TRANSFORMADA DE FOURIER PARA DETECCIÓN DE BANDAS

Ángela María Cárdenas Vargas - 817008, Mariana Cruz Gómez - 817014, Isabella Jaramillo Castaño - 817029
Proyecto Final Señales y Sistemas
Ándres Marino Álvarez Meza, Ph.D.
Departamento de ingenieria eléctrica, electrónica y computación
Universidad Nacional de Colombia - Sede Manizales

***Resumen*—El presente documento se muestra la detección de contenido mediante la identificación de canciones de una banda seleccionada usando los espectros de frecuencias de las mismas previamente recolectados en una base de datos.**

Palabras clave: Fourier, espectro, python, compilación, detección.

I. INTRODUCCIÓN

En el siguiente informe se explica el proceso empleado para la creación de un programa que pretende analizar e identificar si una canción pertenece a un artista o banda en particular, esto mediante la transformada de Fourier para el análisis espectral de las canciones, las cuales se implementan para formar una base de datos, con esta base se analizan nuevas canciones mediante la distancia de los datos obtenidos. El código implementado busca a través del lenguaje de programación python desarrollar un programa de identificación.

Como artista se seleccionó la banda DANSU, clasificada en el género de electro-pop, a partir de eso se presenta una base de datos que contiene segmentos de todas las canciones de dicha banda junto con otras de diferentes géneros, con la intención de compararlas con un segmento nuevo y con la información almacenada se pueda identificar si pertenece al artista y de ser así determinar qué canción es.

II. MARCO DE TEÓRICO

1. Transformada de Fourier: La transformada rápida de Fourier (TRF) es un método matemático para la transformación de una función del tiempo en una función de la frecuencia. A veces se describe como la transformación del dominio del tiempo al dominio de frecuencia. Es muy útil para el análisis de los fenómenos dependientes del tiempo. Una aplicación importante se da en el análisis del sonido. Es importante evaluar la distribución de frecuencias de la energía que transmite un sonido, porque el oido humano ejerce tal capacidad en el proceso de audición.[1]

2. Distancia euclídea: Se trata de una función no negativa usada en diversos contextos para calcular la distancia entre dos puntos , primero en el plano y luego en el espacio. También sirve para definir la distancia entre dos puntos en otros tipos de espacios de tres o más dimensiones. Y para hallar la longitud de un segmento definido por dos puntos de una recta, del plano o de espacios de mayor dimensión.[2]

III. DESARROLLO EXPERIMENTAL

1. Para el análisis computacional del espectro en frecuencias de la señal se usa la transformada rápida de Fourier, la cual maneja tiempo y frecuencia discretas, esta función permite que el análisis de la transformada discreta sea un proceso más óptimo para una cantidad grande de elementos.

2. Para formar la base de datos se buscaron 158 segmentos de canciones de 5 segundos, 126 de ellos que pertenecen a la banda y los otros a diferentes canciones de 15 géneros diversos. Después de cada segmento se obtiene su respectivo espectro, repartiendo en dos diferentes vectores cada canal que sale del espectro.

3. Con los valores de frecuencia obtenidos de cada uno de los espectros, se usa una función para calcular la distancia euclídea entre cada uno de los vectores de un respectivo canal con el nuevo segmento que se ingresa en el programa (de este se obtienen dos vectores del espectro como los de la base de datos) y estos se analizan uno a uno con los de la base de datos.

4. Después estas distancias se almacenan en dos vectores, que mediante funciones que permiten obtener el valor mínimo dentro del vector y la posición donde se encuentra, ayudan a identificar el segmento de la base de datos que más se parece al nuevo segmento analizado.

5. Con otro análisis y teniendo en cuenta que los primeros 32 segmentos de la base de datos que no pertenecen al artista, se puede hacer una comparación con la función if - else, que compara si el número obtenido de la posición de ambos vectores está entre los segmentos del artista o no, de acuerdo a eso el resultado el programa decide si pertenece o no a la banda elegida.

