Desarollo de clasificador de numeros a partir de audios

por
Tomas Díaz, Nicolás Lopez y Manuel Ramirez Silva

Analisis Matematico III, tutorial Sosa - Marzik
Universidad de San Andrés

Junio 2024

Resumen

Este trabajo presenta el desarrollo y la evaluación de dos sistemas para la clasificación automática de dígitos a partir de señales de audio. El primer sistema utiliza técnicas clásicas de procesamiento de señales, como la Transformada de Fourier Discreta (DFT) y los coeficientes cepstrales en la frecuencia melódica ($MFCC$), mientras que el segundo emplea una red neuronal simple. Se compararon diferentes representaciones de las señales de audio (señal en crudo, DFT y $MFCC$) para evaluar su impacto en la precisión de los clasificadores. Los resultados mostraron que el clasificador basado en técnicas clásicas obtuvo una precisión de 17.6 % con señal en crudo, 60.0 % con DFT y 43.32 % con $MFCC$. La red neuronal, en cambio, mostró una precisión de 85 % con DFT, 23.6 % con $MFCC$ y 15.8 % con señal en crudo, subrayando que la DFT fue la técnica más adecuada para la clasificación de dígitos hablados.

1. Introducción

El reconocimiento automático de dígitos a partir de señales de audio es una aplicación práctica que nos permite profundizar en el funcionamiento del procesamiento de señales y del aprendizaje automático. Este informe se enfoca en el diseño y evaluación de dos sistemas distintos para la clasificación automática de dígitos.

La primera parte del trabajo consistirá en el desarrollo de un clasificador que utiliza métodos tradicionales de análisis de señales, como la Transformada de Fourier (DFT) y el cálculo de coeficientes cepstrales en la frecuencia melódica ($MFCC$). En esta parte, se calcularán representaciones promedio de las señales de audio para cada dígito, comparando cada nueva grabación con estas representaciones promedio para determinar el dígito pronunciamiento. La segunda parte explorará el uso de una red neuronal simple para la clasificación, evaluando cómo diferentes representaciones de las señales de audio (onda cruda, módulo de la DFT y coeficientes $MFCC$) afectan el rendimiento del modelo. Los objetivos propuestos para este informe son los siguientes:

- Desarrollar un clasificador de dígitos utilizando técnicas clásicas de procesamiento de señales, como la Transformada de Fourier Discreta (DFT) y los coeficientes cepstrales en la frecuencia melódica ($MFCC$).
- Crear un clasificador de dígitos utilizando una red neuronal simple.
- Comparar el rendimiento de ambos clasificadores utilizando diferentes representaciones de las señales de audio: señal en crudo, módulo de la DFT y coeficientes $MFCC$.
- Evaluar cómo las diferentes técnicas de representación de señales de audio afectan la precisión de los modelos de clasificación.

En este informe se presentarán los resultados obtenidos con ambos enfoques, analizando las métricas de rendimiento y discutiendo las ventajas y desventajas de cada método. El informe se estructura de la siguiente manera: en la sección *Marco Teórico* se describen los conceptos y técnicas utilizadas; en la sección *Desarrollo Experimental* se detallan los procedimientos seguidos; en la sección *Resultados y Discusión* se presentan y analizan los resultados obtenidos; una sección *Conclusión* donde se resumen los hallazgos, y por ultimo, terminamos con *Propuestas de investigación futura* donde se muestran futuras soluciones que prometen nuevas ideas.

2. Marco Teórico

En esta sección se describen los conceptos teóricos fundamentales y las técnicas utilizadas en el desarrollo del trabajo. Se explorarán dos enfoques principales: uno basado en técnicas clásicas de procesamiento de señales y otro basado en redes neuronales para la clasificación de dígitos hablados.

El procesamiento de señales de audio implica la manipulación y análisis de señales de sonido con el objetivo de extraer información relevante. Las señales de audio pueden ser analizadas tanto en el dominio del tiempo, donde una señal se representa como una función de la amplitud respecto al tiempo, como en el dominio de la frecuencia, utilizando herramientas como la Transformada de Fourier Discreta (DFT) para descomponer una señal en sus componentes de frecuencia.

2.1. Señal en Crudo

La señal en crudo se refiere a la representación original de una señal de audio, tal como se ha capturado, sin aplicar ningún tipo de procesamiento previo. Esta señal es una secuencia de muestras de amplitud respecto al tiempo, y contiene toda la información capturada por el dispositivo de grabación, incluyendo tanto las componentes útiles como el ruido. Si bien trabajar con la señal en crudo puede ser complejo debido a su alta dimensionalidad y ruido, permite a los modelos de aprendizaje automático, extraer características directamente de los datos originales. Esto puede ser beneficioso para nuestro propósito de clasificación de audio.

2.2. Transformada de Fourier Discreta (DFT)

La Transformada de Fourier Discreta (DFT) es una herramienta fundamental en el análisis de señales en el dominio de la frecuencia. La DFT permite descomponer una señal en sus componentes de frecuencia, facilitando el análisis de sus características espectrales. Esta descomposición es útil porque muchos patrones importantes en las señales de audio se manifiestan más claramente en el dominio de la frecuencia

que en el dominio del tiempo. Además, la Transformada de Fourier es capaz de reducir la dimensionalidad de los datos y eliminar redundancias, mejorando la eficiencia del procesamiento y la precisión de los modelos de clasificación. La magnitud de la DFT proporciona información sobre la intensidad de las diferentes frecuencias presentes en la señal.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \tag{1}$$

donde $X(k)$ es el coeficiente de la DFT en la frecuencia k, $x(n)$ es la señal en el dominio del tiempo, y N es el número de muestras.

