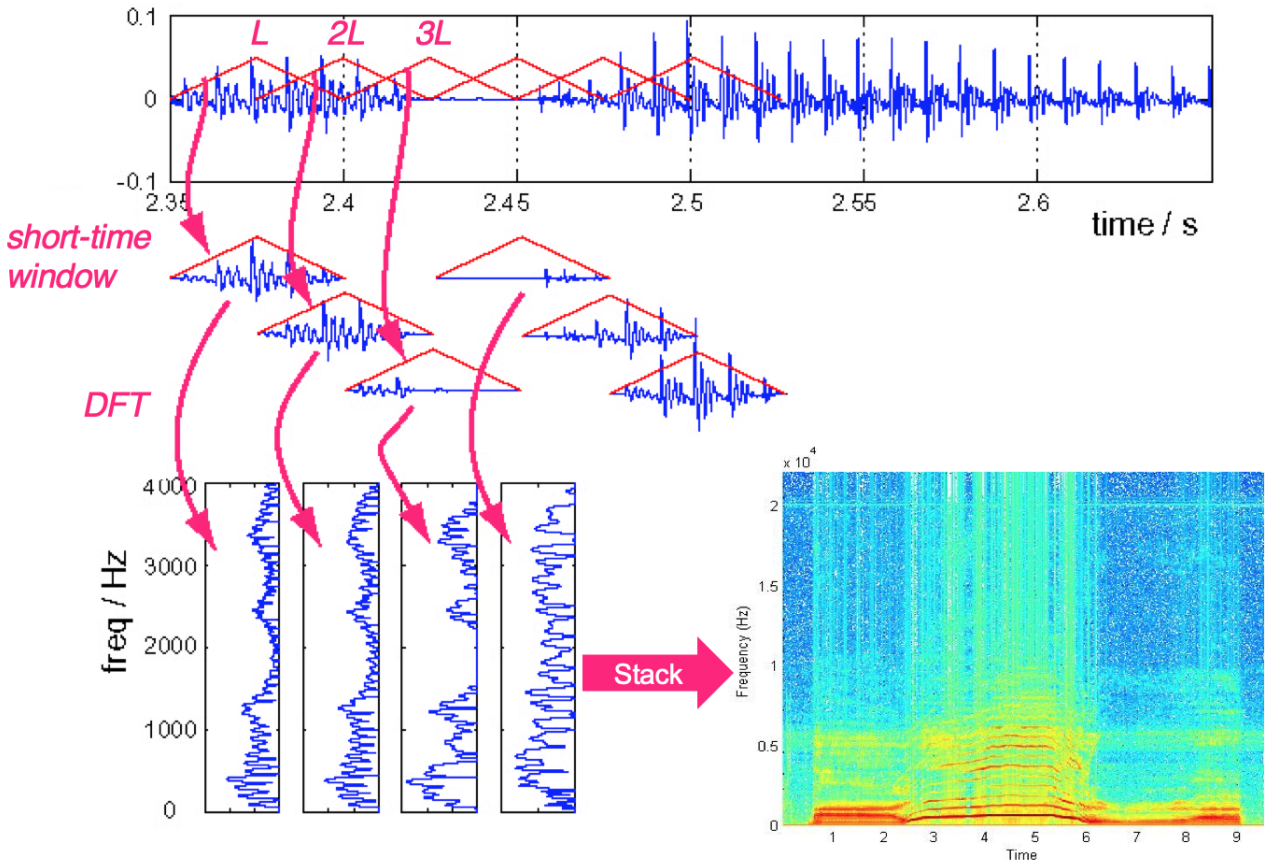


Figura 4-3: proceso de calcular una transformada de Fourier de corta duración de una forma de onda

La ubicación de cada entrada en una STFT determina su tiempo (eje x) y su frecuencia (eje y). La magnitud de un bin TF $|X(t, f)|$ en el tiempo t y la frecuencia f, determina la cantidad de energía que se escucha de la frecuencia f en el tiempo t. Es importante destacar que cada bin en nuestro STFT es complejo, lo que significa que cada entrada contiene un componente de magnitud y un componente de fase. Ambos componentes, son necesarios para convertir una matriz STFT de nuevo a una forma de onda, y así se pueda escuchar [10].

El STFT es invertible, lo que significa que un STFT de valor complejo se puede convertir de nuevo a una forma de onda. Esto se denomina, Transformada de Fourier inversa de corta duración o iSTFT. Esta característica es fundamental durante el proceso de reconstrucción de la forma de onda, luego de la separación de hablantes.

El tipo de ventana es uno de los parámetros importantes a tener en cuenta al

calcular un STFT, para tareas de separación de hablantes, se emplea comúnmente ventanas de tipo hanning y sqrt hann. Otro parámetro importante, es la longitud de la ventana, la cual determina cuántas muestras se incluyen en cada ventana de tiempo corto, este parámetro determina la resolución del eje de frecuencia de la STFT. Cuanto más larga sea la ventana, mayor será la resolución de frecuencia y viceversa. Finalmente, la longitud de salto (Hop Length), determina la distancia expresada en número de muestras, entre ventanas de tiempo corto adyacentes; cuanto menor es la longitud del salto, más veces se representa en el STFT un segmento particular de la señal de audio. Generalmente, la configuración estándar que se usa para estos parámetros, es una longitud de salto equivalente a la mitad de la longitud de la ventana [10].

Una vez aclarado el concepto de la transformada de Fourier, exploraremos las representaciones de tiempo-frecuencia, las cuales se visualizan a través de un espectrograma.

Espectrograma de magnitud y potencia

Sea X un valor de la STFT, $X \in \mathbb{C}^{T \times F}$. Para obtener el espectrograma de magnitud, se calcula el magnitud de cada elemento en el STFT, es decir $|X| \in \mathbb{R}^{T \times F}$. Por otro lado para calcular el espectrograma de potencia, se eleva al cuadrado cada elemento de la STFT $|X|^2 \in \mathbb{R}^{T \times F}$.

En la figura 4-4, se puede apreciar el espectrograma de magnitud (izquierda) y el de potencia (derecha), en el eje x tenemos el tiempo y en el eje y las frecuencias.

Espectrograma logarítmico y logarítmico de potencia

Dado que la audición humana trabaja en escala logarítmica con respecto a la amplitud [10], esta escala resulta importante al momento de trabajar con sistemas de audio en general. Por lo tanto, utilizaremos dicha escala para el cálculo de los espectros.

Sea X un valor de la STFT, $X \in \mathbb{C}^{T \times F}$. Para obtener el espectrograma logarítmico se calcula el logaritmo del valor absoluto de cada elemento de la STFT, es decir:

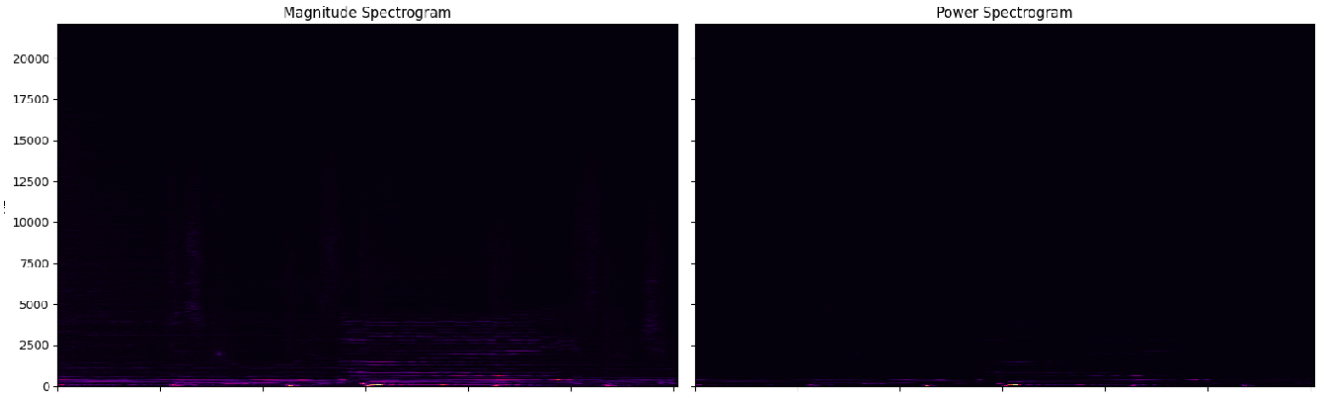


Figura 4-4: Espectrograma de magnitud Vs potencia

$\log |X| \in \mathbb{R}^{T \times F}$. Por otro lado, para obtener el espectrograma de potencia logarítmica, se calcula el logaritmo del cuadrado de cada elemento de la STFT, de la siguiente manera: $\log |X|^2 \in \mathbb{R}^{T \times F}$.

En la figura 4-5, se logra apreciar el espectrograma logarítmico (izquierda) y espectrograma de potencia logarítmica (derecha), en el eje x tenemos el tiempo y en el eje y las frecuencias.

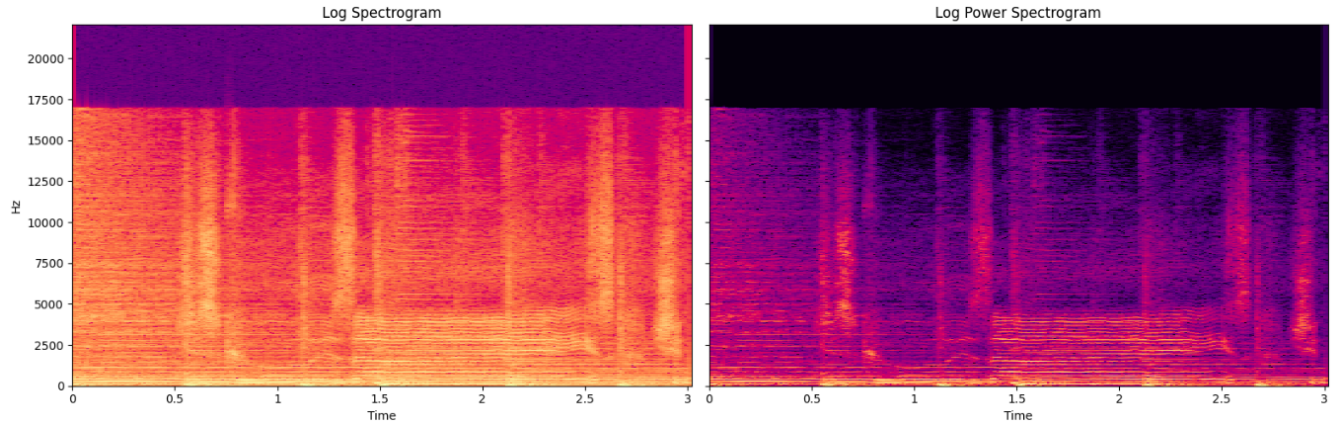


Figura 4-5: Espectrograma de logarítmico Vs espectrograma logarítmico de potencia.

Espectrograma escala Mel

Debido a que la audición humana, también es logarítmica con respecto a las frecuencias y dado que la escala Mel se aproxima a dicha propiedad, de esta forma se tiene una eje de frecuencias aproximadamente logarítmico, lo cual permite reducir la

carga computacional [10].

En la figura 4-6, se muestra un espectrograma en la escala Mel, es claro que en el eje y (frecuencias), la escala es casi logarítmica.

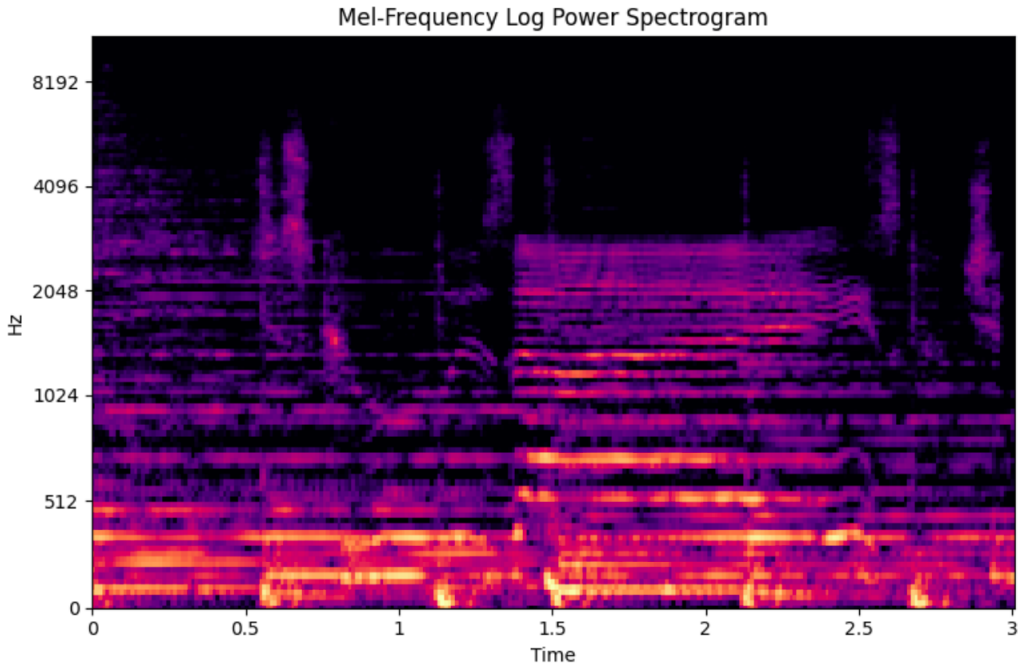


Figura 4-6: Espectrograma de Mel

4.2. Separación de hablantes

El objetivo de la separación de hablantes (Speaker Separation/Multitalker separation) es generar diversos audios, donde solo se encuentren las intervenciones individuales de los múltiples hablantes, a partir de un audio mixto (mezcla). Los hablantes pueden hablar en cualquier orden e incluso hablar al tiempo, por lo tanto, la duración de los audios generados para cada hablante, debe ser igual a la del audio mixto. En otras palabras, la separación de hablantes se puede describir como el proceso de aislar las intervenciones individuales de los hablantes, a partir de una mezcla auditiva de múltiples hablantes. Expresado matemáticamente sería:

sea $y(t)$ una combinación de varias fuentes de hablantes, la cual esta compuesta por N hablantes $x_n(t)$ para $n = 1...N$, tal que:

$$y(t) = \sum_{i=1}^{N} x_i(t) \tag{4.1}$$

El objetivo de la separación de hablantes es recuperar cada uno de los $x_i(t)$ a partir de $y(t)$ [10].