IV. Procedimiento

Pasos que seguimos para realizar el programa:
Nota: la librería de donde salen varias de las funciones usadas en el programa se llama Numpy, la cual está diseñada para cálculos científicos con arreglos.

1. Se ingresan a la base de datos los segmentos de las canciones, de los diferentes géneros y del artista y se les asigna un vector de frecuencias obtenidos del espectro mismo de las canciones. Con el siguiente código se convierte la canción .mp3 a .wav:
   `\command-consola = "ffmpeg -i ■nombre-in++nombre-out`
   `os.system(command-consola)`
   Siendo nombre-in la el segmento en .mp3 y el nombre-out en .wav. También se usa la librería os, que permite acceder a funcionalidades dependientes del sistema operativo, como leer y escribir archivos [3].

2. A los segmentos en .wav se les asigna en un vector que guarda la información:
   `xn,fsn=sf.read('dansu-do-do-do-16.wav')`
   `xn=xn[0:mmax,:]"`

3. `Xwn = np.fft.rfft(xpron,axis=0)` : función para obtener el espectro de la canción con la transformada rápida de Fourier, donde n es el número del segmento subido al código y reducido a 5 segundos en el programa.

4. `xwmn = (abs(Xwn[:,[0]]))`: función para obtener uno de los dos canales arrojados y guardarlo en un vector.

5. `xwmyn = np.transpose(xwmyn)`: función para transponer el vector saliente para un análisis más sencillo.

6. `dist1 = np.linalg.norm(xwmtn-xwmtnuevo)`: función que entrega la norma de un vector [4], en este caso la resta del vector formado con el espectro de un segmento fuera de la base de datos con cada uno de los vectores dentro de la base de datos (distancia euclídea), de esta forma se obtiene una comparación directa entre las canciones.
   Nota: las matrices creadas en el código tienen como finalidad ofrecer una forma más óptima de acceder a los datos de la base.

7. Con estos datos obtenidos se forma un nuevo vector al que se le aplican dos funciones que nos permiten relacionar esos datos con la base de datos:
   - `distmin = vec.min(axis=0)`: Que permite obtener el mínimo valor dentro del vector formado con las distancias.
   - `pos = vec.argmin()`: para obtener la posición del vector, lo que determina en qué parte de la base de datos se encuentra el segmento, si en los primeros segmentos que no son del artista o en los que sí.

8. Teniendo en cuenta que las posiciones de 0 a 31 de los vectores de distancias pertenecen a los segmentos que no son del artista, con la función if-else se realiza una distinción donde si al comparar el nuevo segmento y la posición del vector de distancias donde se encuentra ese mínimo se encuentra en estas posiciones el programa determina si pertenece o no al artista.

9. Si el segmento pertenece a una de las canciones del artista, con un conjunto de funciones de `if-else`, se busca que al usar ambos canales, sacando la posición donde se encuentra el mínimo valor en cada uno y al determinar que ambos tienen el mismo valor de posición, se puede clasificar de acuerdo a los números de los vectores de frecuencias que pertenecen a cada canción en particular dentro de la base de datos.

```
import os
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np

Tmax = 5
fs = 44100
mmax=Tmax*fs

nombre_in = "me_voy_de_la_casa.mp3"
nombre_out = "me_voy_de_la_casa.wav"
command_consola = "ffmpeg -i " + nombre_in +" "+ nombre_out
os.system(command_consola)

import soundfile as sf
x1, fs1 = sf.read('me_voy_de_la_casa.wav')
x1=x1[0:mmax,:]
print('Frecuencia de muestreo %.2f[Hz]\naudio %s' % (fs1,nombre_out))
```

Figura 1. Cambio de Formato .mp3 a .wav

```
xpro1 = x1.copy()
Xw1 = np.fft.rfft(xpro1,axis=0)
vf1 = np.fft.rfftfreq(np.size(xpro1,0),1/fs1)
xwm1 = (abs(Xw1[:,[0]]))
xwmm1 = (abs(Xw1[:,[1]]))
xwmt1 = np.transpose(xwm1)
xwmy1 = np.transpose(xwmm1)
```