2.3. Coeficientes Cepstrales en la Frecuencia Melódica ($MFCC$)

Los coeficientes cepstrales en la frecuencia melódica ($MFCC$) son una representación compacta y perceptualmente relevante de las señales de audio. Se obtienen mediante el análisis de la señal en bandas de frecuencia que imitan la percepción humana del sonido. El proceso para calcular los $MFCC$ incluye la segmentación de la señal en ventanas, la aplicación de la DFT a cada ventana, entre otras.

2.4. Diferencia de Cuadrados

Para clasificar una nueva señal de prueba, se comparó con las representaciones promedio de cada dígito utilizando la diferencia de cuadrados como métrica objetiva. La diferencia de cuadrados entre dos señales x e y se define como:

$$\text{Diferencia de Cuadrados}(x, y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{2}$$

Donde x_i y y_i son los valores de las señales en el instante de tiempo i, y N es el número total de muestras en las señales.

3. Desarrollo experimental

En esta sección, presentamos en detalle la metodología y las técnicas empleadas en el desarrollo de los clasificadores, así como el proceso de recolección de datos, entre otras cosas. Describiremos los pasos y criterios seguidos para la selección de las características más relevantes, los algoritmos de clasificación utilizados y los métodos de validación empleados para evaluar el rendimiento de los clasificadores. Además, abordaremos las fuentes de los datos recolectados, las condiciones bajo las cuales fueron obtenidos y las estrategias implementadas para garantizar la calidad y representatividad de los mismos. Esta descripción permitirá una comprensión clara de las bases experimentales y las decisiones técnicas que fundamentan nuestro trabajo.

3.1. Recolección de muestras

Para el correcto funcionamiento de los clasificadores, usamos un dataset, propuesto por la cátedra, que se llama *Audio MNIST*. El dataset Audio MNIST contiene grabaciones de audio de personas pronunciando dígitos del 0 al 9 en inglés. Cada grabación dura alrededor de un segundo y está muestreada a 8 kHz. El dataset incluye múltiples instancias por cada dígito grabadas por diferentes locutores, lo que permite evaluar la robustez de los modelos ante variaciones en la pronunciación y calidad del habla, además no contienen ruido externo por lo que la clasificación no cuenta con esos problemas. Por tanto, nos aseguramos una representación amplia de las variaciones de las muestras y una calidad suficiente para seguir con el proyecto.

Al tener una gran cantidad de muestras nos otorga una mayor capacidad de nuestro modelo y clasificadores para generalizar y predecir con precisión las grabaciones de prueba. Es decir, obtendremos resultados de clasificación más confiables y de alta calidad.

3.2. Desarrollo de clasificadores

El primer enfoque del experimento se centró en el desarrollo de un clasificador utilizando técnicas clásicas de procesamiento de señales. El objetivo fue calcular representaciones promedio de los audios de los distintos números a detectar. Se utilizaron todas las muestras de entrenamiento (train) para calcular la representación promedio de cada dígito (0-9). Para cada señal de audio, se consideraron tres

representaciones: señal en crudo, transformada de Fourier discreta (DFT) y coeficientes cepstrales en la frecuencia melódica ($MFCC$).

Antes de todo, sea cual sea el feature que vamos a implementar, normalizamos las señales para que todas la misma escala o longitud. Para la señal en crudo y el módulo de la DFT, se calcularon los promedios punto a punto de cada frecuencia. Para los $MFCC$, se calculó el promedio de los coeficientes cepstrales. Para clasificar una nueva señal de prueba, se comparó con las representaciones promedio de cada dígito utilizando la diferencia de cuadrados. El dígito con la menor diferencia con la señal de prueba fue identificado como el dígito pronunciado.

En la segunda parte del experimento, se utilizó una red neuronal simple para la clasificación de los dígitos hablados. La red fue entrenada y evaluada utilizando los datos de entrenamiento (train) y prueba (test) provistos. Se desarrollaron tres versiones del modelo, variando la representación de los datos de entrada: señal en crudo, módulo de la DFT y coeficientes $MFCC$.

3.2.1. Arquitectura de las redes neuronales

Tenemos dos modelos de redes neuronales. En primer lugar, tenemos a la red feedforward, $NeuralNet$, con capas densamente conectadas, la cual procesa señales de audio en su forma cruda o transformada (DFT). Consta de capas ocultas que varían en tamaño desde 4000 hasta 250 neuronas, y una capa de salida de 10 neuronas para la clasificación de los números. Tiene un total de 42635260 parámetros para las señales en crudo y 26639260 parámetros para DFT.

En segundo lugar, tenemos otra red feedforward, $NeuralNet\_mfcc$, que está optimizada para coeficientes cepstrales($MFCC$). Contiene capas ocultas de 320 a 12 neuronas y una capa de salida de 10 neuronas. Tiene un total de 7453 parámetros.

Finalmente, se completó el desarrollo del código necesario para procesar y cargar los datos en las tres representaciones mencionadas. Posteriormente, se llevó a cabo el entrenamiento y la evaluación del modelo mediante el análisis de los datos. Se compararon los resultados obtenidos con cada representación para determinar cuál proporcionaba mejores resultados en términos de precisión y calidad de las predicciones.

