Evidencia 1. Situación problema

Alan Rivera Licea - A01412440
Karla Andrea Palma Villanueva - A01754270
Lorena Citlalli Loera Galeana - A00833159
José Manuel Armendáriz Mena - A01197583
Adrian Pineda Sánchez- A00834710

03 de Diciembre 2022

Resumen

Desarrollo de una app que permita identificar canciones a través de un fragmento de ella utilizando algoritmos basados en un análisis espectral de las canciones. La finalidad de este proyecto es monitorear transmisiones de radio, ya sea analógico o digital, para validar que no se estén violando los derechos de autor.

1. Proceso de identificación de señales y como descomponerlas

El procesamiento de señales se refiere a la ransformación de datos de tal forma que se pueda visualizar cosas en ellos que no son posibles a través de una observación directa. Hoy en día, el procesamiento de señales digitales se realiza principalmente en software y puede ejecutarse en el procesador o la tarjeta gráfica de una computadora de escritorio o en un dispositivo inteligente.

Muchas técnicas de procesamiento de señal están dirigidas hacia situaciones específicas, sin embargo, existen de igual forma técnicas básicas, las cuales se describen a continuación:

- **Filtración**: Elegir el prototipo de filtro que mejor funcione para el tipo de señales que se procesan.
- **Análisis FFT**: Captura ventanas de tiempo cortas y las convierta al dominio de la frecuencia.
- **Análisis modal**: Determina las frecuencis naturales, relaciones de amortiguamiento y las formas modales de cualquier estructura.

2. Transformada discreta de Fourier

La herramienta matemática que nos posibilita describir una función periódica es la transformada de Fourier. El propósito es sintetizar un este tipo de funciones como suma de funciones armónicas. Esta suma debe dar cero fuera del intervalo de duración de la señal, e igual a ella dentro del intervalo.

La Transformada Discreta de Fourier es el equivalente a la Transformada continua de Fourier. Solo que la DFT considera los datos separados por un tiempo T. Por lo tanto tiene la forma:

$$A_k = \sum a_n e^{-jwkT} \tag{1}$$

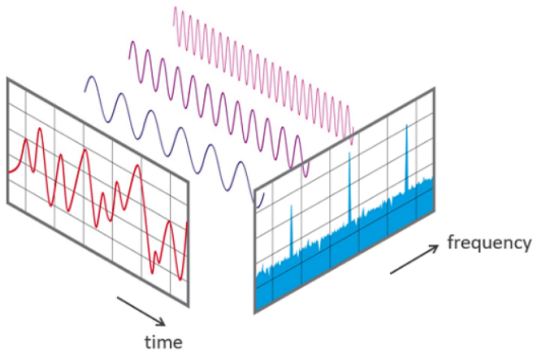
Considerando esta formula, la DFT también se puede representar de forma matricial:

$$\begin{bmatrix} F_1 \\ F_2 \\ \cdot \\ \cdot \\ \cdot \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W & W^2 & W^{n-1} \\ 1 & W^2 & W^4 & W^{n-2} \\ 1 & W^3 & W^6 & W^{n-3} \\ 1 & W^{n-1} & W^{n-2} & W^{n-3} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ \cdot \\ \cdot \\ f_{n-1} \end{bmatrix}$$

2.1. Transformación rápida de Fourier

Método de medición que descompone una señal en sus componentes espectrales individuales y así proporciona información; se utilizan para el análisis de errores, control de calidad y monitorización de las condiciones de los sistemas.

En esta descomposición, se encuentran oscilaciones sinusoidales simples que pasaron a ser frecuencias directas, en las cuales la FFT, determinará la que sea de mayor magnitud.



Anteriormente se introdujo la representación de la Transformada Discreta de Fourier de forma matricial. Si se quisieran calcular los coeficientes A_o y A_1 se obtendría que:

$$A_o = a_o + a_1$$
$$A_1 = a_o - a_1$$

Y para cuatro puntos se tendría:

$$A_o = (a_o + a_2) + (a_1 + a_3)$$
$$A_1 = (a_o - a_2) - i(a_1 - a_3)$$
$$A_2 = (a_o + a_2) - (a_1 + a_3)$$
$$A_3 = (a_o - a_2) + i(a_1 - a_3)$$

Entonces las operaciones se pueden resumir en un diagrama:

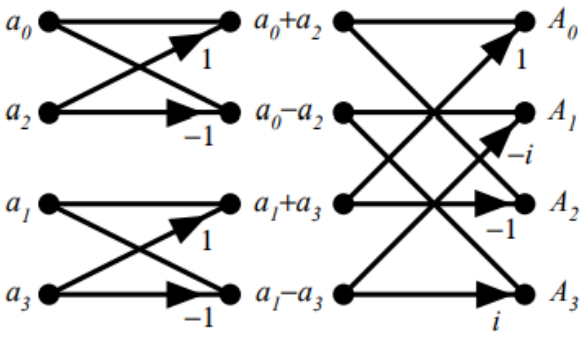


Figura 1: De https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf

El algoritmo FFT reduce el orden de las operaciones que se deben llevar a cabo de $O(N^2)$ a $O(NlogN)$ El algoritmo cambia el orden de los coeficientes para que las operaciones reduzcan el número de operaciones que se deben llevar a cabo. Por lo que el nuevo diagrama tendría la siguiente forma:

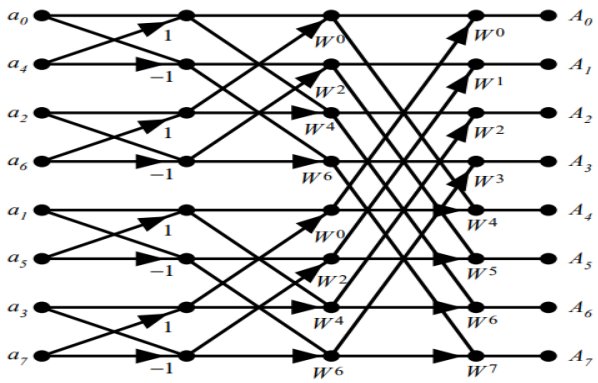


Figura 2: De: https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf

3. Espectrogramas

Se muestran los espectrogramas de las cinco canciones seleccionadas:

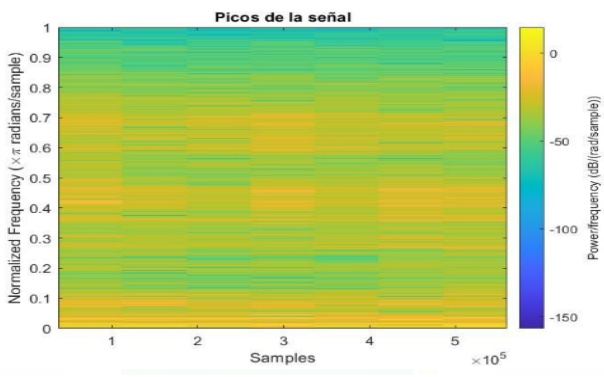


Figura 3: ChickenTeriyacki

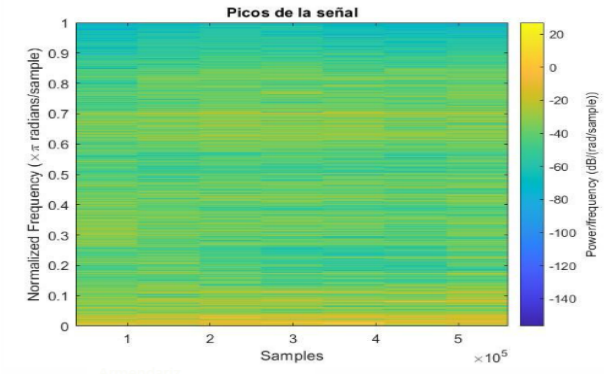


Figura 4: Test & Recognise

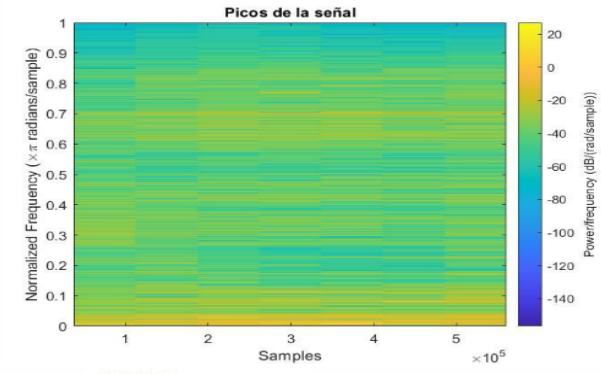


Figura 5: We Don't Believe What's on TV

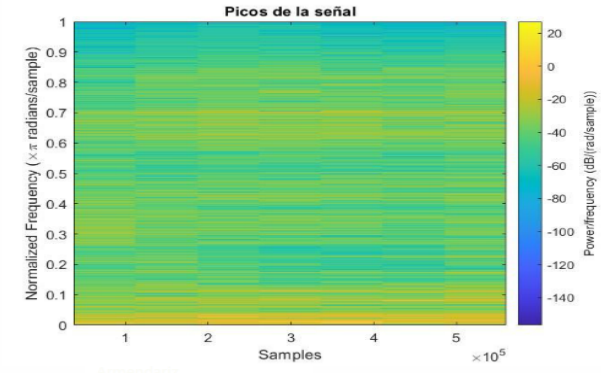


Figura 6: Premieré Gymnopédie

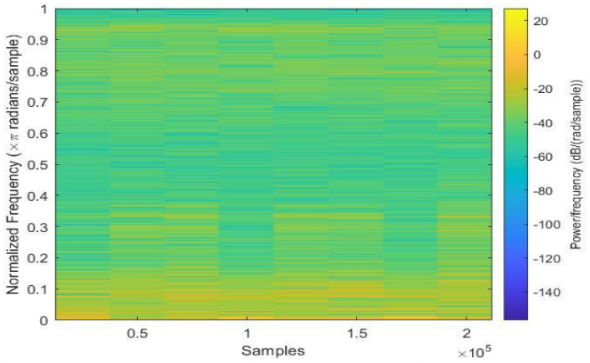


Figura 7: Ojitos lindos

4. Grabar audio

Por medio de la seccion del código $grabacion_audio$ se desarrolló la metodologia del audio para detectar el tipo de canción que se esta captando por medio del dispositivo de nuestro microfono del propio computador, se realizan una serie de muestras como: un muestreo, nuestra preferencia de calidad de procesamiento a traves del espacio de memoria guardado a traves del buffering por medio de los bits utilizados, entre otras secciones. Para grabar el audio se utiliaron las funciónes de MATLAB audiorecorder y record blocking (para pedirle al usuaraio de reproduzca la canción) y después se transformó el archivo en una señal con la función *getaudiodata* para poder trabajar con la información. Dentro de esta mismam función, se agrega el camino en el que las canciones que funcionaran cómo base de datos estan guardadas, aquí mismo se convierten a estas canciones en una tabla con toda la informacipon necesaria para leerlas.

5. Hash

Para crear esta tabla o base de datos, del camino en donde se han guardado los archivos .mp3 se obtienen las canciones y se utiliza la función *structure2table* en matlab.En esta misma parte, se cambian los archivos a formato .mat, al final se obtendra una lista de valores.

6. Transformada de Fourier y Normalización

De la señal obtenida por medio de la grabacion de nuestro dispositivo procedemos a realizar una Transformada de Fourier (con el fin de pasar de un dominio tiempo a un espacio de frecuencias). Se realiza una normalización de la señal debido a ciertas interferencias en la captacion de la señal por medio de nuestro dispositivo con respecto a los audios originales de las canciones guardadas.

Para obtener la Transformada Rápida de Fourier se hace uso de la función ya implementada en matlab fft en la señal que se obtuvo al grabar la canción. Y, como se mencionó con anterioridad, se normaliza, para ello, primero se toma una muestra de todos los datos obtenidos de la grabación, ya que se está trabajando con una muestra más pequeña, el tiempo en el que se ejecutaran las operaciones será menor con el fin de eficientar el procesamiento, esta muestra se le toma el máximo local y se divide la señal transformada entre el máximo local respectivo para normalizar toda la señal y que sea adecuada para el analisis con respecto de las señales originales de las canciones ya almacenadas.

El código devuelve dos gráficas o diagramas. El espectrograma sale con la función *spectrogram* en matlab, y la transformada normalizada se obtiene haciendo un plot de la canción ya procesada, en este caso la señal con la Transformada de Fourier se guarda en la variable SF, por lo que la gráfica se obtiene con $plot(SF)$.