Algunos enfoques recientes de separación de fuentes, utilizan el aprendizaje profundo para aprender una representación directamente de la forma de onda, mientras que otros emiten máscaras, concepto que abordaremos a continuación:

4.2.1. Enmascaramiento o Masking

El enmascaramiento o como se conoce comúnmente en inglés Masking, es una herramienta que tiene muchos usos en diferentes aspectos de la informática, especialmente en el aprendizaje automático (Machine Learning), dentro del modelado del lenguaje y la visión por ordenador.

Una máscara dentro de la separación de hablantes, es una matriz que tiene el mismo tamaño que un espectrograma y contiene valores en el intervalo inclusivo $[0, 1]$. Cada valor de la máscara determina qué proporción de energía de la mezcla original aporta una fuente. Dicho en otras palabras, para un bin (entrada) tiempo frecuencia concreta, un valor de 1 en la máscara, dejará pasar todo el sonido de la mezcla y un valor de 0 anulara todo el sonido de la misma [10].

Cuando aplicamos una máscara a la mezcla original, multiplicando elemento a elemento, obtenemos la fuente limpia que genera dicha máscara, por lo tanto, si tenemos dos fuentes (hablantes), necesitaremos dos máscaras para obtener las fuentes limpias de los respectivos hablantes. Matemáticamente este proceso se define de la siguiente manera:

Sea $\hat{M}_i \in [0{,}0, 1{,}0]^{T \times F}$ la máscara de la i-th fuente y $|Y| \in \mathbb{R}^{T \times F}$ una mezcla representada por un espectrograma de magnitud. Es posible estimar la i-ésima fuente limpia, la cual denotaremos como S_i de la siguiente manera:

$$S_i = \hat{M}_i \odot |Y|. \tag{4.2}$$

Para obtener resultados óptimos de separación, es importante que la máscara estimada sea lo más precisa posible, y solo permita extraer información de un solo hablante a la vez (hablante de interés).

Una propiedad que se debe cumplir respecto a las máscaras de una mezcla de N fuentes (hablantes), es que la suma de elementos de cada máscara debe ser igual a una matriz de unos, la cual denotaremos como: $J \in [1]^{T \times F}$ y esta dado por [10]:

$$J = \sum_{i=1}^{N} \hat{M}_i. \tag{4.3}$$

Existen diferente tipos de máscaras, estas se clasifican de acuerdo a los posibles valores que pueden tomar dentro de un rango definido. A continuación, describiremos las más utilizadas en la separación de hablantes:

Máscara binaria

Este tipo de máscaras ya no son muy utilizadas en la actualidad, sin embargo, son útiles para darnos una intuición sobre cómo funcionan las máscaras en la práctica. Estas máscaras como su nombre lo indica (binarias), sólo pueden tomar dos posibles valores: 0 o 1. Dada la propiedad de que la suma de todas las máscaras de la mezcla suman una matriz de unos, supone entonces que cualquier entrada (bin) tiempo frecuencia, sólo está dominado por exactamente una fuente en la mezcla, siendo esta una restricción que genera grandes fallos, en caso de que dos o más hablantes se encuentren hablando al tiempo [10].

Soft Masks o *Ratio Masks*

Las *Soft mask*, máscaras blandas o máscaras suaves, son aquellas que pueden tomar cualquier valor dentro del intervalo inclusivo [0,1]. Al permitir un rango de valores más flexibles, se puede distribuir la energía de la mezcla a las diferentes fuentes, en otras palabras: La energía de una mezcla procedente de una entrada tiempo frecuencia, puede dividirse entre las diferentes fuentes (hablantes). Estas máscaras suelen dar mejores resultados que las máscaras binarias, ya que no es frecuente que toda la

energía de una mezcla, se asigne siempre a una misma fuente[10].

Ideal Masks

Es la óptima, permite una separación perfecta de la fuente, sin embargo, para su cálculo se requiere la fuente aislada verdadera [10]. Esta se suele usar como límite superior del rendimiento de un enfoque de separación de hablantes.

4.2.2. Reconstrucción de la onda

Un punto importante dentro de la separación de hablantes, es poder reconstruir las formas de ondas de los diferentes hablantes. Asumiendo que contamos con unas buenas máscaras y qué podemos aplicarlas a la mezcla, para obtener el espectro de magnitud de los hablantes estimado, todavía tenemos el problema de convertir estos espectros de magnitud en la forma de onda, ya que necesitamos definir una fase para los mismos, para luego poder así aplicar la iSTFT (inversa STFT).

La fase es crucial para poder describir una señal de audio, existen varios enfoques en la separación de hablantes basados en máscaras que manejan dicha fase. Una forma fácil y muy común de lidiar con la fase, es simplemente copiar la fase de la mezcla y agregarla al espectro de magnitud de las fuentes estimadas. Esta estrategia no es perfecta, pero funciona bien. Generalmente cuando se obtienen resultados no tan favorables, la fase no es la culpable [10].

Supongamos que tenemos la STFT de una mezcla, esta la denotamos como: $Y \in \mathbb{C}^{T\times F}$ y la máscara estimada $\hat{M}_i \in [0{,}0, 1{,}0]^{T\times F}$ de la i-th fuente. Podemos obtener el espectro de magnitud de la i-th fuente de la siguiente manera:

$$\hat{X}_i = \hat{M}_i \odot |Y| \tag{4.4}$$

Donde $\hat{X}_i \in \mathbb{R}^{T\times F}$ representa un espectrograma de magnitud de la i-th fuente estimada. Ahora para poder agregar la fase, simplemente copiamos la fase de la mezcla al espectrograma de magnitud estimado, es decir $\hat{X}_i$:

$$\tilde{X}_i = \hat{X}_i \odot e^{j \cdot \angle Y} \tag{4.5}$$

Donde $j = \sqrt{-1}$ y $\angle$ representa el ángulo de la STFT de valor complejo de Y y $\tilde{X}_i \in \mathbb{C}^{T \times F}$ [10]. De esta manera la i-th fuente estimada tendría componente compleja, similar a una STFT.

Finalmente si colocamos todo junto tenemos:

$$\tilde{X}_i = (\hat{M}_i \odot |Y|) \odot e^{j \cdot \angle Y}. \tag{4.6}$$

Otro camino para poder agregar el componente de fase, es estimar directamente la fase, usando algoritmos especiales como el algoritmo de Griffin-Lim [3], el cual intenta reconstruir la componente de fase de un espectrograma, calculando iterativamente una STFT y una STFT inversa. Este algoritmo suele converger entre 50 y 100 iteraciones, sin embargo puede dejar artefactos artificiales en el audio. Otro algoritmo es el Multiple Input Spectrogram Inversion (MISI), una variante del Girffin-Lim, el cual fue diseñado para la separación de múltiples fuentes[10].

Finalmente, una forma de evitar lidiar con la fase es estimar directamente la forma de la onda. Para eso, se han propuesto arquitecturas de aprendizaje profundo (Deep Learning), con un enfoque End-to-End. En estos sistemas, la entrada y salida son formas de onda directamente, siendo el modelo el que define cómo quiere representar la fase [10].

En la siguiente sección, describiremos la arquitectura base, bajo la cual trabajamos en este trabajo de grado.

4.3. Arquitectura base

En la actualidad, la mayoría de aproximaciones (arquitecturas) utilizadas para la separación de hablantes, se basan en las redes neuronales profundas. Básicamente, estas arquitecturas son entrenadas con una gran cantidad de audios, compuestos por una mezcla y las diferentes fuentes aisladas. La red durante el proceso de entrenamien-

to, debe producir una salida por cada fuente (fuente estimada), para luego comparar cada salida con la fuente aislada verdadera (fuente real). Dicha comparación, es utilizada para actualizar los pesos de la red y poder generar mejores resultados en futuras iteraciones. Este proceso es conocido como back-propagation.

Dado que es necesario contar con la mezcla y las fuentes aisladas, muchos de los sistemas de separación de hablantes, son sistemas de aprendizaje automático supervisado. Para poder obtener buenos resultados, es importante contar con una cantidad considerable de datos de entrenamiento.

A continuación describiremos en detalle una de las arquitecturas que marcó un hito en el campo de la separación de hablantes:

Conv-TasNet

Conv-Tasnet, es una red de separación de audio en el dominio del tiempo totalmente convolucional, con un enfoque End-to-End[7].

Esta arquitectura esta compuesta de 3 componentes claves, los cuales se observan en la figura 4-7. De manera global, esta arquitectura cuenta con un Encoder, el cual genera una representación de alta dimensionalidad de segmentos de la forma de onda de la mezcla. Luego, tenemos un componente de separación, el cual calcula un máscara para cada una de las fuentes objetivo. Finalmente, un Decoder reconstruye las formas de onda de las fuentes, con base en las características enmascaradas [7].

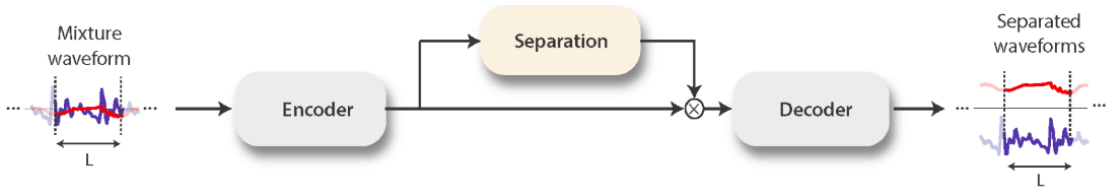


Figura 4-7: Componentes claves de la arquitectura Conv-TasNet. Tomado de [7].

1. Encoder

   Transforma segmentos cortos de la forma de onda de la mezcla en un espacio de características intermedio. Para esto, emplea un codificador lineal, sustituyendo

así el la STFT utilizada para la extracción de características, por una representación basada en datos, la cual se optimiza conjuntamente con un paradigma de entrenamiento End-to-End.

2. Separation

   Este módulo estima una función multiplicativa (máscaras), para cada fuente en cada paso de tiempo. Dichas máscaras se obtienen mediante una red convolucional temporal (TCN), la cual esta formada por bloques convolucionales dilatados apilados en 1-D, esto le permite a la red, modelar las dependencias a largo plazo de la señal de voz, manteniendo un tamaño de modelo pequeño.

   Finalmente, la separación de los hablantes se consigue aplicando un conjunto de funciones de ponderación (máscaras), a la salida del codificador.

3. Decoder

   Reconstruye la forma de la onda usando un decodificador lineal, el cual juega el papel de la iSTFT.

Como se observa en la figura 4-8, tanto el Encoder como el Decoder, está compuesto por un bloque convolución 1-D, el cual se describe en la figura 4-9. Donde cada bloque, consiste en una operación de convolución 1×1 seguida de una operación de convolución en profundidad (D-conv) o depthwise convolution, con una función de activación no lineal y una normalización añadida entre cada una de las dos operaciones de convolución. Finalmente, dos bloques lineales de $1 \times$ 1-conv, sirven como ruta residual y ruta de conexión de salto para el siguiente bloque, respectivamente [7].

Respecto al módulo de separación, el cual se logra apreciar en la figura 4-8, es el encargado de estimar las máscaras basándose en la salida del codificador. Este módulo esta compuesto por diferentes capas, las cuales a su vez contienen diversos bloques convolucionales 1-D. Cada bloque convolucional 1-D, tiene diferentes factores de dilatación, los cuales aumentan exponencialmente, esto garantiza una ventana de contexto temporal lo suficientemente grande, logrando así, aprovechar las dependencias de largo alcance de la señal del habla. Los diferentes colores de los bloques

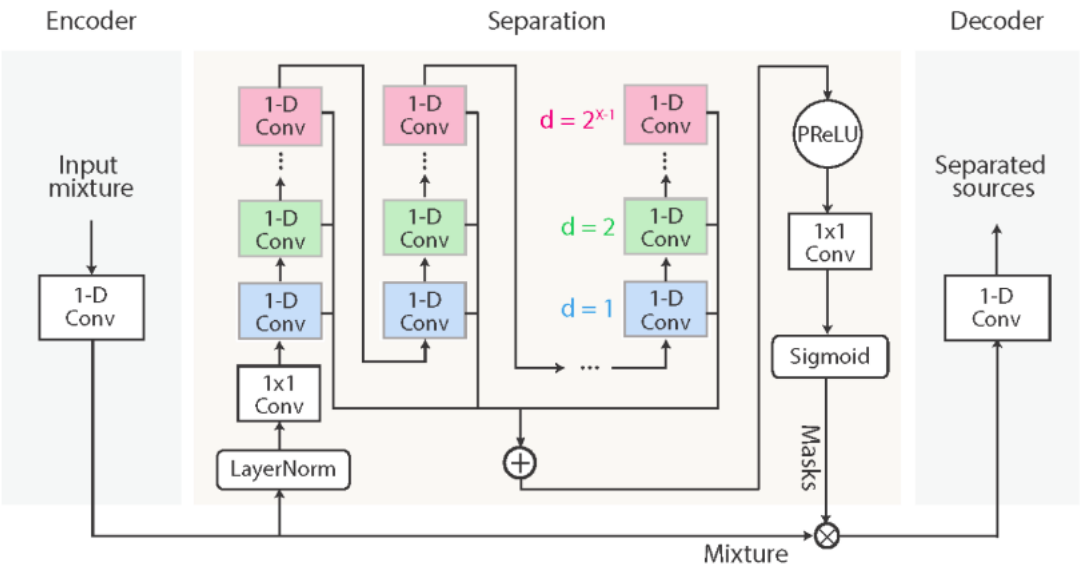


Figura 4-8: Diagrama de flujo arquitectura Conv-TasNet. Tomado de [7].

convolucionales 1-D denotan diferentes factores de dilatación[7].