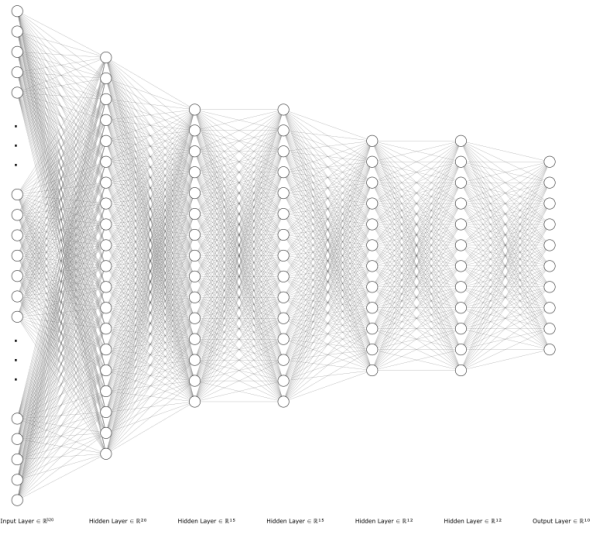

Figura 1: Architecture of MFCC's Neural Network

En la Figura (1) se presenta la arquitectura de $NeuralNet\_mfcc$, donde los puntos suspensivos representan una mayor cantidad de neuronas en la capa de entrada. La arquitectura de $NeuralNet$, como se mencionó anteriormente, contiene un gran número de neuronas por capa, lo que hace que no sea óptimo representarlo completamente en un gráfico.

4. Resultados y discusión

En esta sección, se expondrán de manera detallada los resultados obtenidos a lo largo del desarrollo de este proyecto. Además, se realizará un análisis crítico de dichos resultados, discutiendo su relevancia y las implicaciones que estos tienen en el contexto del estudio.

4.1. Resultados parte 1

Ahora, nos proponemos exponer los resultados obtenidos al aplicar distintas representaciones de la señal de audio en clasificadores de dígitos clásicos, aplicando transformaciones a las señales de audio. Se evaluaron tres metodologías para variar las entradas: la señal sin procesar, la Transformada de Fourier Discreta (DFT) y los Coeficientes Cepstrales de Frecuencia Mel (MFCC). Otra cosa a mencionar es que evaluamos la precisión tanto para el set de test como de entrenamiento para cada clasificador, reportando el rendimiento tanto con datos conocidos como desconocidos, y poder evaluar la capacidad de generalización del modelo.

Primeramente, para el clasificador mas común, donde la entrada es una señal cruda, sin ningún tipo de transformación o preprocesamiento, obtenemos los siguientes resultados. Respecto al set de entrenamiento, obtenemos una precisión del 48.6 %, mientras que para el set de entrenamiento observamos una precisión del 17.6 % sobre el total. Podemos notar que no son números que nos satisfagan (ya que si elige de manera aleatoria debería tener cerca de un 10 % de precisión). De igual manera, es un resultado esperable ya que hay muchas posibles fuentes de error para este tipo de dato. En primer lugar, el corrimiento de las señales puede llegar a ser un problema, ya que de no estar alineadas, la precisión del clasificador se puede ver afectada severamente. Entre otros, las señales generalmente aportan una cantidad de ruido significativo que genera diferencias entre las señales que el clasificador no es capaz de diferenciar, por ende, la precisión se ve gravemente disminuida.

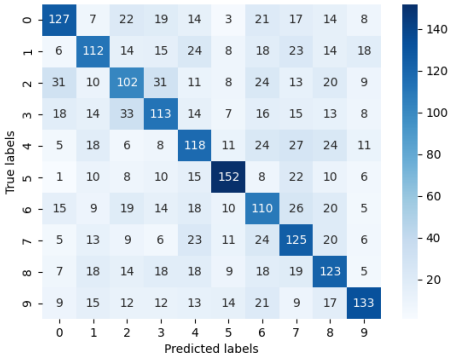

Figura 2: matriz de confusión (clasificador: promedios, set: entrenamiento)

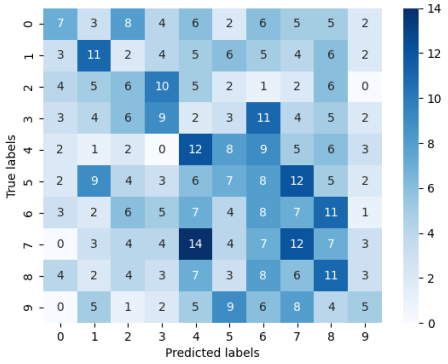

Figura 3: matriz de confusión (clasificador: promedios, set: prueba)

Al observar las matrices de confusión en las Figuras 2 y 3 del primer clasificador, podemos ver que en el conjunto de entrenamiento clasifica bastante bien, lo que indica que ha aprendido características para diferenciar los distintos audios. Sin embargo, esto no se refleja en el conjunto de prueba. Se evidencia la dificultad del clasificador para encontrar las mismas características y patrones consistentes en el nuevo conjunto de datos.

Por otro lado, nos encontramos con el segundo clasificador, en el cual calculamos la transformada de Fourier de la señal y le pasamos eso como entrada al clasificador. En este caso, tenemos una precisión de 62.52 % para el set de entrenamiento y 60 % para el set de prueba. Podemos notar que es el modelo más consistente en lo que respecta a los clasificadores, y es también el que nos va a generar la mejor precisión de los tres.