Figura 2. Respectivos vectores de la transformada rápida deFourier

```
xpro1 = x1.copy()
Xw1 = np.fft.rfft(xpro1,axis=0)
vf1 = np.fft.rfftfreq(np.size(xpro1,0),1/fs1)
xwm1 = (abs(Xw1[:,[0]]))
xwmm1 = (abs(Xw1[:,[1]]))
xwmt1 = np.transpose(xwm1)
xwmy1 = np.transpose(xwmm1)
plt.plot(vf1,abs(Xw1))
plt.legend(('canal 1','canal 2'))
plt.title(r'Espectro audio me voy de la casa')
plt.xlabel(r'f[Hz]',fontsize = 14)
plt.ylabel(r'$|X[n]|$',fontsize = 14)

ext(0, 0.5, '$|X[n]|$')
```

Figura 3. Gráfica del espectro

ON A ANALIZAR

```
nombre_in = "1.mp3"
nombre_out = "1.wav"
command_consola = "ffmpeg -i " + nombre_in +" "+ nombre_out
os.system(command_consola)

import soundfile as sf
xnuevo, fsnuevo = sf.read('1.wav')
xnuevo=xnuevo[0:mmax,:]

print('Frecuencia de muestreo %.2f[Hz]\naudio %s' % (fsnuevo,nombre_out))

ecuencia de muestreo 44100.00[Hz]
dio 1.wav
```

```
xpronuevo = xnuevo.copy()
Xwnuevo = np.fft.rfft(xpronuevo,axis=0)
vfnuevo = np.fft.rfftfreq(np.size(xpronuevo,0),1/fsnuevo)
xwmnuevo = (abs(Xwnuevo[:,[0]]))
xwmtnuevo = np.transpose(xwmnuevo)
xwmnuevo1 = (abs(Xwnuevo[:,[1]]))
xwmynuevo = np.transpose(xwmnuevo1)
```

Figura 4. Análisis del nuevo segmento

```
if (pos1 <= 31 and pos2 <= 31) :
     print ("la canción NO es de DANSU")
elif (pos1 <= 50 and pos2 <= 50) :
    print ("la canción es de DANSU y la detectamos como All you got")
elif (pos1 <= 68 and pos2 <= 68) :
    print ("la canción es de DANSU y la detectamos como Do do do")
elif (pos1 <= 85 and pos2 <= 85) :
    print ("la canción es de DANSU y la detectamos como Don't you give up")
elif (pos1 <= 101 and pos2 <= 101) :
    print ("la canción es de DANSU y la detectamos como Lost in the city")
elif (pos1 <= 119 and pos2 <= 119) :
    print ("la canción es de DANSU y la detectamos como Love is ours")
elif (pos1 <= 136 and pos2 <= 136):
    print ("la canción es de DANSU y la detectamos como Run")
elif (pos1 <= 157 and pos2 <= 157) :
    print ("la canción es de DANSU y la detectamos como Realize")

 canción es de DANSU y la detectamos como Run
```

Figura 5. Condicionales que detectan la respectiva canción

V. Resultados

El programa detecta con eficiencia las canciones de la banda seleccionada gracias a la cantidad de elementos compatibles en la base de datos, estos permiten que una adecuada comprensión de la canción se lleve a cabo por el programa y así se pueda determinar con éxito cualquier tipo de entrada incluso cuando el elemento que incide no coincide con los elementos guardados dentro de la memoria del programa, ya que podemos detectar que las canciones en general tiene un espectro continuo que solo puede generar cambios en ciertos picos a los cuales podemos denominar centrales de la canción y estos caracterizan a las canciones que se pretenden encontrar .

VI. Conclusiones

1. Se detectó que hay ciertas discordancias en algunos tipos de género debido a la cantidad de espectros recolectados en la base de datos, los que más se confunden son algunos tipos de pop que se conectan con el género correspondiente de las canciones a detectar.