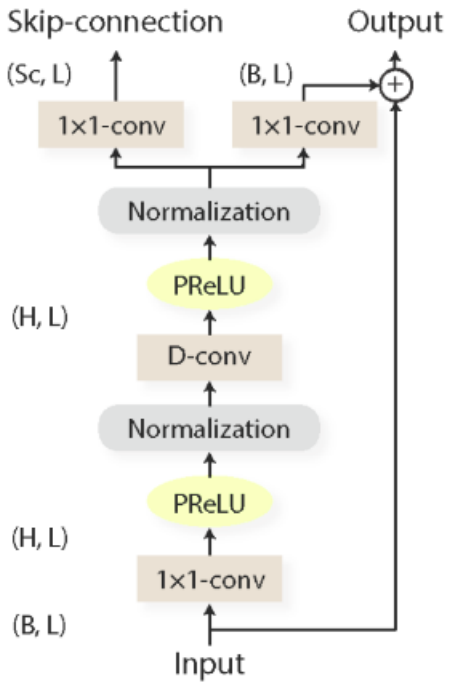


Figura 4-9: Estructura de un bloque convolucional (1-D Conv) arquitectura Conv-TasNet. Tomado de [7].

La función de costo que emplea el modelo Conv-TasNet, es la relación fuente-ruido invariable en escala (SI-SNR). El objetivo de entrenamiento de este sistema

end-to-end es maximizar dicha medida. La SI-SNR, esta definida por:

$$\text{SI-SNR} : \ = 10 \log_{10} \frac{\| \mathbf{s}_{\text{target}} \|^2}{\| \mathbf{e}_{\text{noise}} \|^2} \tag{4.7}$$

Donde $\mathbf{s}_{\text{target}}$ y $\mathbf{e}_{\text{noise}}$, estan dado por:

$$\mathbf{s}_{\text{target}} := \frac{\langle \hat{\mathbf{s}}, \mathbf{s} \rangle \mathbf{s}}{\|\mathbf{s}\|^2} \tag{4.8}$$

$$\mathbf{e}_{\text{noise}} := \hat{\mathbf{s}} - \mathbf{s}_{\text{target}} \tag{4.9}$$

Donde $\hat{\mathbf{s}} \in \mathbb{R}^{1 \times T}$ y $\mathbf{s} \in \mathbb{R}^{1 \times T}$ son la fuente estimada y la fuente verdadera, respectivamente, y $\|\mathbf{s}\|^2 = \langle \mathbf{ss} \rangle$ denota la potencia de la señal. La invariabilidad de la escala se garantiza normalizando $\hat{s}$ y s a media cero antes del cálculo[7].

Finalmente, el sistema Conv-TasNet, supera significativamente métodos de enmascaramiento basados en representaciones tiempo-frecuencia, para mezclas de dos y tres hablantes. Además, Conv-TasNet supera a varias máscaras ideales de magnitud de tiempo-frecuencia, en la separación del habla de dos hablantes, evaluada tanto por medidas objetivas de distorsión, como por la evaluación subjetiva de la calidad. Por último, Conv-TasNet tiene un tamaño de modelo significativamente pequeño (5.1 Millones de parámetros) y una latencia mínima comparado con otros modelos, lo que lo convierte en una solución adecuada para aplicaciones de separación del habla en tiempo real[7].

4.4. Métricas de rendimiento

Medir y definir una métrica es de vital importancia dentro del proceso experimental, aún más si existen múltiples hiper-parámetros que se deben variar hasta encontrar un modelo y configuración con los resultados óptimos.

En el campo de la separación de fuentes (hablantes), existen dos categorías principales en las que se clasifican las diferentes métricas: Métricas objetivas y subjetivas,

las cuales describiremos a continuación:

4.4.1. Medidas objetivas

Las medidas objetivas son aquellas que mediante un conjunto de cálculos matemáticos, compara las señales generadas por un sistema de separación con las fuentes aisladas verdaderas. Estas medidas presentan algunas dificultades, ya que existen aspectos de la percepción humana que son muy difíciles de captar sólo con medios informáticos. Sin embargo, comparado con las medidas subjetivas, estas son más rápidas y económicas de obtener[10].

A continuación describiremos las medidas objetivas más utilizadas por diferentes arquitecturas de separación de hablantes, pero antes debemos definir matemáticamente que compone a una fuente estimada:

Sea $\hat{s}_i$, la fuente estimada del i-th hablante, la cual esta compuesta por cuatro elementos[10]:

$$\hat{s}_i = s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}} \tag{4.10}$$

Dónde s_{target} es la fuente verdadera, e_{interf}, e_{noise} y e_{artif} son el error de la interferencia, ruido y artefactos agregados, respectivamente[10]. Calcular estos términos es complejo, si desea conocer más sobre cómo calcularlos exactamente puede consultar el artículo original [14].

La unidad de medida de las siguientes métricas son los decibelios (dB), los valores más altos indican mejor desempeño. Para calcular todas estas medidas, es necesario contar con la fuente aislada verdadera. Además, estas métricas se calculan sobre una señal que se ha dividido en ventanas de tiempo de corta duración, siendo así la medida final un promedio.

Source-to-Artifact Ratio (SAR)

La proporción de artefactos respecto a la fuente, mide la cantidad de artefactos no deseados que tiene una fuente estimada, en relación con la fuente verdadera [10].

Matemáticamente se define como:

$$\text{SAR} := 10 \log_{10} \left(\frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2} \right) \tag{4.11}$$

Source-to-Interference Ratio (SIR)

La proporción de interferencia en una fuente, mide la cantidad de otras fuentes presentes en la fuente estimada [10]. Definida por:

$$\text{SIR} := 10 \log_{10} \left(\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2} \right) \tag{4.12}$$

Source-to-Distortion Ratio (SDR)

Relación señal distorsión, es considerada una medida que indica qué tan bien suena una fuente estimada, en otras palabras, esta mide la "calidad general" de las fuentes estimadas. Por lo general, es la métrica que la mayoría de aproximaciones reportan en sus trabajos [10]. Se calcula de la siguiente manera:

$$\text{SDR} := 10 \log_{10} \left(\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2} \right) \tag{4.13}$$

Signal-to-Noise Ratio (SNR)

Relación señal ruido, define la relación entre la potencia de la señal y la potencia del ruido que corrompe la señal. Esta se describe como:

$$\text{SNR} := 10 \log_{10} \left(\frac{\|s_{\text{target}}\|^2}{\|s_{\text{target}} - \hat{s}\|^2} \right) \tag{4.14}$$

Scale-Invariant Source-to-Distortion Ratio (SI-SDR)

Debido a la forma en que SDR calcula los términos e_{interf}, e_{noise}, e_{artif}, hace que las puntuaciones se inflen artificialmente. Por lo tanto, surge la relación de fuente a distorsión invariable en escala (SI-SDR), la cual remedia esto eliminando la dependencia del SDR en la escala de amplitud de la señal [5]. Esta se calcula de manera

similar, usando la ecuación 4.13, sin embargo, para lograr la invariancia en la escala, se realiza un proceso de reescalado previo a los cálculos.

Scale-Invariant Source-to-Noise Ratio (SI-SNR)

Relación fuente-ruido invariable en escala, se calcula de la misma forma que la ecuación 4.14, sin embargo, la invariancia de escala se asegura al normalizar tanto la fuente estimada como la verdadera a media cero antes del cálculo [7]. Esta métrica también suele ser utilizada como función de costo por modelo como [7].

4.4.2. Medidas subjetivas

Las medidas subjetivas implican que los calificadores humanos realicen una puntuación subjetiva de los resultados del sistema de separación. Estas medidas son costosas, requieren un mayor esfuerzo en tiempo y los resultados están sujetos a la percepción de los evaluadores humanos. Pueden ser más fiables que las medidas objetivas, ya que en este proceso participan oyentes reales [10]. Las medidas más utilizadas en las diferentes aproximaciones del estado del arte son: la evaluación perceptiva de la calidad subjetiva (PESQ) y la puntuación media de opinión (MOS), las cuales son puntuaciones que surgen de encuestas realizadas a oyentes a los que se les pide que califiquen la calidad de la separación en un escalada dada.

4.5. Similitud entre hablantes

Luego de probar los modelos Conv-TasNet [7], Dual-path RNN [6] y SepFormer [13], con diferentes audios del conjunto de datos de prueba del corpus de grabaciones de llamadas, usando modelos pre-entrenados, brindados por los autores de las aproximaciones. Se identificó a través de una encuesta de percepción de la calidad de separación, que para hablantes con voces similares, todos estos modelos obtenían resultados poco favorables, comparados con los resultados obtenidos cuando las voces de los hablantes en cuestión sonaban muy diferentes. Un ejemplo claro se presentaba

cuando los hablantes eran de géneros opuestos, el modelo prestaba mejores resultados de separación a nivel perceptivo y a nivel de métrica.

La encuesta fue realizada a 23 oyentes en donde se utilizaron 4 muestras audio, de las cuales dos eran de personas con voces perceptualmente similares, mientras que las otras eran de voces diferentes. A cada encuestado se le preguntó: en una escala de 1 a 5, donde 1 significa que no realiza separación de hablantes y 5 que la separación de hablantes es perfecta, ¿Cómo evalúa la calidad de la separación de hablantes?. Cada muestra contenía el audio mixto, audio del hablante 1 y audio del hablante 2.

Respecto a la encuesta de calidad de separación. los audios con voces perceptualmente similares obtuvieron una similitud coseno de 0.07 y 0.16 respectivamente, mientras que los audios con voces perceptualmente diferentes tuvieron una similitud coseno de 0.03 y 0.07. Todas las similitudes fueron calculadas usando el Speech Embedding Wav2Vec, modelo que se describe más adelante.

Una vez consolidado los resultados de la encuesta, se procedió a obtener valor de correlación entre las variables: similitud de hablantes y la calificación promedio de la calidad de separación, obteniendo así un valor de -0.7, valor que permite generar una hipótesis sobre una relación inversa entre la similitud de hablantes y los resultados de separación, dado que ha mayor similitud, menor son los resultados favorables de una buena separación a nivel perceptivo y a nivel de métrica (SI-SDR). Para poder validar dicha hipótesis, primero definimos la similitud entre hablantes como: la similitud de coseno entre los vectores representativos de los hablantes involucrados en un audio.

Para poder obtener dichos vectores representativos, usamos un speech embedding (Embebimiento de voz). Es un modelo que para nuestro caso, toma como entrada una fuente de audio (forma de onda) directamente sin procesar y genera una matriz, la cual contiene las características del audio en cuestión, que luego serán usadas como nuestro nuestros vectores representativos.

Los Speech Embedding, cumplen el mismo objetivo que los Word Embedding en el campo del NLP (Procesamiento de lenguaje natural). Su objetivo es realizar un transformación de una fuente de información en un vector representativo de la misma. En la siguiente sección describiremos los modelos Wav2Vec y Pyannote, estos nos

permitirán obtener los vectores representativos, ya que estos cuentan con un Speech Embedding.

4.5.1. Embebimientos de voz

El reconocimiento de voz es una tarea que gracias al desarrollo del aprendizaje automático, específicamente el aprendizaje profundo ha logrado avances significativos. Sin embargo, esta tecnología está disponible solo para una pequeña parte de los miles de idiomas que se hablan en el mundo. Esto debido a que los sistemas actuales de reconocimiento del habla, requieren miles de horas de habla transcrita para alcanzar un rendimiento aceptable[1]. Requerimiento que simplemente es difícil de tener, para los casi 7.000 idiomas que se hablan en el mundo.

Debido a esta necesidad puntual, nace los modelos Wa2Vec y variantes: Wav2Vec 2.0 y Wav2Vec-U, los cuales se pueden entrenar para realizar reconocimiento automático de voz (ASR), usando audios sin procesar ni etiquetar (tener la transcripción). A continuación, describiremos un modelo en concreto, el cual utilizamos como embedding para el cálculo de la similitud entre hablantes.

Wav2Vec 2.0

El objetivo de este modelo es aprender representaciones de audio, sin usar ningún dato etiquetado (transcripciones). En otras palabras, se busca aprender la estructura del habla a partir de audio sin procesar y sin etiquetar. Representaciones que luego serán útiles para tareas de reconocimiento de voz.

Wav2Vec 2.0, codifica el audio del habla a través de una red neuronal convolucional multicapa y luego enmascara tramos de las representaciones latentes del habla resultantes. Las representaciones latentes se introducen en una red transformer, para construir representaciones contextualizadas. El modelo se entrena mediante una tarea de contraste, en la que hay que distinguir entre la representación latente verdadera y los distractores. Este modelo, parece resolver el problema de la ausencia de datos etiquetados, ya que con solo una hora de datos etiquetados, Wav2Vec 2.0,

supera el estado del arte de los modelos ASR, el cual utiliza un subconjunto de 100 horas. Además, con sólo diez minutos de datos etiquetados y un preentrenamiento con 53.000 horas de datos sin etiquetar, este modelo logra conseguir una tasa de error de palabras (WER) de 4,8/8,2[1]. Debido a su desempeño, este modelo promete ser el camino para construir modelos de reconocimiento de habla, con cantidades limitadas de datos transcritos (etiquetados).