Esto se debe a que, al descomponer una señal en sus componentes de frecuencia, esta tiene una representación en el dominio de la frecuencia que puede destaca los componentes de frecuencia de la señal, lo que facilita la identificación de patrones que son críticos para la clasificación y puede que no sean tan evidentes en el dominio del tiempo, especialmente en señales complejas y ruidosas. Asimismo, al trabajar en el dominio de la frecuencia, el modelo maneja mejor las variaciones y el ruido presentes en las señales, lo que contribuye a una mayor robustez y estabilidad en la clasificación. Esto lo podemos verificar si nos fijamos en ambas matrices de confusión del clasificador (Figuras 4 y 5), podemos notar que mantienen una proporcionalidad no solo en los casos de éxito sino también en cada uno de los casos de fracaso. Esto es algo que no notamos en el primer clasificador ya que los errores se distribuyen de forma mucho mas aleatoria.

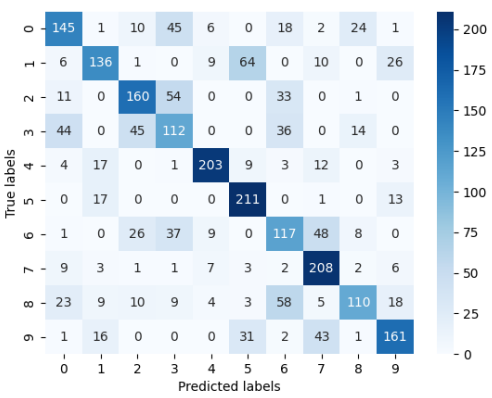

Figura 4: Matriz de confusión (clasificador: DFT, set: entrenamiento)

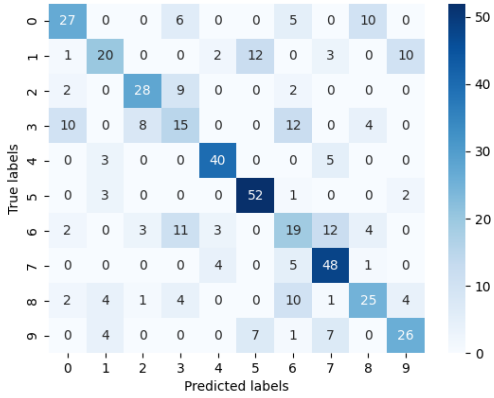

Figura 5: Matriz de confusión (clasificador: DFT, set: entrenamiento)

Además, como se menciona en el marco teórico, la DFT reduce la dimensionalidad de la señal de forma significativa, lo que puede mejorar la eficiencia del clasificador y reducir el sobreajuste, proporcionando así un equilibrio óptimo entre la precisión en los conjuntos de entrenamiento y prueba. Esta capacidad de la DFT para extraer características mas robustas explica por qué este clasificador logra la mejor precisión y consistencia entre los modelos evaluados.

Por ultimo, nos remontamos al clasificador de coeficientes ceptrales (*mfcc*). Para este ultimo caso, podemos reportar una precisión del 43.32 % y un 43.4 % para el set de entrenamiento y test respectivamente. Ya de por si podemos notar que es el único clasificador de los tres que mejora su precisión para el set de test (en comparación con el del entrenamiento). Asimismo, también es el que tiene la menor diferencia entre las precisiones para ambos conjuntos de datos.

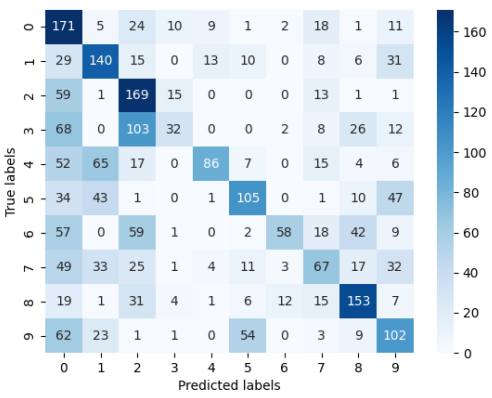

Figura 6: Matriz de confusión (clasificador: MFCC, set: entrenamiento)

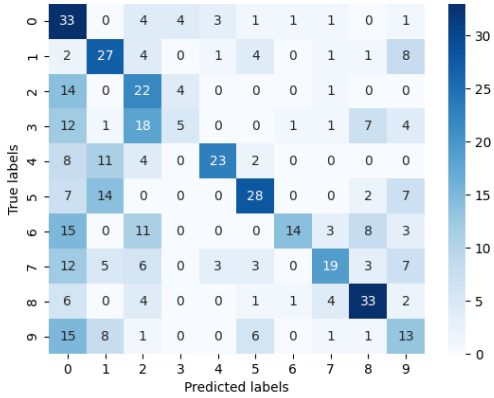

Figura 7: Matriz de confusión (clasificador: MFCC, set: entrenamiento)

A partir de sus matrices de confusión (Figuras 6 y 7) podemos ver también que, al igual que en el caso del clasificador del DFT, existe una cierta proporcionalidad en la gran mayoría de los puntos de la matriz, tanto en los casos de predicciones correctas como en los casos de desaciertos. Si bien no parece ser tan idénticas como las de la DFT, a grandes rasgos mantienen el mismo criterio, que habla sobre la capacidad de generalización del clasificador para conjuntos de datos que desconocemos.

Otra cosa que podemos observar de las matrices de confusión de este clasificador es como tiene una gran precisión para números que están en el medio de nuestro dominio (números como el 4,5 o 6), mientras que los números mas extremos se reflejan una mayor cantidad de error. Esto puede ocurrir porque los dígitos en el medio del rango pueden tener características acústicas más distintivas y consistentes, facilitando su identificación. Mientras que, los dígitos en los extremos (como el 0, 1, 2) pueden tener características más similares entre sí o ser pronunciados con más variabilidad, lo que dificulta al clasificador la extracción de patrones consistentes con los MFCC.