La metodología dentro del análisis del código se centra en la detección de máximos locales con el fin de eficientar el tiempo de respuesta y la presición en cuestión de nuestras variables utilizadas.//

Esto se logra a traves de la deteccion de máximos locales en ciertos puntos del espectograma que podemos ir guardando cada parte de la información capturada de la señal respectiva para posteriormente señalarlos como puntos significativos del programa, a través de esto podemos agruparlas como una combinación lineal de vectores y hacer la comparación individual de las frecuencias con el intervalo temporal que separa cada una de ellas al analizarlas en torno a un conjunto de frecuencias seguidas. Ahora el resultado obtenido puede ser comparable debido a que el nombre Referenceza actúa como un identificador de la señal estudiada anteriormente, funcionando como una lista de listas (o matriz) de los valores frecuenciales así como del delta o intervalo temporal existente entre cada una de ellas con respecto a la frecuencia anterior y próxima.

7. Datos Almacenados, Identificación de la canción en conjunto

Por medio de la creación de nuestro identificador en formato de matriz en el cual almacenamos tanto los valores frecuenciales de la señal así como los intervalos temporales con respecto a sus frecuencias continuas en la funcion $hash.m$ (la cual es similar a nuestra función de $Graficas.m$ con la excepción del despliegue de las mismas).

Nuestra matriz comparativa es exportada con respecto a un nuevo nombre por medio de la canción procesada, de esta forma se crea un directorio o banco comparativo donde el identificador pasará a ser el nombre de la canción guardada para ser analizado con respecto de la señal recibida por el micrófono. Se realiza una comparación por medio de la sección $ismembertol$, identificando si la $referencia$ se encuentra en la canción analizada, posteriormente a través de un intervalo de confianza (debido a que la obtención de los valores pueden fluctuar con respecto de los originales por distintos motivos como: sonidos de fondo, interferencias o sonidos atípicos, etc.) en cuestión de la comparativa entre los vectores individuales analizados de la matriz identificador de cada canción que lleva su nombre respectivo, tanto del intervalo o distancia temporal de las frecuencias así como los valores frecuenciales de las mismas a través de la obtención de un 1 como verdadero y un 0 como falso en cuestión de la comparativa realizada a identificar el parecido y finalmente el análisis con respecto a obtener un coeficiente de comparación en cada uno a través de la obtención de la media individual que será almacenado dentro de la variable *comparación* y posteriormente se hará un análisis obteniendo con respecto de la que obtenga el coeficiente de valor máximo (en torno a la media de los valores boolenanos promediados) arrojando la canción con mayor parecido.

8. Gráficas Frecuencias

Fourier normalizado:

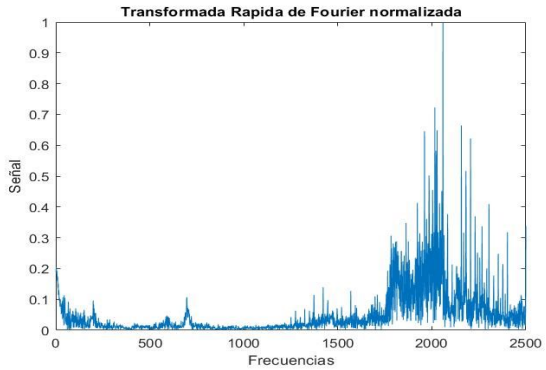


Figura 8: ChickenTeriyacki

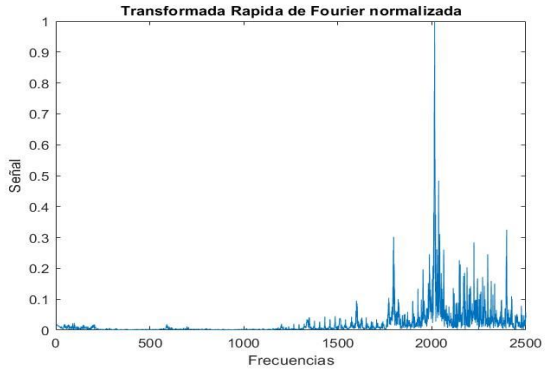


Figura 9: Test & Recognise

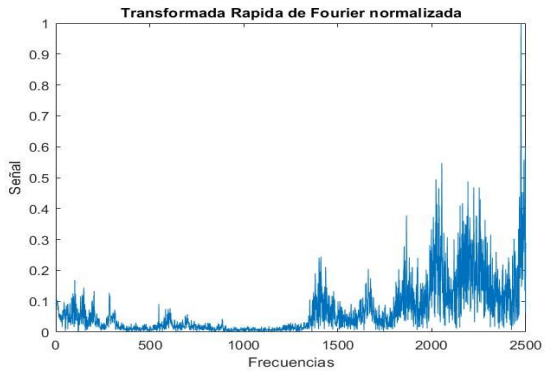


Figura 10: We Don't Believe What's on TV

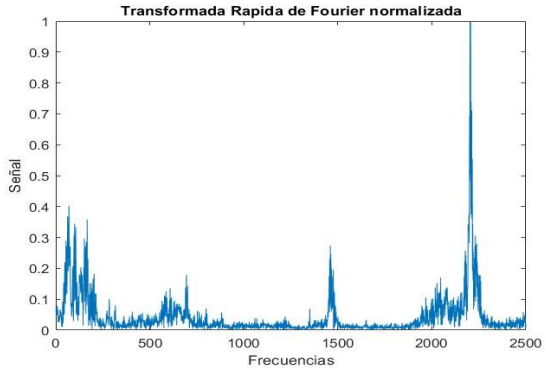


Figura 11: Premieré Gymnopédie

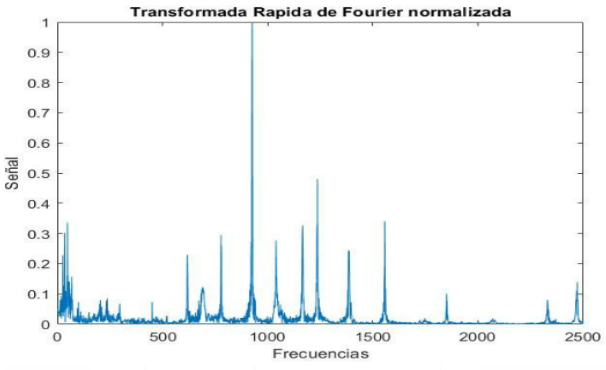


Figura 12: Ojitos lindos

9. Conclusiones

A pesar de que el programa es capaz de identificar canciones, está muy lejos de la aplicación con el enfoque en este proyecto final, en el cual se inspira el código: "shazam.^estoto se debe a que el programa cuenta con ciertas limitaciones. Comenzando por la cantidad de elementos en la base de datos (que es de cinco canciones) y el peso que involucra cada canción. Además, la Transformada Rápida de Fourier que se implementa en matlab solo funciona con ciertos intervalos de tiempo muy cortos. Para que nuestro sistema sea más parecido al de shazam lo más probable es que se necesite el uso de redes neuronales, para la implementación de cuestiones de machine learning e IA para enseñarle al código a distinguir entre canciones con mayor precisión y eficiencia, incluso si estas son muy parecidas entre si. Aún así, la FFT de Fourier resulta útil para separar una señal en sus frecuencias e identificar cómo está compuesta la canción que se grabó, también resulta util debido a que cambia de un dominio temporal, a uno de frecuencias, por lo que es más facil realizar la comparativa de la información de las canciones. Consideramos como grupo que esto será útil en otras asignaturas en el futuro. Como tal, la transformada de Fourier resulta eficiente para leer información sobre señales de forma más sencillas ya que el teorema del muestreo nos dice que no es necesario tener toda la información para reconstruir de manera exacta la señal completa.

10. Anexos

$https://drive.google.com/file/d/1mwqcd_xAd4doV80sJ4gC-U6wNqFPZ1Nf/view?usp=sharing$

11. Bibliografía

FFT. (2017, 7 marzo). NTI audio. https://www.nti-audio.com/es/servicio/conocimientos/transformacion-rapida-de-fourier-fft

¿Qué es el Procesamiento de Señales? (2022, 21 febrero). DEWESoft. Recuperado 4 de diciembre de 2022, de https://dewesoft.com/es/daq/que-es-procesamiento-de-senal

University of Oxford. (s. f.). Lecture 7 - The Discrete Fourier Transform. Information Engineering. Recuperado 4 de diciembre de 2022, de https://www.robots.ox.ac.uk/ sjrob/Teaching/SP/l7.pdf

Heckbert. (1998). Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm. CMU School of Computer Science. Recuperado 4 de diciembre de 2022, de https://www.cs.cmu.edu/afs/andrew/scs/cs/15-463/2001/pub/www/notes/fourier/fourier.pdf