Este modelo utiliza el paradigma de aprendizaje self-supervised o auto-supervisado. Este paradigma aprende representaciones de datos generales, a partir de ejemplos no etiquetados, para luego afinar el modelo usando un conjunto de datos etiquetados[1].

Pyannote

Pyannote [2] es un conjunto de herramientas de código abierto escrito en Python para la diarización de hablantes. Basado en el framework Pytorch, este contiene modelos pre-entrenados que cubren una amplia gama de dominios para la detección de actividad de voz, detección de cambio de altavoz, detección de voz superpuesta y embebimiento de voz, este último es el que utilizaremos para generar los vectores representativos de los hablantes y poder calcular una similitud entre ellos.

Específicamente el modelo embebimiento, se basa en la arquitectura canónica de Time Delay Neural Network (TDNN) de ***vectores x***, pero con bancos de filtros reemplazados por funciones de SincNet entrenables [11].

Finalmente, usando la distancia del coseno directamente, este modelo alcanza una tasa de error equivalente (EER) del 2,8 % con el conjunto de prueba del dataset VoxCeleb 1 [2].

4.6. Conv-TasNet modificada

En la sección sobre similitud entre hablantes, mencionamos que existe una posible relación inversa entre la similitud de hablantes y los resultados de separación, obteniendo mejores resultados de separación tanto a nivel de métrica como perceptivos, cuando la similitud entre hablantes es menor.

Asumiendo que esta hipótesis es verdadera, incluir un término a optimizar dentro de la función de costo del modelo base (Conv-TasNet), el cual este relacionado con la similitud entre hablantes, podría mejorar los resultados del modelo actual, ya que durante el proceso de entrenamiento, estaríamos indicándole al modelo, cuales muestras probablemente requieran mayor atención. Siendo estas, aquellas que tengan un mayor error, el cual estará definido por el error base (SI-SDR) y un error asociado a la similitud entre hablantes.

Para poder agregar el término asociado a la similitud de hablantes dentro de la función de costo, debemos realizar una modificación a la arquitectura Conv-TasNet como se observa en la figura 4-10, agregando a la salida del decodificador un componente que calcula los vectores de embebimiento (Wav2Vec o Pyannote), para luego usar estos y calcular la similitud coseno y así poder computar una función de costo global.

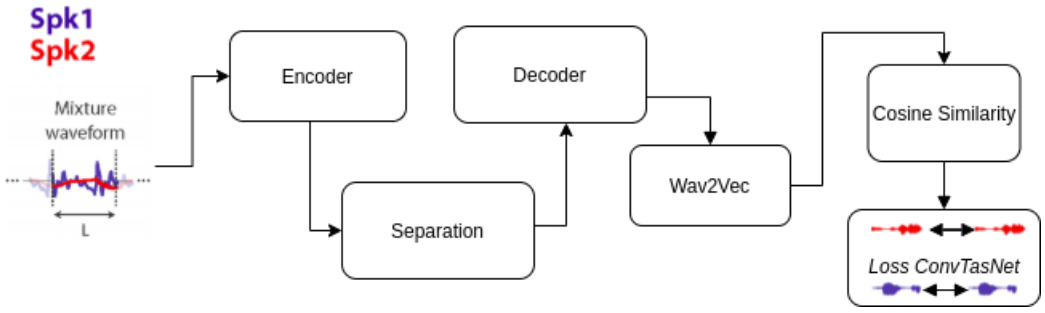


Figura 4-10: Componentes claves de la arquitectura Conv-TasNet modificada usando Wav2Vec

Agregando dicho término de similitud a la función de costo base, penalizaríamos aquellas muestras de entrenamiento, cuyos hablantes tengan una mayor similitud. Si la similitud entre hablantes es mayor, mayor será el error aportado.

Antes de describir la nueva función de costo, definiremos la similitud de coseno como:

$$\text{Cosine Speech Similarity} = \frac{x_1 \cdot x_2}{\text{máx}\left(\|x_1\|_2 \cdot \|x_2\|_2, \epsilon\right)} \tag{4.15}$$

Donde x_1 y x_2 son en este caso, los vectores representativos de los hablantes

involucrados en el audio. Dichos vectores, son obtenidos del modelo Wav2Vec 2.0. Finalmente, ϵ, es utilizado para evitar la división por cero.

Por otro lado, la similitud entre hablantes, esta dada por:

$$\text{Speech Similarity} = -1 * WeightSL * \log\left(\frac{1 - \text{Cosine Speech Similarity}}{2}\right) \quad (4.16)$$

Donde $WeightSL$, es un término utilizado para darle un peso o importancia a la componente de similitud, dentro de la función de costo general. El signo negativo, se utiliza para poder indicar al modelo que debe maximizar dicho término, esto debido a que el algoritmo de gradiente descendente minimiza una función por defecto.

Finalmente, la expresión completa que define la función de costo es:

$$\text{Loss ConvTasNet} = \text{SI-SDR} + \text{Speech Similarity} \quad (4.17)$$

Donde SI-SDR, es el error medido en el modelo base(Conv-TasNet) y el Speech Similarity, es el error asociado a la similitud entre hablantes.

Esta modificación realizada al modelo Conv-TasNet, requiere una fase experimental en la cual se evalúen diversos valores de $WeightSL$ y diversos vectores representativos. Esto debido a que el modelo Wav2Vec 2.0, cuenta con 12 diferentes salidas correspondientes a capas intermedias del modelo, por lo tanto existen 12 representaciones diferentes de un mismo audio. Adicionalmente se debe explorar el Pyannote embedding. Dichos experimentos se describen en el capítulo 5.

Capítulo 5

Marco experimental

En la siguiente sección, describiremos todos los componentes claves relacionados con la fase experimental: corpus, experimentos, validación, etc.

5.1. Corpus de llamadas telefónicas

Para poder entrenar estos modelos, es necesario contar con un conjunto de datos, compuesto por 3 audios por muestra: un audio que contiene la mezcla o grabación de la llamada y dos audios con las intervención individuales de los hablantes. Es de aclarar que todos los audios deben tener la misma duración.

Luego de realizar un búsqueda en la web, se logró encontrar 3 conjuntos de grabaciones de llamadas en español (corpus): CallFriend, CallFriend-Caribbean-Accent-Spanish, CallHome [9], los cuales son proporcionados por TalkBank, un proyecto liderado por Brian MacWhinney y la Universidad Carnegie Mellon, cuyo objetivo es fomentar la investigación fundamental en el estudio de la comunicación humana, con énfasis en la comunicación hablada [8]. Estos conjuntos de datos, contiene grabaciones de llamadas en formato mp3 y wav, grabadas a una frecuencia de muestreo de 8KHz y en canal estéreo (dos canales), lo cual permite obtener fácilmente las fuentes o intervenciones individuales de los hablantes, ya que fueron grabadas en canales diferentes.

La tabla 5.1, describe la duración de cada uno de los conjuntos de datos:

id	Nombre Corpus	Duración(Horas)
1	CallFriend-Spanish(CF)	36.6
2	CallFriend-Caribbean-Accent-Spanish	26
3	CallHome-Spanish	32.4
4	CallFriend-Caribbean-CallHome (All)	95.01

Cuadro 5.1: Conjuntos de datos de llamadas.

Los corpus: CallFriend-Spanish y CallFriend-Caribbean-Accent-Spanish , consta de 60 conversaciones telefónicas sin guión entre hablantes nativos de español. Las conversaciones grabadas, duran máximo 30 minutos. Este conjunto de datos fue obtenido gracias a una campaña realizada a través de Internet, publicaciones (anuncios) y contactos personales. En esta campaña, participaron 100 personas por dialecto, que iniciaron las llamadas, cada uno realizó una única llamada telefónica y la mayoría de los participantes llamaron a familiares o amigos cercanos. Una vez completada con éxito la llamada, la persona que llamó recibió un pago de 20 dólares. Este conjunto de datos, tuvo dos auditorías humanas; en la primera, se verificó que se estuviera hablando el idioma correcto y la calidad de la grabación. La segunda auditoría, fue realizada por un hablante nativo, familiarizado con la variación dialectal del español, esto con el objetivo de etiquetar, las conversaciones como çaribeñas.º "no caribeñas", según los atributos particulares del habla de los participantes.

Por otro lado, el corpus CallHome-Spanish, fue obtenido de la misma manera, a través de una campaña, sin embargo en esta campaña participaron 200 personas que iniciaron las llamadas. A cada persona, se le permitió hablar hasta 30 minutos, además, también recibió un pago de 20 dólares luego de realizar la llamada.

Finalmente, el conjunto de datos que hemos denominado CallFriend-Caribbean-CallHome (All), es la unión de los 3 conjuntos de datos. Este conjunto de datos, fue creado con el objetivo de experimentar, el comportamiento de los modelos al aumentar la cantidad de datos. Comportamiento que se explicará en detalle en las próximas secciones.

Nombre Corpus	Entrenamiento (Horas)	Validación (Horas)	Prueba (Horas)	Duración Total (Horas)
CallFriend-Spanish(CF)	26.4	2.9	7.3	36.6
CallFriend-Caribbean-Accent-Spanish	18.2	2	5.8	26
CallHome-Spanish	24.1	2.42	5.91	32.4
CallFriend-Caribbean-CallHome (All)	68.68	7.32	19.01	95.01
CallFriend-Spanish-15	25.67	3.34	7.6	36.6

Cuadro 5.2: Distribución conjunto de datos.

5.2. Experimentos y validación

5.2.1. División corpus

Cada uno de los 4 corpus de datos, presentados en la sección anterior, fue dividido en 3 diferentes subconjuntos: entrenamiento, prueba y validación. Con una distribución de: 70 %, 20 % y 10 % respectivamente, del tamaño total del conjunto de datos. Debido a que los audios de los corpus son de larga duración, alcanzando duraciones máximas de 30 minutos, se realizó una división de los mismos en porciones de 30 segundos. Adicional, para el corpus CallFriend-Spanish, se creó una versión con porciones de 15 segundos(CallFriend-Spanish-15).

La tabla 5.2, muestra la distribución en horas de los subconjuntos de datos, creados para cada uno de los corpus. El conjunto de entrenamiento, como su nombre lo indica, es utilizado para realizar el entrenamiento del modelo, el conjunto de validación, se emplea para seleccionar el mejor de los modelos entrenados y finalmente, el conjunto de prueba, nos permite saber el comportamiento de dicho modelo con muestras que jamás ha visto, además de obtener medidas de desempeño del mismo y saber cuando terminar el proceso de ajuste.

La figura 5-1, muestra la distribución de los 4 corpus en horas, para cada uno de los subconjuntos(entrenamiento, validación y prueba).

Finalmente, debido a que un mismo hablante se encuentra en diferentes muestras, se debe garantizar que todas las muestras de un mismo hablante queden en uno de los tres subconjunto (entrenamiento, validación, prueba). Esto garantiza que el modelo

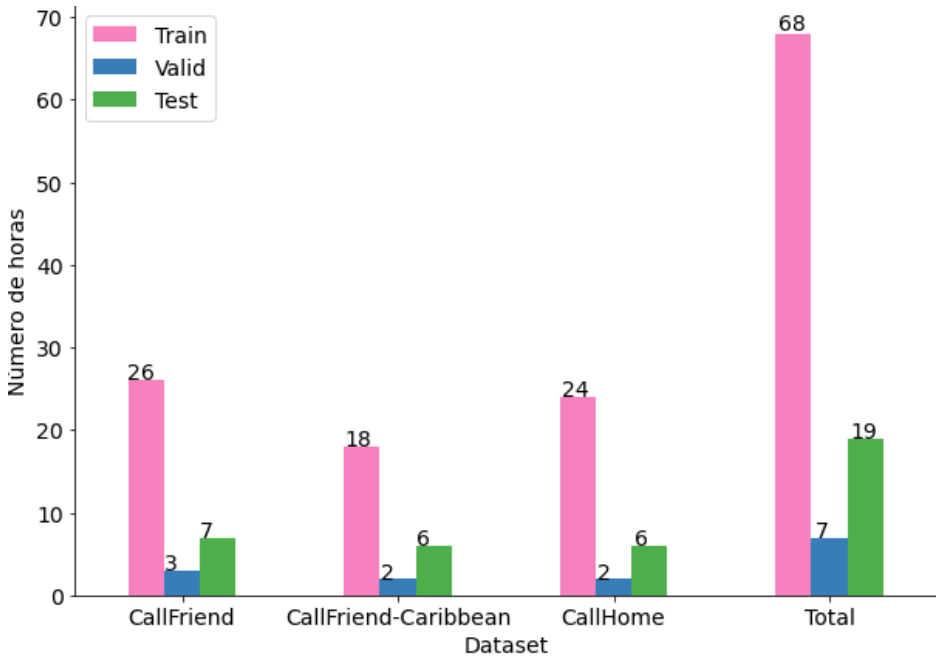


Figura 5-1: Distribución en horas de los corpus de grabación de llamadas en español

no se sobreajuste a un conjunto específico de hablantes, además, las métricas calculadas con el conjunto de pruebas no sean optimistas, ya que si el modelo durante el entrenamiento, procesa audios de un hablante que a su vez se encuentra en el conjunto de prueba, esté al conocerlo previamente podría generar resultados favorables pero sesgados, ya que dicho hablante ya es conocido por el modelo. Para poder garantizar dicha condición, se utilizó la metodología de validación Group Shuffle Split.