Adicionalmente, para este clasificador decidimos también variar la cantidad de coeficientes que tenia cada señal para ver como afectaba en el rendimiento del clasificador. Los resultados que obtuvimos se

muestran en la figura 8, en donde podemos ver una especie de convergencia a partir de los 20 coeficientes (tal vez este no sea el caso del set de entrenamiento), que es generalmente el numero predeterminado de coeficientes para aplicar $mfcc$. Además, es necesario destacar que es el algoritmo que tiene mas tiempo de ejecución de los tres, y la diferencia es bastante significativa (ver Figura 9). Por lo tanto, al ver que convergía la precisión del modelo y que nos era bastante costoso computacionalmente seguir con el desarrollo, no vimos necesario indagar mas.

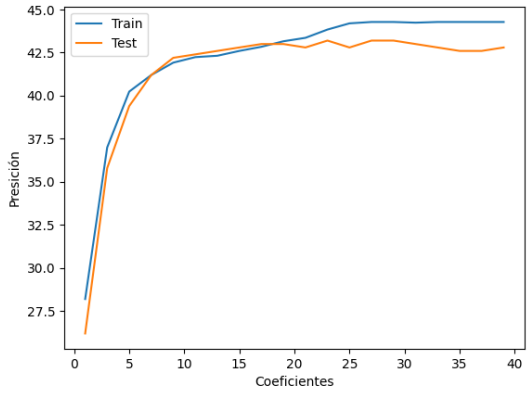

Figura 8: Precisión del modelo $mfcc$ para distinta cantidad de coeficientes

	Entrenamiento	Prueba
Señal cruda	1.39 s	0.78 s
DFT	1.11 s	0.23 s
MFCC	15.05 s	2.78 s

Figura 9: Tiempo de ejecución para cada clasificador (Python v. 3.12.2)

Aparte de las características particulares que describimos de cada clasificador, encontramos problemas naturales a todos los clasificadores que afectaban directamente a su precisión. El caso más evidente fue el de la normalización de las señales. En la figuras 10 y 11, mostramos la precisión de ambos sets (set de prueba y set de entrenamiento) para señales normalizadas y no normalizadas. Como podemos observar, si bien no vemos cambios muy notorios en la parte del set de entrenamiento, podemos notar que la precisión baja drásticamente en el set de prueba cuando no aplicamos la normalización de señales. Podríamos considerar que el modelo aprende mal y se aprende únicamente las señales que se le da, pero al darle señales con distinto largo de las que se le entrenó da una precisión muy baja; esto se podría considerar una especie de sobreajuste de nuestro modelo.

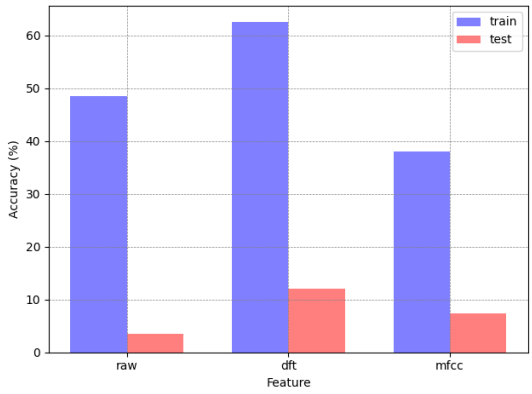

Figura 10: Precisión de señales no normalizadas

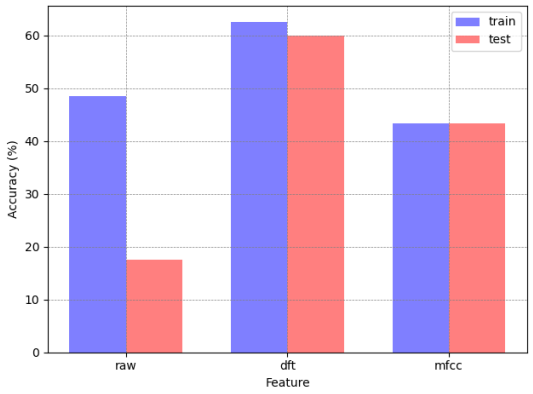

Figura 11: Precisión para señales normalizadas

Esta baja precisión de los clasificadores se debe a que no hicimos previamente una normalización de nuestras señales antes de calcular cualquiera de estos clasificadores. La normalización asegura que las señales tengan la misma escala o longitud, lo cual es crucial para compararlas de manera significativa y para aplicar operaciones consistentes como el promedio o la transformada de Fourier. En nuestro caso, al normalizar la longitud de las señales antes de calcular el promedio (raw), la transformada de Fourier (dft), o los coeficientes cepstrales de frecuencia mel (mfcc), te aseguras de que todas las señales contribuyan de manera equitativa a los resultados finales. Esto es esencial para obtener análisis precisos y comparables, garantizando así que los resultados reflejen correctamente las características de las señales en estudio y permitan una clasificación efectiva en tu aplicación.

4.2. Resultados parte 2

En esta sección se presentan los resultados obtenidos al aplicar diferentes representaciones de la señal de audio en redes neuronales para la clasificación de dígitos hablados. Se evaluaron tres enfoques: señal en crudo, Transformada de Fourier Discreta (DFT) y Coeficientes Cepstrales de Frecuencia Mel (MFCC).