2. La distancia euclídea es una distancia sensible a las unidades de medida de las variables: las diferencias entre los valores de variables medidas con valores altos contribuirán en mucha mayor medida que las diferencias entre los valores de las variables con valores bajos. Como consecuencia de ello, los cambios de escala determinarán, también, cambios en la distancia entre los individuos, así se pueden generar problema en las canciones que inciden de diferentes géneros.

3. Las coincidencias en algunas canciones se generan por los ritmos agudos que a medida se van generando en las canciones pop o en algunas melodias romanticas.

4. Al momento de analizar la canción fuera de la base de datos se notó que para el código propuesto no se podía incluir toda la canción y que máximo, para que no hubiera alteraciones en los resultados, esta canción o más bien el segmento a utilizar no debía sobrepasar los 40 segundos, esto porque se le aplicaba el mismo proceso que a los demás segmentos que solo duraban 10 segundos o menos.

5. Se debe tener en cuenta que todos los segmentos incluidos en el código tengan la misma frecuencia de muestreo ya que esto afecta al tamaño de cada vector, los cuales deben tener las mismas dimensiones para los análisis propuestos en el programa.

6. Se utilizaron ambos canales arrojados por el espectro ya que ofrecía una mejor exactitud de análisis, en especial para detectar qué canción del artista se estaba analizando.

VII. Alternativas del estado del arte

De los nuevos avances en el tema de detección de contenidos se pueden tener en cuenta aplicaciones como Shazam, Youtube y Facebook, que usan un programa basado en el análisis del espectrograma de las canciones, que es la representación gráfica del espectro de frecuencias [5]. También de aplicaciones como Spotify que usan esta detección de contenidos para encontrar similitudes entre las

canciones que reproducen los usuarios y así poder generar recomendaciones.

1. Una forma de mejorar el algoritmo de detección de canciones es usando la metodología empleada por el algoritmo de Shazam que se basa en comparaciones entre el espectrograma de las canciones, con esta representación gráfica se analizan distancias entre las frecuencias obtenidas del espectro (con puntos claves seleccionados del espectrograma). La base de datos se conforma de relaciones temporales entre las frecuencias (que busca obtener las frecuencias y las distancia temporal entre cada una), esto le permite a Shazam acelerar el proceso de búsqueda, aunque requiera un aumento mucho mayor en los datos guardados en la base (uso de más memoria). Entonces al momento de analizar un segmento de la canción que se quiere identificar el programa obtiene los puntos claves con el espectrograma, de este las relaciones temporales de frecuencia y busca similitudes con las canciones analizadas de la base de datos, así la canción con la mayor cantidad de relaciones iguales será la canción que se está buscando [6].

2. Otra de las opciones que se usan hoy día es en aplicaciones como Spotify, que si bien no identifica canciones como tal, si usa programas para recomendación de contenido que se basa en varias cosas, entre ellas la comparación de la música que el usuario escucha, estas comparaciones necesitan del análisis de audio y búsqueda de similitudes con el contenido a recomendar. Este modo de análisis de audio de Spotify se basa en las redes neuronales convolucionales, las cuales se usan en el reconocimiento facial, el cambio es que la red es entrenada con representaciones tiempo-frecuencia que se unen para formar el espectrograma de la canción y así permite que la red tenga un entendimiento de la canción para sacar características claves de la misma y darle a la aplicación la opción de comprender las similitudes fundamentales con otras canciones y así ofrecer buenas recomendaciones [7].

VIII. Referencias

[1]http://hyperphysics.phy-astr.gsu.edu/hbasees/Math/fft.html

[2]https://www.ecured.cu/Distancia-eucl

[3]https://uniwebsidad.com/libros/python/capitulo-10/modulos-de-sistema

[4]https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.norm.html

[5]http://musiki.org.ar/Espectrograma-(sonograma)

[6]https://www.ee.columbia.edu/-dpwe/papers/Wang03-shazam.pdf

[7]https://promocionmusical.es/como-funcionan-algoritmos-recomendacion-spotifyModelo-de-Recomendacion-3-Modelos-de-Audio-sin-Procesar