Capítulo 6

Resultados

6.0.1. Evaluación modelos pre-entrenados

Luego de realizar una revisión en el estado del arte de aproximaciones para la separación de hablantes, principalmente enfocadas en técnicas de Deep Learning. Se eligieron 3 arquitecturas(DPRNN [6] , SepFormer[13], Conv-TasNet[7]), para ser evaluadas con el conjunto de prueba de CallFriend-Spanish y CallFriend-Spanish-15. Esto con el objetivo de seleccionar una arquitectura base, sobre la cual realizar una fase experimental.

Para las 3 aproximaciones, se buscó su respectivo código fuente y modelos pre-entrenados, para poder realizar el cálculo de métricas del conjunto de prueba. Se realizaron algunas adaptaciones al código para poder monitorear y obtener datos de interés durante la prueba.

La tabla 6.1, muestra el desempeño de los 3 modelos usando la métrica SI-SDR, la cual se mide en decibeles (dB), un mayor valor indica un mejor desempeño. Cómo se logra apreciar, el modelo Conv-TasNet es superior a los demás modelo, con ambos corpus. Sin embargo, luego de realizar una inspección manual de los audios, notamos que en muchos audios del corpus CallFriend-Spanish-15, solo interviene 1 hablante, esto debido a la corta duración del audio. Por lo tanto, decidimos trabajar con audios de duración de 30 segundos, en los cuales hay una mayor probabilidad de que dos hablantes se encuentren involucrados. Condición que se acerca mas a el contexto del

Modelo	CallFriend-Spanish SI-SDR(dB)	CallFriend-Spanish-15 SI-SDR(dB)
Conv-TasNet	6.9	6.63
DPRNN	-0.70	-0.97
SepFormer	3.31	3.78

Cuadro 6.1: Desempeño modelos de la literatura con corpus CallFriend-Spanish

Modelo	CallFriend-Spanish Tiempo inferencia(Segundos)	CallFriend-Spanish-15 Tiempo inferencia(Segundos)
Conv-TasNet	5.5	3.1
DPRNN	6.5	3.6
SepFormer	3.4	1.5

Cuadro 6.2: Tiempo de inferencia promedio por audio de los modelo de la literatura, usando corpus CallFriend-Spanish.

problema que estamos abordando.

Por otro lado, como se logra apreciar en la tabla 6.2, el modelo SepFormer, logra tener el mejor desempeño respecto al tiempo de inferencia promedio por audio. Logrando un tiempo de inferencia promedio de 3.4 segundo, para audios de 30 segundos y 1.5 segundos, para audios de 15 segundos, aproximadamente la mitad del tiempo que necesita el modelo Conv-TasNet, sin embargo, su desempeño a nivel de métrica de separación, es inferior a la mitad de lo que logra conseguir el modelo Conv-TasNet. Por otro lado, el modelo DPRNN tiene un desempeño poco alentador, tanto en la métrica de separación como en tiempo de inferencia.

Respecto a la métrica de separación, el modelo DPRNN, no obtiene resultados favorables, ya que este modelo introduce artefactos artificiales en los audios generados. Esto fue validado de manera manual, escuchando una cantidad considerable de audios separados por este modelo.

Otro aspecto importante para la selección del modelo base, es la cantidad de parámetros de entrenamiento. Cómo se logra apreciar en la tabla 6.3, el modelo DPRNN tiene la menor cantidad de parámetros con tan solo 2.6 millones, por otro lado el SepFormer, tiene una gran cantidad de parámetros (26 millones), condición natural de los Transformers, componente clave dentro del modelo SepFormer. El modelo

Modelo	Cantidad de parámetros(Millones)
Conv-TasNet	5.1
DPRNN	2.6
SepFormer	26

Cuadro 6.3: Cantidad de parámetros, modelos de la literatura.

Conv-TasNet, tiene aproximadamente el doble de parámetros del DPRNN, pero su desempeño a nivel de métrica y tiempo de inferencia es superior.

Finalmente, dado que el modelo Conv-TasNet, tiene el mejor desempeño a nivel de métrica de separación (SDR), superando al segundo modelo(SepFormer), por aproximadamente el doble de la métrica obtenida y teniendo más de 20 millones de parámetros menos. Se decide utilizar al modelo Conv-TasNet, como el modelo base. Modelo sobre el cual se realizará una fase de experimentación, donde se realizara varios entrenamientos: desde cero, partiendo de un modelo pre-entrenado (Transfer learning) y ajuste fino (Fine-tuning), todo esto con el objetivo de mejorar el desempeño del mismo, usando los corpus de llamadas telefónicas y agregando componentes adicionales, los cuales fueron mencionados en la sección de: **ConvTasNet modificada**.

6.1. Entrenamiento, validación y evaluación de arquitectura base (Conv-TasNet)

6.1.1. Experimentos usando el corpus CallFriend-Spanish

Utilizamos el corpus CallFriend-Spanish, el cual contiene audios de 30 segundos, 26.4 horas de entrenamientos, 7.3 horas de prueba y 2.9 horas de validación. Inicialmente, realizamos el entrenamiento del modelo desde cero por 200 épocas, sin embargo, se presentó una parada anticipada en la época 146, obteniendo una métrica de 9 dB de desempeño, cómo se logra apreciar en la tabla 6.4.

Debido a que no se logró obtener una mejora significativa entrenando desde cero, realizamos 5 entrenamientos partiendo del modelo pre-entrenado, compartido por los

Tipo entrenamiento	Epocas de entrenamiento	Epocas entranadas	SI-SDR	Mejor epoca
Desde cero	200	146	9.035	116
Transfer-Learning	10	10	9.50	8
Transfer-Learning	20	20	11.51	17
Transfer-Learning	50	50	12.21	47
Transfer-Learning	100	100	13.18	90
Transfer-Learning	200	64	12.29	34

Cuadro 6.4: Entrenamientos modelo Conv-TasNet usando CallFriend-Spanish.

autores de la arquitectura. Esta técnica, conocida como Transfer learning, es muy popular en el campo del aprendizaje profundo y ha sido ampliamente utilizada en diversos campos, ya que ha demostrado mejorar el desempeño de los modelos y reducir los tiempos de entrenamiento, aprovechando el aprendizaje adquirido por un modelo entrenado para solucionar problemas relacionados.

Cómo se logra apreciar en la tabla 6.4, con solo entrenar el modelo por 10 épocas usando Transfer Learning, logramos superar al modelo entrenado desde cero por 146 épocas. Aquí se nota claramente el éxito de esta técnica. Se realizaron más entrenamientos, aumentando la cantidad de épocas. El modelo entrenado por 100 épocas y usando Transfer Learning, es el que mejores resultados obtiene, con un SI-SDR de 13.18 dB, seguido del modelo entrenado por 200 épocas, el cual obtiene un valor de SI-SDR de 12.29 dB. Cabe resaltar que, estas métricas son obtenidas usando el conjunto de validación del corpus CallFriend-Spanish.

Luego de identificar las dos mejores configuraciones del modelo, se procedió a realizar la evaluación de los mismos, utilizando el conjunto de datos de prueba del corpus CallFriend-Spanish, datos que ninguno de los modelos ha visto. Con estos resultados podremos determinar si existe una mejora real del modelo, respecto al proporcionado por los autores y con el cual se obtuvo un SI-SDR de 6.9 dB para el mismo corpus.

Cómo se logra apreciar en la figura 6-1, el mejor modelo conseguido es el entrenado usando Transfer Learning por 100 épocas(CF-reTrain100). Este obtiene un SI-SDR de aproximadamente 10 dB, superando en 3 dB al modelo brindado por los autores

(CF-pretrained). Por otro lado, el peor modelo es el entrenado desde cero durante 200 épocas. El modelo entrenado con Transfer Learning por 200 épocas, es el segundo mejor modelo. Esto nos demuestra nuevamente que usando Transfer Learning, podemos obtener resultados superiores.

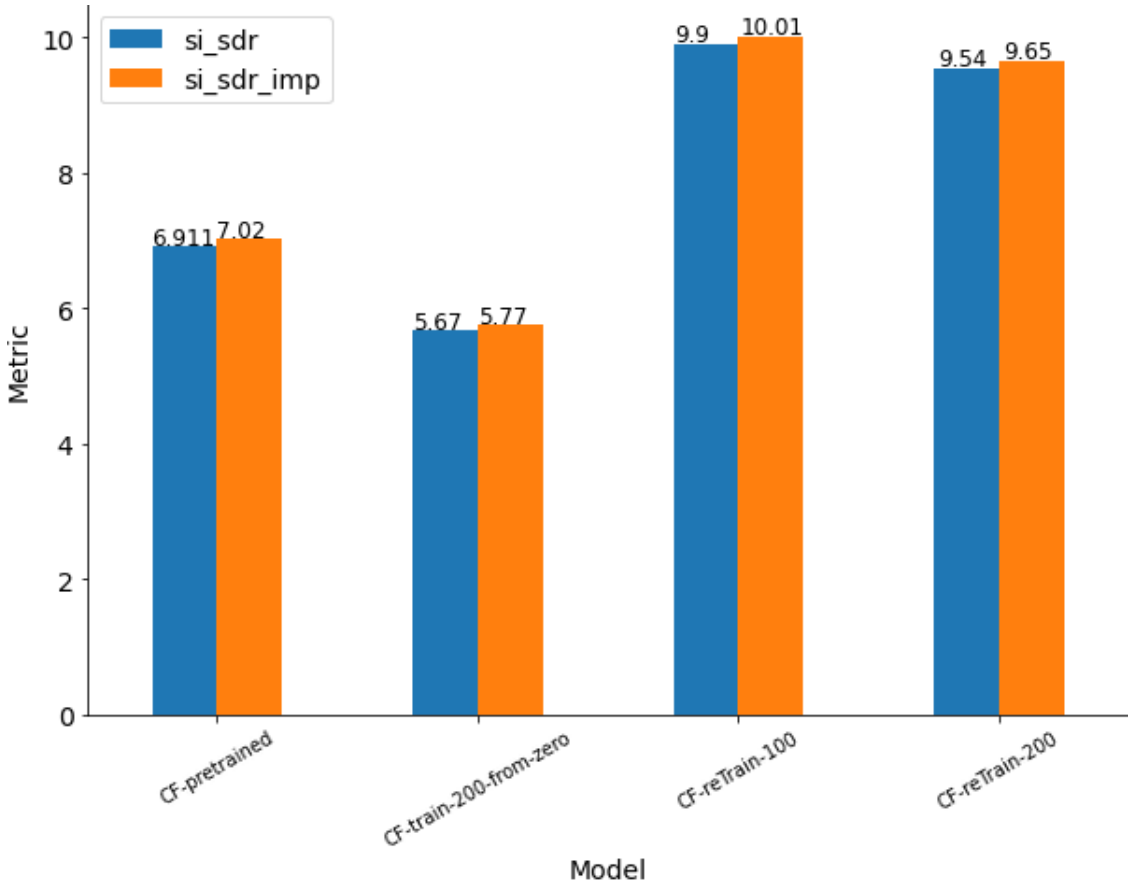


Figura 6-1: Desempeño modelos Conv-TasNet entrenado con CallFriend-Spanish.

Experimentos usando el corpus CallFriend-Caribbean-CallHome (All)

Luego de realizar entrenamientos con el corpus CallFriend-Spanish, decidimos realizar dos experimentos utilizando el corpus ALL. Este corpus contiene 68.68 horas para entrenamiento, 7.32 horas para validación y 19 horas de prueba.

Inicialmente, se entrenó el modelo desde cero por 200 épocas, sin embargo, el entrenamiento tuvo una parada temprana en la época 102, ya que no se logró una mejora en la métrica en validación, obtenido un SI-SDR de 13.62 dB como se logra

Tipo entrenamiento	Epocas de entrenamiento	Epocas entranadas	SI-SDR	Mejor epoca
Desde cero	200	102	13.62	72
Transfer-Learning	200	93	16.13	63

Cuadro 6.5: Entrenamientos modelo Conv-TasNet usando corpus CallFriend-Caribbean-CallHome (All).

apreciar en la tabla 6.5. Luego, se utilizó Transfer Learning, entrenando el modelo por 200 épocas, nuevamente, se presentó una parada temprana durante el entrenamiento en la época 93, logrando un SI-SDR de 16.13 dB , el cual es aproximadamente 3 dB mejor que entrenar el modelo desde cero. Estos valores fueron obtenidos utilizando el conjunto de validación del corpus ALL.

Realizando la evaluación de los dos modelos, utilizando el conjunto de prueba del corpus CallFriend-Caribbean-CallHome (All), se obtiene un SI-SDR de 7.29 dB, para el modelo entrenado desde cero y un SI-SDR de 12.4 dB, para el modelo entrenado con Transfer Learning, como se observa en la figura 6-2. Además, el modelo brindado por los autores obtiene un SI-SDR de 6.98 dB, con este mismo conjunto de datos. A a nivel perceptual, el modelo entrenado usando Transfer Learning es superior al entrenado desde cero, lo cual nos demuestra una vez más el poder de esta técnica.

Aunque los modelos entrenados con el corpus ALL y CallFriend-Spanish no pueden ser comparables entre sí, ya que fueron entrenados y validados con diferentes fuentes de información, se realizó una comparación a nivel perceptual, utilizando audios que ninguno de los modelos había procesado. Identificando así, que el modelo entrenado usando Transfer Learning y el corpus CallFriend-Spanish (CF-reTrain-100), presenta un desempeño superior a los demás modelos.