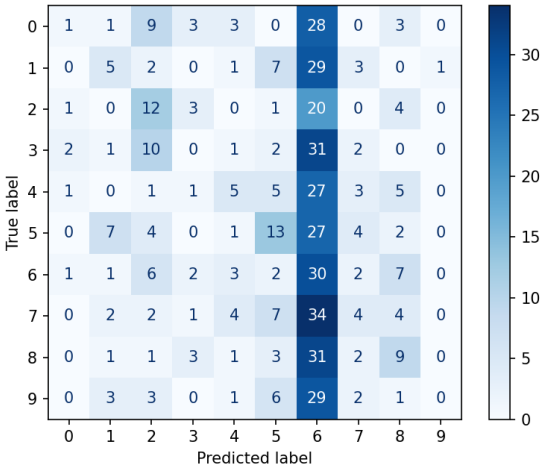

Figura 12: Matriz de confusión para la red neuronal utilizando la señal en crudo

La red neuronal que utilizó la señal en crudo mostró una precisión del 15.8 %. La matriz de confusión correspondiente (Figura 12) indica una tendencia a clasificar la mayoría de los dígitos como el número 6.

Como podemos ver, la señal en crudo, aunque rica en información, requiere una red neuronal más sofisticada o preprocesamiento adicional para mejorar su desempeño, como se observa en su baja precisión del 15.8 %. La matriz de confusión indica el sesgo significativo hacia el número '6', posiblemente debido a que las señales de audio no comienzan en el mismo instante de tiempo, la estructura de la red no es adecuada para este tipo de datos o el número 6 tiene características similares a los otros dígitos.

En cambio, la red neuronal que utilizó la DFT logró una precisión del 85 %, mostrando un rendimiento significativamente mejor. La matriz de confusión correspondiente (Figura 13) muestra que la mayoría de los dígitos fueron clasificados correctamente.

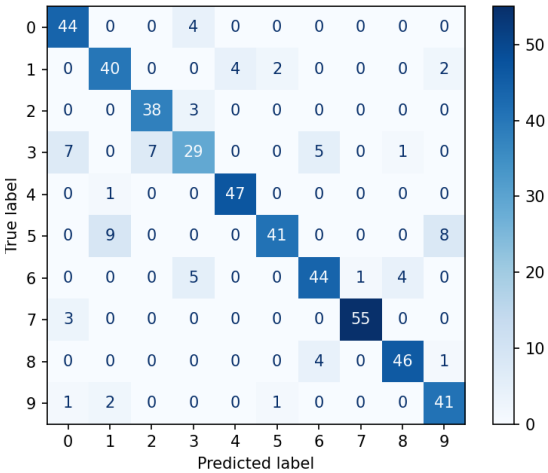

Figura 13: Matriz de confusión para la red neuronal utilizando DFT

La DFT, al descomponer la señal en sus componentes de frecuencia, proporciona una representación más adecuada para este problema específico, logrando una precisión que obtuvo. La matriz de confusión muestra que el modelo puede identificar correctamente la mayoría de los dígitos, destacando la eficacia de la DFT para capturar características relevantes para la clasificación de dígitos. Este resultado se debe a que la DFT descompone la señal en sus componentes de frecuencia, resaltando características distintivas de cada dígito y reduciendo el ruido y las variaciones temporales.

Es importante destacar que aunque ambos modelos utilizan la misma arquitectura, NeuralNet, la gran diferencia entre la precisión de la DFT y la señal en crudo detalla la importancia del preprocesamiento

adecuado de los datos de entrada. La representación más estructurada y significativa de las características a través de la DFT facilita un aprendizaje más afectivo por parte del modelo.

Por último, la red neuronal que utilizó los MFCC tuvo una precisión del 23.6 %. La matriz de confusión correspondiente (Figura 14) revela una alta tasa de errores en la clasificación.

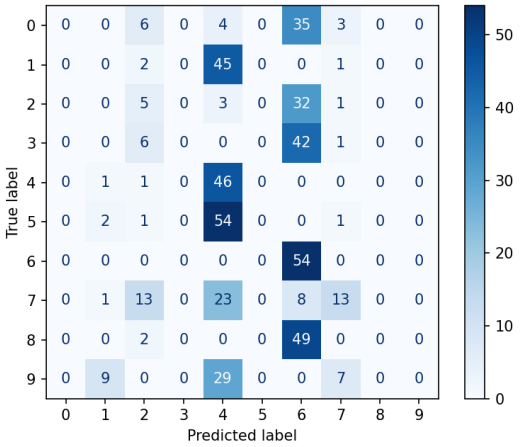

Figura 14: Matriz de confusión para la red neuronal utilizando MFCC

Los MFCC, aunque son una representación avanzada y optimizada del sonido, no resultaron ser efectivos para la clasificación de dígitos, con una precisión del 23.6 %. La matriz de confusión revela que el modelo tiende a predecir mayormente los dígitos '4' y '6', lo cual claramente afecta el rendimiento del clasificador. Esta limitación se debe a que los MFCC están más orientados a la identificación de hablantes, capturando características de la envolvente espectral que son más relevantes para distinguir entre voces que para diferenciar entre distintos dígitos.

En resumen, la elección de la representación de la señal es crucial para el rendimiento de las redes neuronales en la clasificación de dígitos hablados. La DFT se destacó como la técnica más adecuada en este contexto específico, mientras que la señal en crudo y los MFCC requieren ajustes adicionales para alcanzar un rendimiento óptimo.