Por otro parte, utilizando el modelo que fue entrenado usando Transfer Learning y el corpus CallFriend-Spanish (CF-reTrain-100), realizamos varias pruebas con audios del conjunto de prueba CallFriend-Spanish, para determinar algunas posibles condiciones bajo las cuales el modelo tiene un buen desempeño, y otras donde su comportamiento no esta tan favorable. Esto con el objetivo de hacer alguna modificación al mismo y poder mejorar su desempeño.

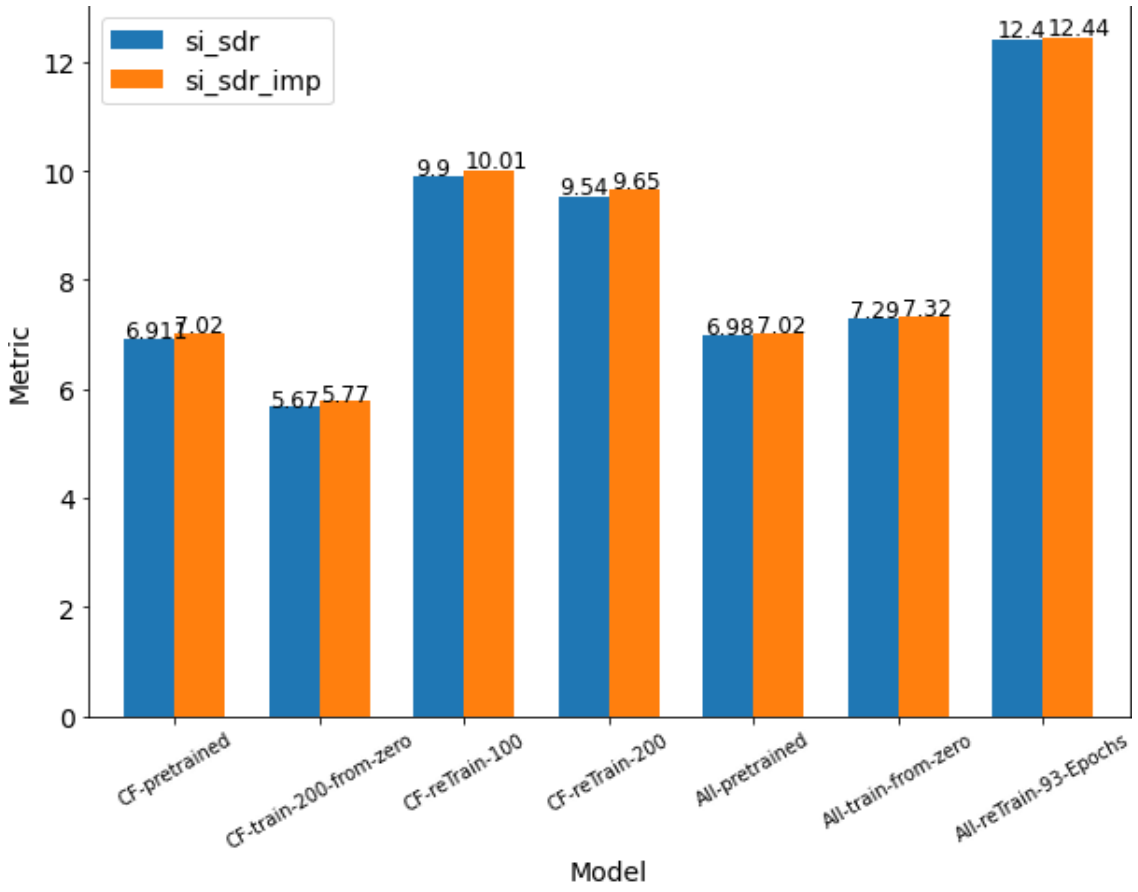
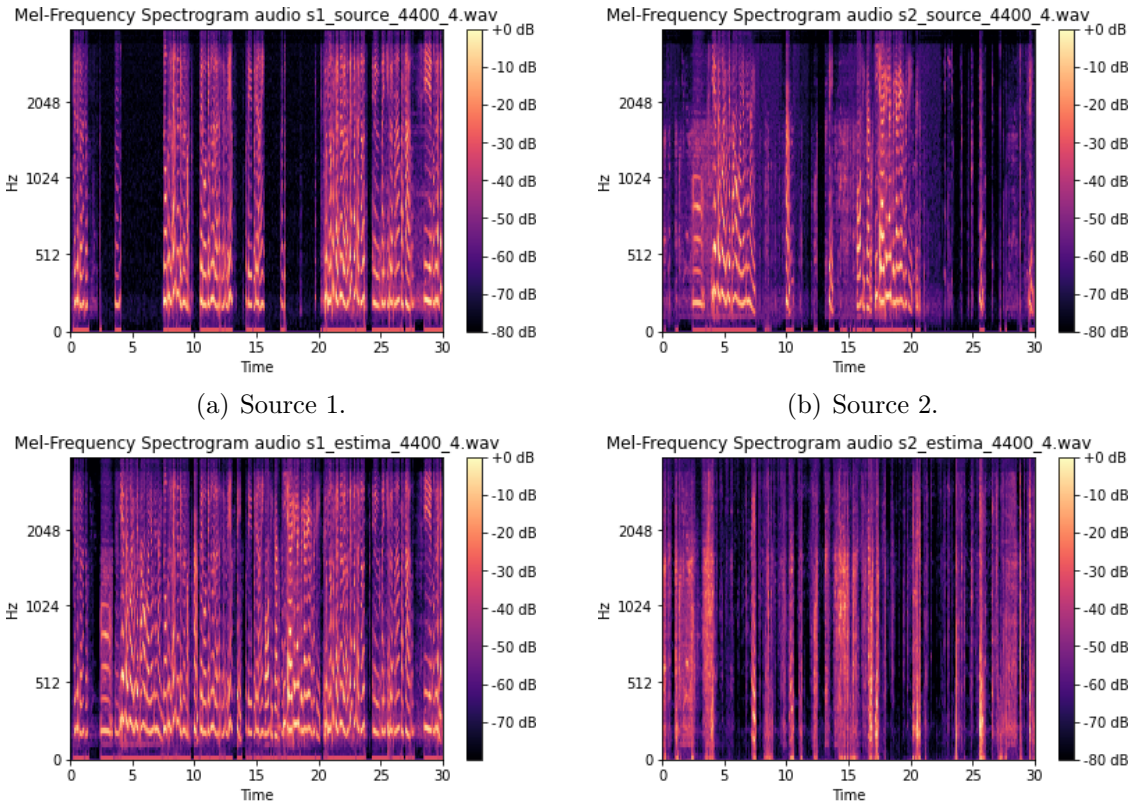


Figura 6-2: Desempeño modelos Conv-TasNet entrenado con corpus CallFriend-Caribbean-CallHome (All).

Se identificó que cuando los hablantes de la llamada tienen estilos de voces perceptualmente similares, al modelo le cuesta realizar una buena separación. Esto se ve reflejado a nivel de métrica y a nivel perceptual.

En la figura 6-3, se muestran los espectrogramas en escala Mel de la fuente limpia y estimada de un audio, donde están involucradas dos mujeres con voces perceptualmente similares, el modelo obtuvo un desempeño de -9.14 dB para la métrica SI-SDR, el cual es poco favorable. De manera gráfica, podemos ver que las estimaciones se encuentran muy distantes de las fuentes limpias.

Por otro lado, cuando las voces son perceptualmente diferentes los resultados son favorables, caso que se puede ver claramente en la figura 6-4, donde los espectrogramas estimados y limpios son gráficamente similares. Además a nivel de métrica de separación (SI-SDR), se obtiene un valor de 21.74 dB.



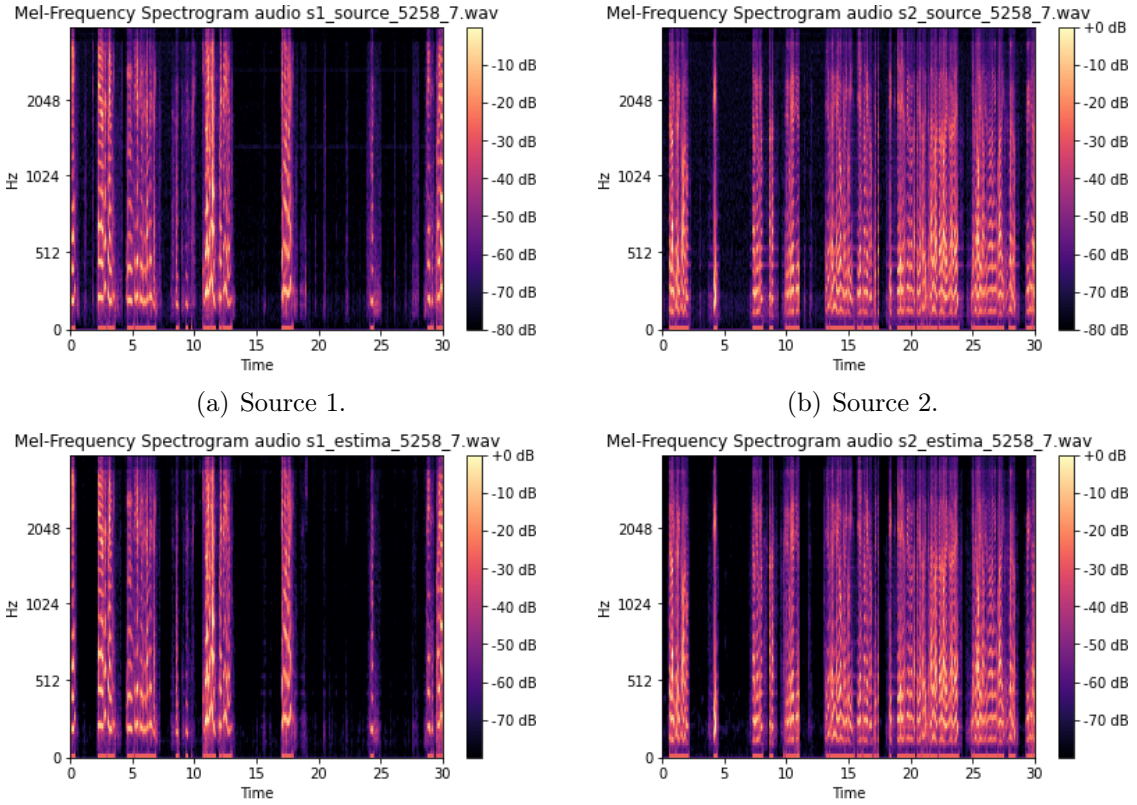
(a) Source 1.

(b) Source 2.

(c) Estimate 1.

(d) Estimate 2.

Figura 6-3: Comparación espectrogramas fuentes estimadas y originales de un audio de ejemplo con mala separación.



(a) Source 1.

(b) Source 2.

(c) Estimate 1.

(d) Estimate 2.

Figura 6-4: Comparación espectrogramas fuentes estimadas y originales de un audio de ejemplo con buena separación.

A partir de las observaciones anteriores, se decidió realizar una modificación a la arquitectura original, creando así la ConvTasNet modificada la cual fue descrita en el Marco teórico y cuyos resultados se muestran a continuación.

6.1.2. Entrenamiento y validación arquitectura ConvTasNet modificada

Como se describió en la sección: **ConvTasNet modificada**, al modelo Conv-TasNet, se le realizó una modificación a su función de costo, para que incluya un error asociado a la similitud entre hablantes. De esta manera, podríamos obtener un modelo cuya capacidad de separación mejore, en escenarios donde los hablantes tengan voces perceptualmente similares.

Se realizaron un total de 15 experimentos usando Wav2Vec como embedding, en donde se entrenaron modelos Conv-TasNet modificados, usando Transfer Learning. Todos los modelos fueron entrenados durante 100 épocas. Cada modelo tiene un valor de $WeightSL$ diferente, el cual puede ser 5, 10 o 20. Además, dado que el Speech Embedding cuenta con 12 posibles vectores representativos, los cuales corresponden a las salidas de las 12 capas de dicho modelo. Se tomaron las capas 1, 2, 3, 4 y 12. Realizando la combinación de los posibles valores de estos dos hiper-parámetros. Obtuvimos un total de 15 experimentos. Finalmente, todos los experimentos se realizaron utilizando el corpus CallFriend-Spanish.

El objetivo inicial es determinar si existe una mejora real a nivel de métrica de separación y perceptual. Adicional a esto, se busca determinar una posible configuración, obteniendo el $WeightSL$ ideal y la capa del Speech Embedding que genere los mejores resultados.