5. Conclusión

En el informe exploramos dos enfoques diferentes para la clasificación automática de dígitos hablados, uno basado en técnicas clásicas de procesamiento de señales y otro utilizando redes neuronales.

En la primera parte, se evaluaron clasificadores clásicos aplicando transformaciones a las señales de audio, tales como la señal sin procesar, la Transformada de Fourier Discreta (DFT) y los Coeficientes Cepstrales de Frecuencia Mel ($MFCC$). Los resultados mostraron que la DFT fue la técnica más efectiva, logrando una precisión del 60 % en el conjunto de prueba, mientras que la señal en crudo y los $MFCC$ obtuvieron precisiones significativamente menores, 17.6 % y 23.6 % respectivamente. La DFT mostró ser superior debido a su capacidad para destacar los componentes de frecuencia de la señal, facilitando la identificación de patrones importantes para la clasificación. En el caso de la señal en crudo, los errores se distribuyeron de manera aleatoria, reflejando la dificultad del clasificador para encontrar patrones consistentes. Para los $MFCC$, a pesar de variar las propiedades como el número de coeficientes, la precisión no mostró mejoras significativas, ya que convergió sin incrementar su rendimiento notablemente.

En la segunda parte, se implementaron redes neuronales simples para la clasificación, evaluando cómo diferentes representaciones de las señales de audio afectan el rendimiento del modelo. La DFT demostró ser la técnica más efectiva, alcanzando una precisión del 85 %, mientras que la señal en crudo obtuvo un rendimiento significativamente inferior (15.8 %) y los $MFCC$ mostraron una precisión del 23.6 %. Estas diferencias resaltan la importancia de seleccionar una representación adecuada de la señal para mejorar el rendimiento del clasificador.

Además, el aprendizaje automático desempeña un papel fundamental al proporcionar opciones para el procesamiento y análisis avanzado de señales, lo cual es crucial y lo convierte en una optativa muy valiosa para explorar el potencial de las señales en diversas aplicaciones prácticas.

En conclusión, la elección de la representación de la señal es crucial para el éxito de los modelos de clasificación de dígitos hablados. Las técnicas clásicas como la DFT ofrecen una representación estructurada y efectiva de las características de la señal, facilitando un aprendizaje más efectivo por parte de los modelos. La DFT, al descomponer la señal en sus componentes de frecuencia, no solo resalta las

características distintivas de cada dígito, sino que también ayuda a eliminar el ruido y las variaciones temporales, proporcionando una base más clara y precisa para la clasificación.

6. Propuestas de investigación futura

Si bien los métodos empleados son dentro de todo efectivos, acá dejamos ideas que consideramos que pueden llegar a ser una mejor solución a este problema.

6.1. Alineamiento de señales

El alineamiento temporal de señales es un método de preprocesamiento de señales que se utiliza para que estas puedan ser comparadas, o combinadas, sin tener un desfasaje temporal.

Para lograr esto, se suele calcular la convolución lineal y la correlación cruzada entre las señales, para así obtener el desfasaje entre las señales. Una vez calculado el desfasaje, se realiza un corrimiento de la señal con desfasaje para poder sincronizar los periodos temporales.

Esta herramienta podría ser utilizada en nuestro trabajo para poder obtener mejores resultados al comparar los audios. Puesto que, aunque todas las señales tengan una duración de 1 segundo, no todas comienzan cuando inicia el audio. Entonces, intentamos aplicar este método pero no nos funciono, pero si aplicásemos una alineación de las señales, obtendríamos comparaciones más precisas y, en consecuencia, obtendríamos un modelo con mayor accuracy.

6.2. Huellas Digitales y Patrones Espectrales

Además, otro algoritmo que se podría usar para la clasificación de números del 0 al 9 a partir de archivos de audio sería el algoritmo basado en la extracción de características del espectrograma del audio o patrones -huellas digitales- relevantes para clasificar correctamente los dígitos hablados. Este reconocimiento de patrones se podría realizar mediante redes neuronales profundas, específicamente utilizando modelos de redes neuronales convoluciones (CNN), que permiten procesar y detectar eficazmente patrones en los espectrogramas.

Este algoritmo, que es el que desarrollo Shazam para identificar canciones, transforma fragmentos de audio en "huellas digitales"que pueden compararse con una vasta base de datos. El proceso comienza con el usuario grabando un audio del ambiente y este audio se convierte del dominio temporal al dominio frecuencial mediante la Transformada Rápida de Fourier [1].

Una vez transformado el audio, el algoritmo detecta los picos más prominentes en el espectrograma del audio, conocidos como *"landmarks"*. Estos picos se eligen por su robustez frente a ruidos y distorsiones, asegurando que solo se seleccionen aquellos con mayor contenido energético en su vecindad [2]. Esta selección de picos es crucial, ya que permite que las características relevantes del audio sobrevivan incluso en condiciones de ruido significativo.