En los primeros 5 experimentos, se fijó el valor de $WeightSL$ en 5 y se permuto el número de la capa del Speech Embedding dentro de los posible valores (1,2,3,4,12). Como se visualiza en la figura 6-5, el mejor resultado que se obtuvo, fue un valor de 10.6 dB en SI-SDR, el cual se logró en la capa 3. Dicho modelo lo nombraremos como: CF-TL-LossAdd-3-Trans-5-Weight. Este resultado supera al modelo CF-reTrain-100,

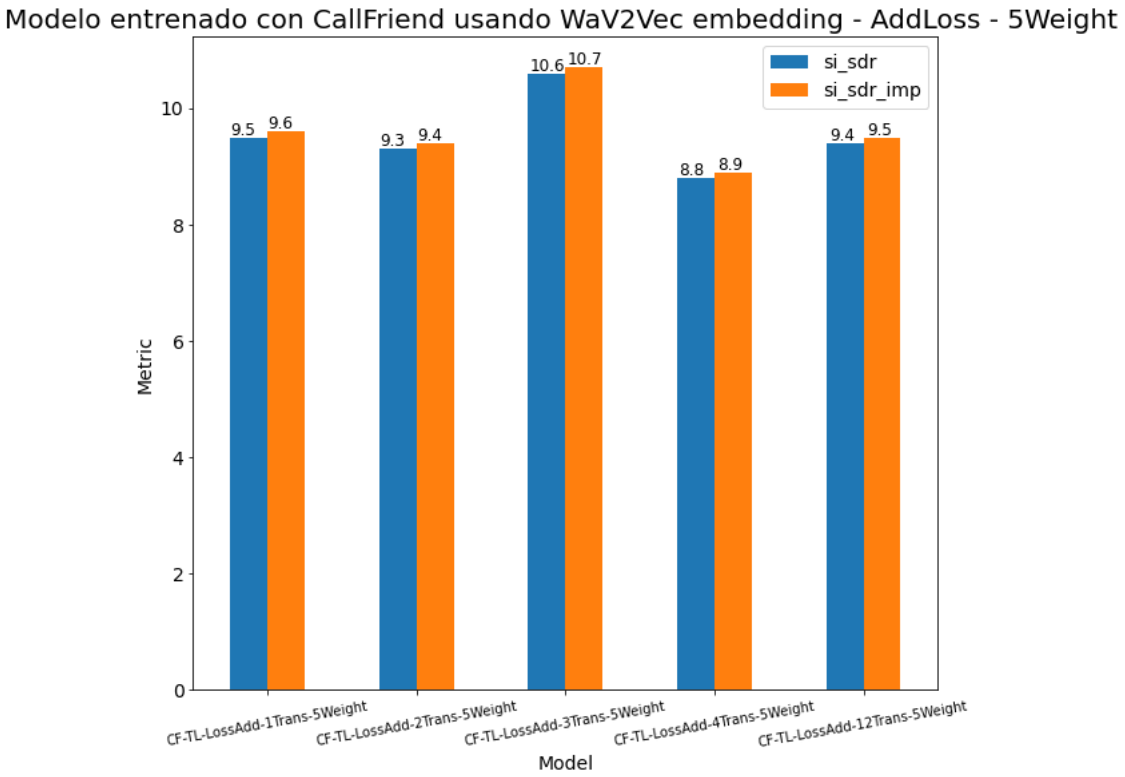


Figura 6-5: Resultados modelo Conv-TasNet modificado entrenado con CallFriend-Spanish y $WeightSL$ de 5.

el cual obtuvo un SI-SDR de 9.9. Demostrando así una mejora, la cual se debe a la modificación realizada a la función de costo.

En los siguientes 5 experimentos, se fijó el valor de $WeightSL$ en 10. Obteniendo dos modelos con un SI-SDR superior a 10 dB, como se muestra en la figura 6-6. Dichos modelos, se lograron usando la capa 2 y 4 del Speech Embedding y a pesar de que son superiores al modelo CF-reTrain-100, no logran superar al modelo CF-TL-LossAdd-3-Trans-5-Weight.

Para los últimos 5 experimentos, se fijó el valor de $WeightSL$ en 20. Los resultados no son tan prometedores ya que ninguno de los modelos logra un valor de SI-SDR superior a 10 dB, como se logra apreciar en la figura 6-7.

De estos 15 experimentos podemos concluir:

- Un mayor valor para $WeightSL$, no implica un mejor desempeño del modelo.
- No existe un número de capa del Speech Embedding, que garantice un mejor

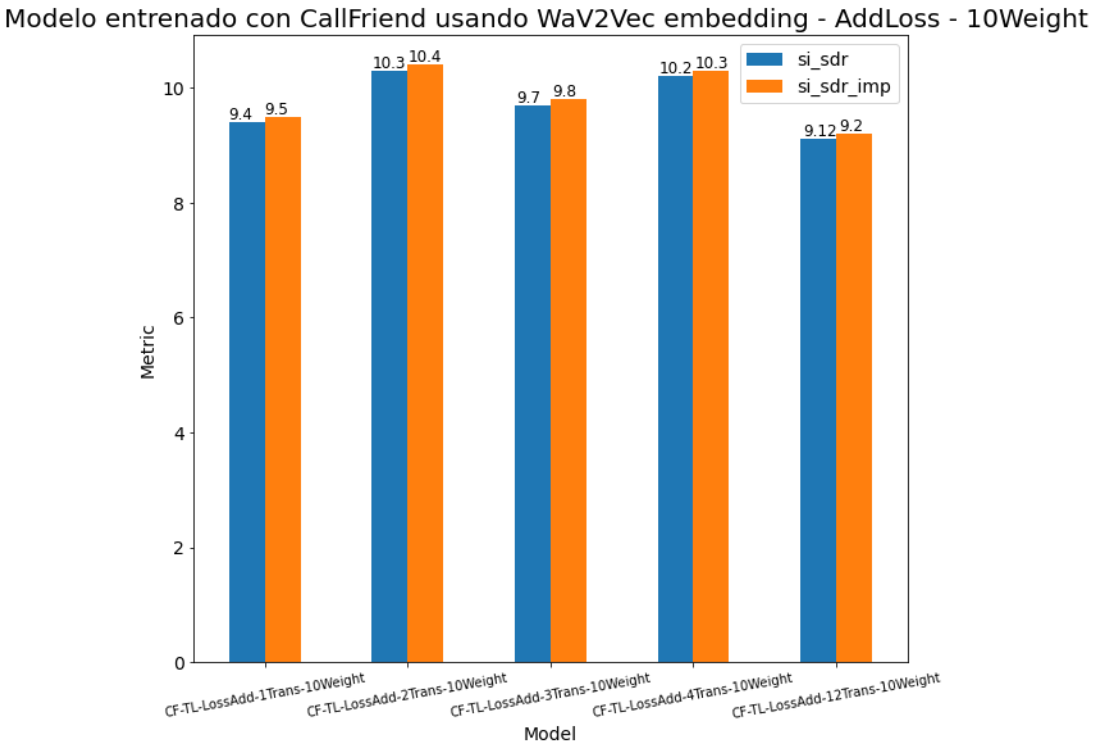


Figura 6-6: Resultados modelo Conv-TasNet modificado entrenado con CallFriend-Spanish y $WeightSL$ de 10.

desempeño del modelo.

- El modelo Conv-TasNet modificado, con un $WeightSL$ de 5 y que utiliza la capa 3 del Speech Embedding, es el mejor modelo encontrado. Por lo tanto, a nivel de métrica dicha modificación al modelo base significó un avance.

Por otro lado utilizando Pyannote como embedding y fijando el $WeightSL$ en 1, se obtiene un SI-SDR DE 9.7 dB, con un $WeightSL$ en 5 logramos un SI-SDR de 9.06 dB y finalmente con un $WeightSL$ de 10 conseguimos un SI-SDR de 8.7. Todos estos resultados fueron obtenidos utilizando el conjunto de prueba del corpus CallFriend-Spanish, por lo tanto son comparables con los resultados de los experimentos realizados con Wav2Vec, en donde a nivel de métrica dicho modelo es superior.

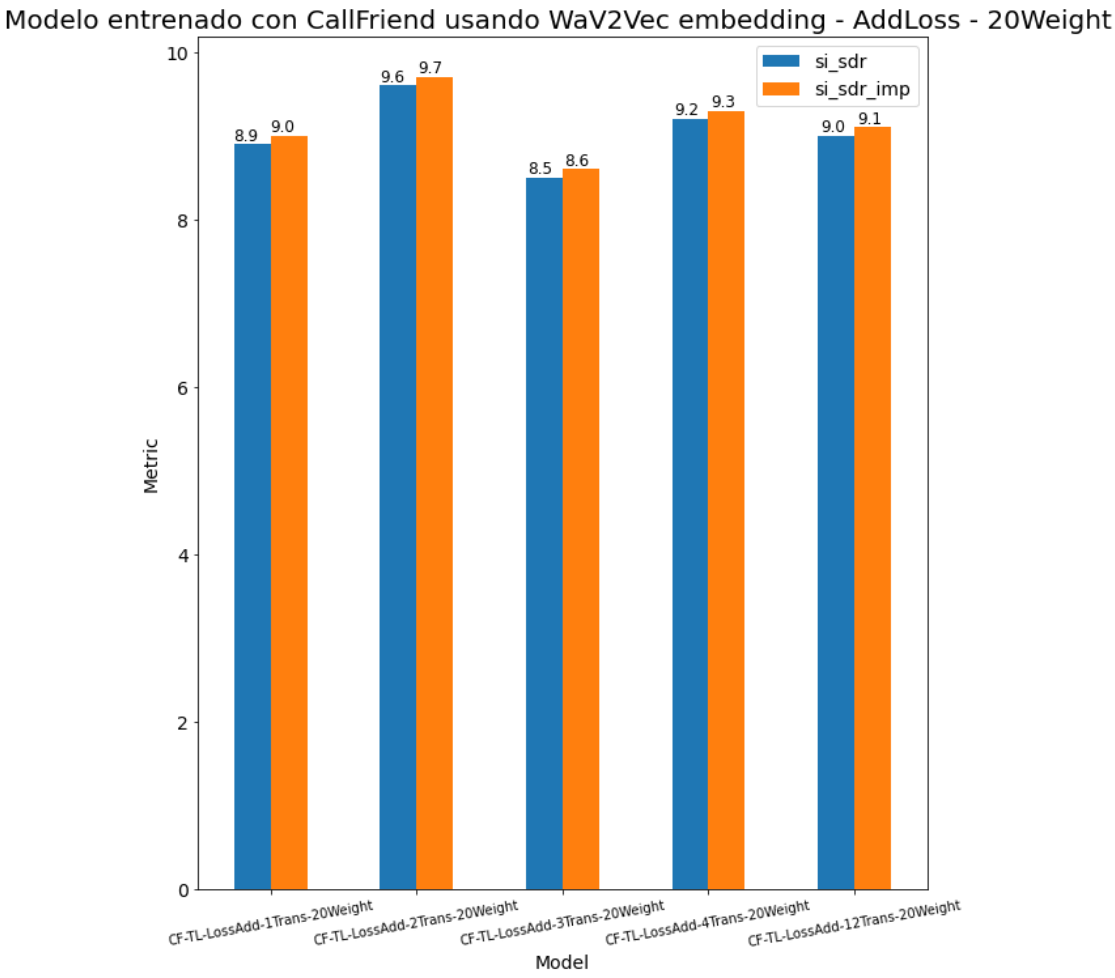


Figura 6-7: Resultados modelo Conv-TasNet modificado entrenado con CallFriend-Spanish y $WeightSL$ de 20.

6.2. Despliegue

Para el despliegue del modelo en tiempo real, se empleó la librería sounddevice, esta nos proporciona un API para reproducir y grabar matrices Numpy que contiene señales de audio. Disponible para sistemas operativo como: Linux, macOs y Windows.

Por otro lado, se utilizó la librería Tkinter para realizar una interfaz gráfica amigable con el usuario y poder utilizar el modelo en tiempo real. Como se muestra en la figura 6-8, la interfaz gráfica cuenta con dos simples botones: un botón para empezar a grabar, el cual al ser presionado comenzará a capturar el audio del micrófono e internamente ejecutara el modelo en tiempo real, reproduciendo al mismo tiempo las intervenciones de cada hablante a través de la salida de audio, usando un canal

específico para cada hablante (audífono izquierdo o derecho).

Figura 6-8: Interfaz gráfica de usuario del modelo desplegado en tiempo real.

Respecto al componente de separación en tiempo real, la figura 6-9, nos ilustra el funcionamiento del mismo, empezando por la captura del audio en tiempo real, el cual es manejado por una función (Audio callback) que se ejecuta en el Thread 1, esta función es la encargada de recibir los diferentes segmentos del audio en tiempo real y almacenarlos en una cola, para luego ser procesados por la función Separation uno a uno en orden de llegada.

La función Separation de la figura 6-9, cuenta con dos subprocesos: el primero se encarga de realizar la separación de las intervenciones de los hablantes del segmento en cuestión, el segundo toma los dos audios generados y los asigna a su ubicación correspondiente (Vector de intervenciones hablante 1 o Vector de intervenciones hablante 2), dependiendo de la similitud que exista entre dicho segmento y los segmentos de referencia de los hablantes, dichos segmentos tienen una duración de 1 segundo y son grabados al inicio del proceso, usando los botones: Grabar hablante 1 y Grabar hablante 2. Finalmente, se reproduce los audio de los hablantes por diferentes canales y conservando el orden, es decir, si el hablante 1 se reproduce por el canal 1 (audífono izquierdo), todas las intervenciones del mismo deberían escucharse por ese mismo canal.

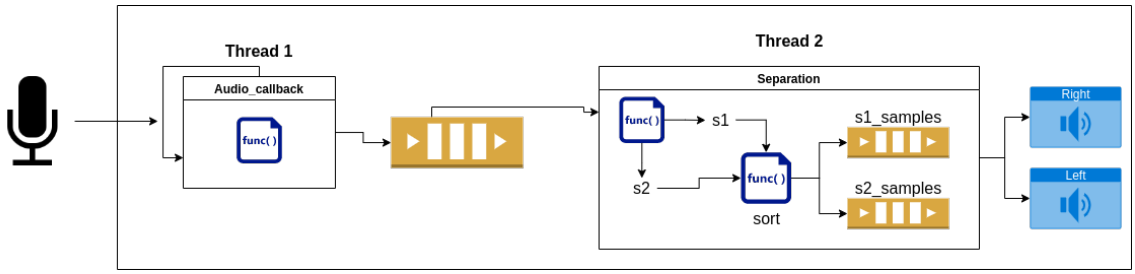


Figura 6-9: Diagrama funcionamiento del modelo desplegado en tiempo real.

Una vez realizado el montaje de la arquitectura mencionada en la figura 6-9, realizamos un experimento para medir el desempeño del sistema, para esto utilizamos 40 muestras del conjunto de prueba del corpus CallFriend-Spanish(CF), se realizó la separación de hablantes cambiando el tamaño del segmento a procesar en tiempo real, desde 1 hasta 10 segundos.

Para calcular la cantidad de errores, se realiza el cálculo de la distancia euclidiana entre los segmentos de audio limpio y los estimados, si dicha distancia supera un umbral previamente definido (1.0), el segmento estimado contiene un error, luego se cuentan los errores por muestras, se suman y se promedian. El valor de dicho umbral se determinó de manera experimental, observando el comportamiento de la distancia euclidiana en diferentes segmentos, usando 40 muestras del conjunto de pruebas del corpus CallFriend-Spanish, cada muestra fue separada utilizando diferentes segmentos de longitud de separación en tiempo real (1,2,...,10, 15, 20, 25 y 30 segundos), luego se graficaron las distancia entre segmentos de 1 segundo (30 segmentos y/o distancias) como se observa en la figura 6-11, se seleccionó un valor de referencia, el cual fue 5 y se procedió a comparar visualmente si los segmentos estimados que superaban dicho umbral eran muy diferentes a los verdaderos, usando la gráfica 6-10, en este caso se encontró eran muy diferentes y por lo tanto disminuimos el valor del umbral, hasta que las diferencias empezaron a ser mínimas visualmente, obteniendo un valor de 1.0 como umbral final.

Supongamos que realizamos la separación de hablantes en tiempo real usando segmentos de 2 segundos, la figura 6-10 muestran la forma de la onda limpia y estimada de dos hablantes presentes en una llamada. En la figura 6-10, se logra apreciar una

alto grado de similitud entre las formas de onda limpia y estimada, sin embargo hay dos pequeñas diferencias en dos segmentos en concreto, estas diferencias la podemos observar en la figura 6-11, la cual contiene las distancias entre segmentos de 1 segundo de las fuentes limpias y estimadas, para esta muestra en concreto usamos segmentos de longitud de 2 segundo para la separación en tiempo real, claramente podemos ver que hay dos segmentos que superan el valor del umbral (9.79 y 5.68), por lo tanto estos dos segmentos contiene un error de sincronización, dando así como resultado un error del 6.6 % (2/30 segmentos). De esta manera es como calculamos la cantidad de errores de estimación y logramos realizar la gráfica 6-12.

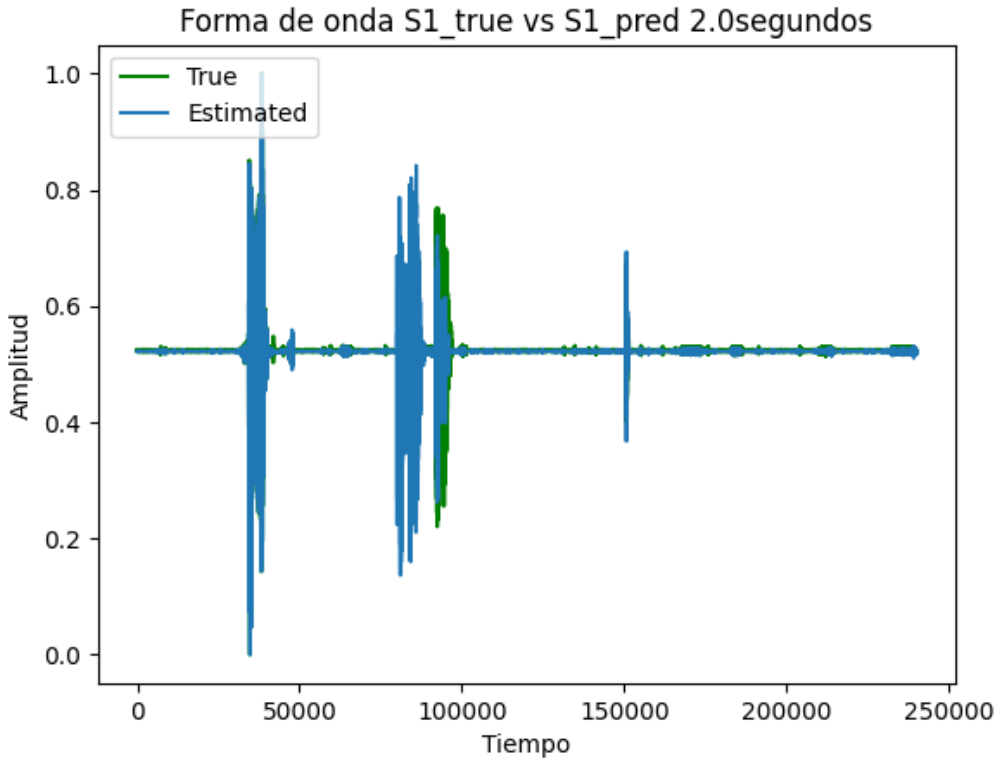


Figura 6-10: Forma de onda estimada vs forma de onda limpia usando segmentos de 2 segundos de duración, hablante número 1.

La gráfica 6-12, nos muestra una tendencia clara de una relación inversa entre la longitud del segmento a procesar en tiempo real y el porcentaje de errores después de los 15 segundos de longitud. Una mayor longitud de segmento a procesar en tiempo real, nos permitirá tener mejores resultados. Esto evidencia de manera cuantitativa

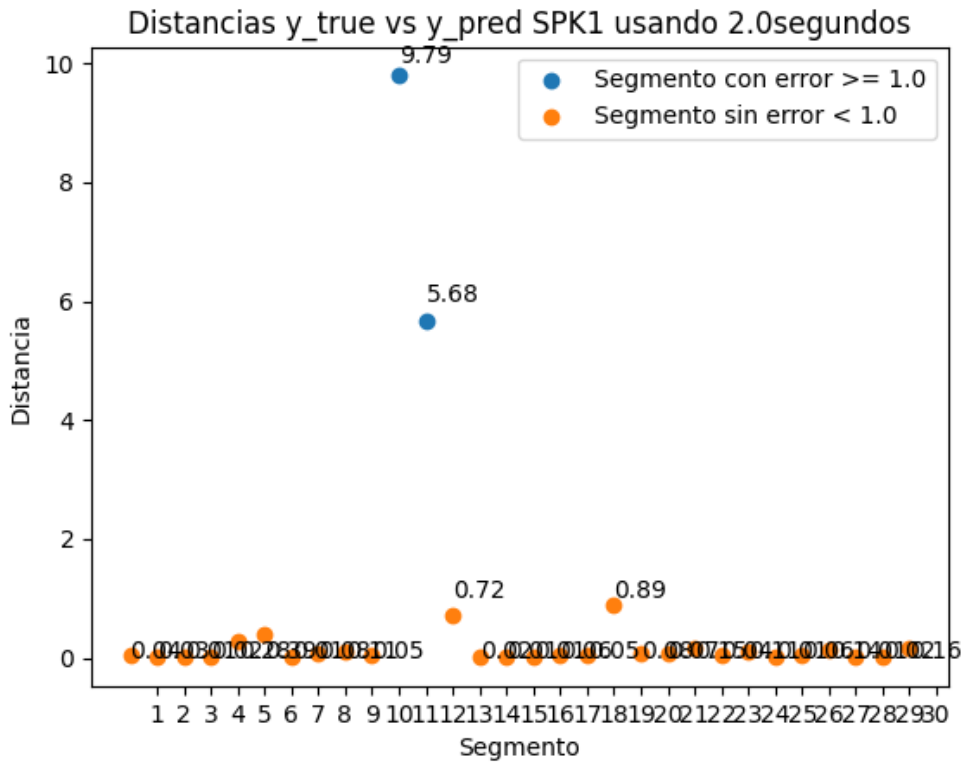


Figura 6-11: Distancias entre segmentos de 2 segundos de duración, hablante número 1.

el bajo rendimiento del sistemas desplegado en tiempo real usando segmentos de 1 segundo de duración.

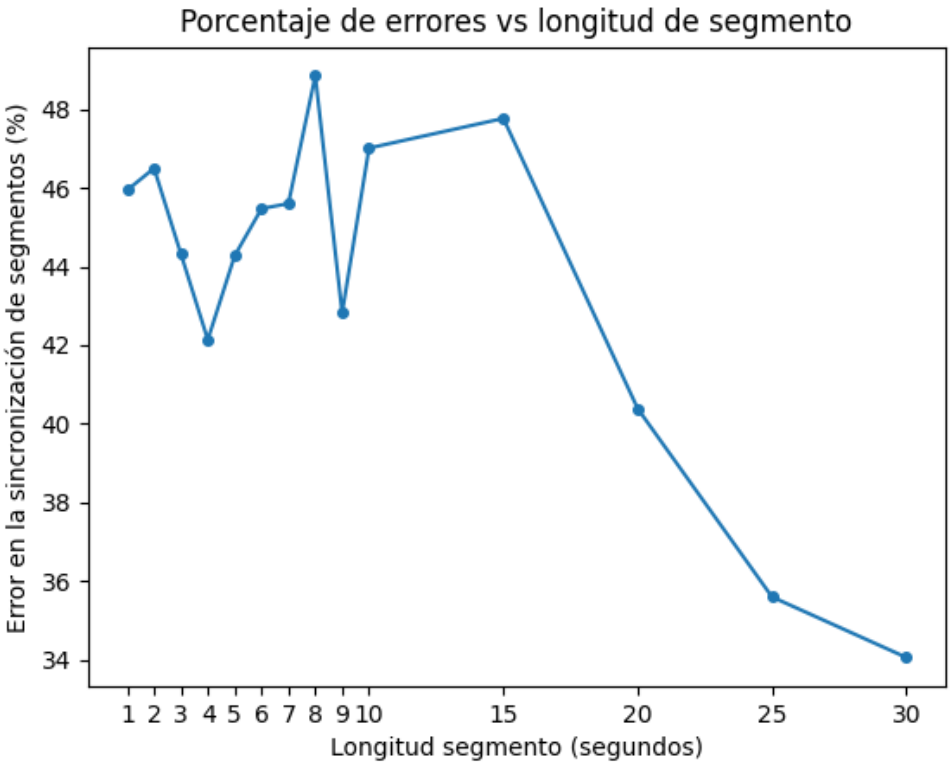


Figura 6-12: Porcentaje de errores vs longitud de segmento

Capítulo 7

Conclusiones

- El uso de Transfer Learning durante el proceso de entrenamiento del modelo Conv-TasNet, mejoró notablemente los resultados de separación de hablantes, pasando de una métrica (SI-SDR) de 6.9 dB a 9.9 dB, mejorando en 3 dB la calidad de la separación. Esto reafirma la gran utilidad de esta técnica, durante los procesos de entrenamiento.
- Se logró validar mediante una encuesta, la hipótesis sobre la relación inversa entre la similitud de hablantes y la calidad de separación de hablantes. Sin embargo, una de las muestras evaluadas presentaba un valor elevado de similitud entre sus hablantes (0.07) y la calidad de separación resulto ser buena (5), pero los hablantes involucrados eran de diferente sexo, lo cual podría ser la razón por la cual el desempeño del modelo no decae. Se sugiere para trabajos futuros desarrollar con mayor profundidad dicha hipótesis y explorar otros Speech Embeddings.
- El modelo Conv-TasNet modificado, fue el modelo con el mejor desempeño (10.6 dB en SI-SDR). La modificación a la función de costo usando el término de similitud entre hablantes, definido como la similitud coseno entre los vectores representativos dados por el Speech Embedding Wav2Vec, permitió mejorar el rendimiento del modelo en 0.7 dB.
- Usar el modelo Wav2Vec entrenado con un corpus en español no garantiza

mejoras en la métrica del modelo.

- La arquitectura planteada para el despliegue permitió el consumo del modelo en tiempo real, sin embargo dada la poca duración de los segmentos procesados (1 segundo), en la mayoría de los casos solo una hablante se encontraba presente, por lo tanto la separación daba como resultado 1 hablantes y ruido, pasando a ser un problema de identificación de hablantes más que un problema de separación de hablantes.

- Existe una relación inversa entre la longitud del segmento a procesar en tiempo real y el porcentaje de errores, una mayor longitud garantiza menor cantidad de errores de separación de hablantes.

- Todos los modelos fueron entrenados, ajustados y validados con audios de una duración de 30 segundos, sin embargo al realizar la prueba del modelo en tiempo real, usamos segmentos de 1 segundos (para garantizar el concepto de tiempo real), lo cual es un dominio distante de la realidad conocida por el modelo, lo cual explica su limitado desempeño.

Bibliografía

[1] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*, 2020.

[2] Hervé Bredin, Ruiqing Yin, Juan Manuel Coria, Gregory Gelly, Pavel Korshunov, Marvin Lavechin, Diego Fustes, Hadrien Titeux, Wassim Bouaziz, and Marie-Philippe Gill. pyannote.audio: neural building blocks for speaker diarization. In *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 2020.

[3] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

[4] John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[5] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630. IEEE, 2019.

[6] Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 46–50. IEEE, 2020.

[7] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266, 2019.

[8] Brian MacWhinney. The talkbank project. In *Creating and digitizing language corpora*, pages 163–180. Springer, 2007.

[9] Brian MacWhinney and Johannes Wagner. Transcribing, searching and data sharing: The clan software and the talkbank data repository. *Gesprachsforschung: Online-Zeitschrift zur verbalen Interaktion*, 11:154, 2010.

[10] Ethan Manilow, Prem Seetharman, and Justin Salamon. *Open Source Tools & Data for Music Source Separation.* https://source-separation.github.io/tutorial, Oct. 2020.

[11] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028. IEEE, 2018.

[12] Julius Orion Smith. *Spectral audio signal processing.* W3K, 2011.

[13] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25. IEEE, 2021.

[14] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

[15] DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726, 2018.