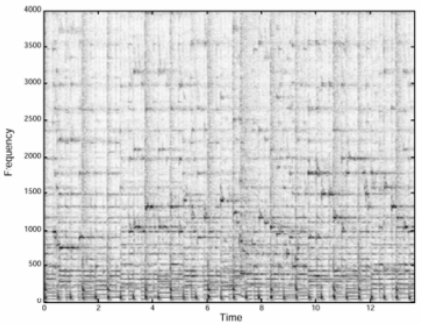

Figura 15: Espectrograma

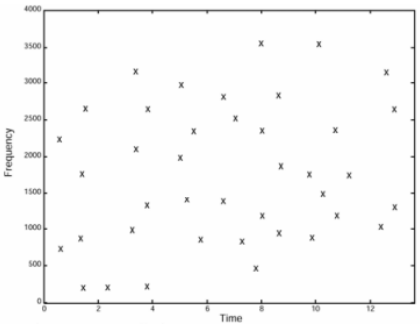

Figura 16: Mapa de constelación

A partir de los landmarks, se crean hashes combinando pares de puntos de tiempo-frecuencia cercanos. Cada hash es una representación única del audio y se almacena junto con un identificador de tiempo. Esto produce una *"mapa de constelación"* que facilita la comparación rápida y eficiente [1]. La combinación de puntos en los hashes mejora la especificidad del proceso, reduciendo la probabilidad de coincidencias falsas. De esta forma reducimos un espectrograma entero a un conjunto de coordenadas.

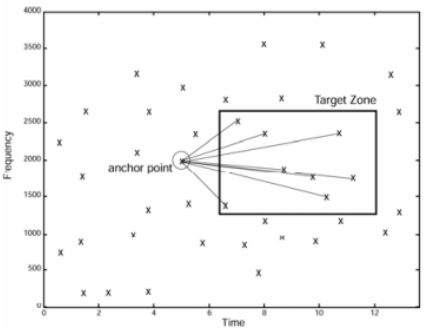

Figura 17: Espectrograma

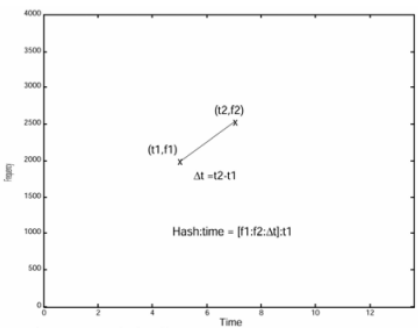

Figura 18: Mapa de constelación

El siguiente paso es la comparación y búsqueda en la base de datos. El fragmento de audio grabado se procesa para generar hashes, que luego se comparan con los almacenados en la base de datos. Esta comparación busca coincidencias significativas en los intervalos de tiempo relativos entre los hashes del fragmento grabado y los de la base de datos [1]. Este método permite una búsqueda rápida y precisa, incluso en bases de datos muy grandes.

Finalmente, la coincidencia se verifica analizando la distribución temporal de los hashes coincidentes. Si los hashes del fragmento y los de la base de datos coinciden en sus intervalos de tiempo relativos, se confirma la correspondencia. Esta técnica permite identificar canciones incluso si el audio de entrada está distorsionado o tiene ruido de fondo [2]. Por ejemplo, en la figura 19 vemos un scatterplot de tiempos del database versus tiempos del sample para un track no coincidente y su histograma de valores δ_{tk}. Por otro lado, la figua 20 muestran un scatterplot similar, pero para un track coincidente, junto con su histograma de valores δ_{tk}, destacando una línea diagonal de coincidencias, y un offset que tiene la grabacion.

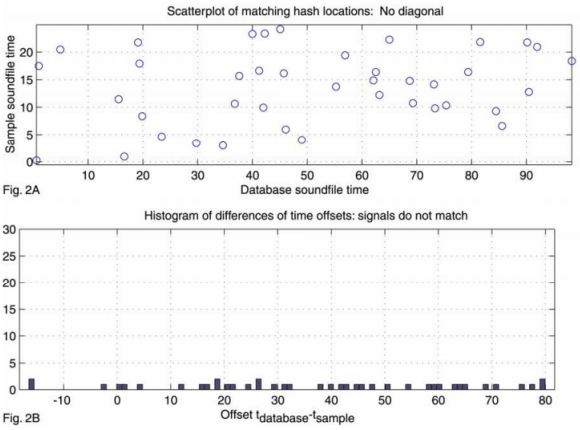

Figura 19: Espectrograma

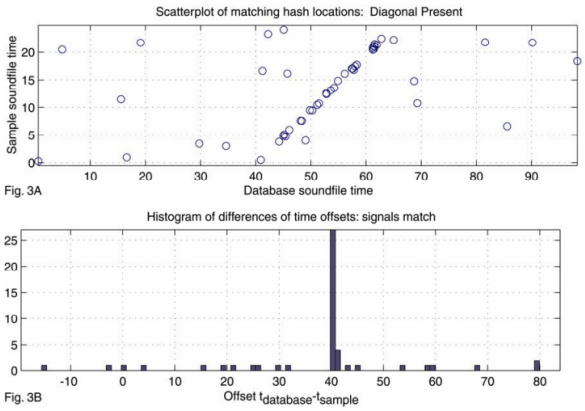

Figura 20: Gráfico de correlación del database vs. samples (arriba), gráfico del offset (abajo)

La eficacia del algoritmo se ve reforzada por su capacidad para manejar altos niveles de ruido y distorsión, manteniendo una alta tasa de acierto en la identificación de canciones. Si bien esto no es un problema del dataset, si se piensa en un proyecto a gran escala es una buena medida a tener en cuenta.

7. Bibliografia

[1] Li-Chun Wang, Avery. An Industrial-Strength Audio Search Algorithm, www.ee.columbia.edu/ dpwe/papers/Wang0 shazam.pdf. Accessed 18 June 2024.

[2] Unknown, Author. “Audio Fingerprinting: ‘Shazam It!’” Audio Fingerprinting: “Shazam It!,” 29th October 2015, serensweekly.blogspot.com/2015/10/audio-fingerprinting-shazam-it.html.