Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Selección de características para la clasificación de géneros musicales

Autora: Judith Gálvez Capitán
Tutora: María Auxiliadora Sarmiento Vega
Co-tutora: Irene Fondón García

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Selección de características para la clasificación de géneros musicales

Autora:
Judith Gálvez Capitán

Tutora:
María Auxiliadora Sarmiento Vega
Profesora Contratada Doctora

Co-tutora:
Irene Fondón García
Profesora Titular

Dpto. de Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2020

Trabajo Fin de Grado: Selección de características para la clasificación de géneros musicales

Autora: Judith Gálvez Capitán

Tutora: María Auxiliadora Sarmiento Vega

Co-tutora: Irene Fondón García

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A mi familia

A mis maestros

Resumen

En el presente trabajo se realiza un estudio sobre diferentes características para la clasificación de géneros musicales. Este trabajo se valida en una base de datos pública que permite la clasificación en 10 géneros musicales.

En primer lugar, se hace una breve introducción sobre el trabajo. En esta parte también se realiza un estado del arte recogiendo los distintos tipos de clasificación de géneros musicales usados con anterioridad para establecer los antecedentes del tema que versa este trabajo.

A continuación, se exponen los tipos de características escogidos para dicha clasificación y una breve explicación de los motivos, relacionados con la información recogida en el estado del arte del apartado anterior. Así como una descripción de cada característica y su forma de calcularlo.

En tercer lugar, se explica el código usado para obtener cada tipo de característica escogida y su uso para la posterior clasificación. Se expone el código necesario para el cálculo de las diferentes características y el tipo de clasificador usado y sus características.

Posteriormente, se comparan los resultados obtenidos para cada característica y se recogen dichos resultados en tablas para una mejor visualización. También se muestran gráficos con los resultados obtenidos mediante el clasificador seleccionado.

Por último, se realiza una conclusión finalacerca de dichos resultados donde se expone el tipo de característica que proporciona mejores resultados.

El objetivo del trabajo es obtener las características que presentan mayor precisión para la clasificación de géneros musicales.

Abstract

In the present work a study is carry out on different characteristics for the classification of musical genres. This work is validated in a public database that allows classification into 10 musical genres.

First, a brief introduction about this work is made. In this part, a state of the art is also carry out, which recognizes the different types of classification of musical genres used previously to establish the background of the subject matter of this work.

Then, the types of characteristics chosen for the classification are exposed and a brief explanation of the reasons, related to the information collected in the state of the art in the previous section. As well as a description of each characteristic and its method of calculation.

Third, the code used to obtain each type of characteristics and its use for classification is explained. The code necessary for the calculation of the different characteristic and the type of classifier used and its characteristics are exposed.

Subsequently, the specific results for each characteristic are compared and these results are collected in tables for a better visualization. Charts with the results obtained by the selected classifier are also shown.

Finally, a conclusion is made about these results where the type of characteristic that provides better results are exposed.

The objective of the work is to obtain the characteristics that present greater accuracy for the classification of musical genres.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Notación

e	Número e
π	Número pi
sin	Función seno
cos	Función coseno
$\sum$	Sumatorio
:	Tal que
$<$	Menor que
$\geq$	Mayor o igual que

1 INTRODUCCIÓN

El aumento del conocimiento depende por completo de la existencia del desacuerdo.

- Karl Popper -

Esta memoria se dividirá en 5 capítulos. En la primera parte, se recoge una breve introducción del trabajo. En la segunda parte, se realiza una breve explicación de los diferentes tipos de características a estudiar. A continuación, se detalla el código utilizado para la obtención de las características y la clasificación mediante éstas. Posteriormente, se exponen los resultados obtenidos. Por último, se realiza una conclusión y se enuncian líneas futuras de investigación tras la realización del trabajo.

El objetivo de este trabajo es seleccionar las mejores características para la clasificación de géneros musicales. Para ello, se estudian diferentes tipos de características para hallar la que permite mayor precisión a la hora de clasificar una canción dentro de 10 géneros musicales distintos.

Para la validación de la clasificación se ha usado una base de datos pública de 10 géneros musicales, con 100 canciones de cada género. Cada canción tiene una duración de 30 segundos.

A continuación, se va a realizar un estudio del estado del arte en clasificación de géneros musicales.

1.1 Estado del arte

El ISMIR (International SocietyforMusicInformationRetrieval) es un foro internacional para la investigaciónsobre la organización de datos relacionados con la música. Investigadores de todo el mundo se reúnen en la conferencia anual realizada por esta sociedad. En noviembre de 2019 tuvo lugar en Delft (Países Bajos) la vigésima conferencia de la sociedad internacional de recuperación de información musical (ISMIR).

Desde su creación, ISMIR ha sido el foro líder mundial para la investigación sobre el modelado, creación, búsqueda, procesamiento y uso de datos musicales.

MIREX (MusicInformationRetrievalEvaluationeXchange) es un marco de trabajo comunitario para la evaluación formal de los sistemas y algoritmos de recuperación de información musical (MIR). Esta comunidad define anualmente tareas y desafíos para promover avances en este campo.

Algunos ejemplos de tareas son las siguientes:

- Tareas de clasificación de audio (prueba/entrenamiento). Muchas tareas en la clasificación musical pueden caracterizarse en un proceso de dos etapas: entrenar modelos de clasificación utilizando datos

etiquetados y probar modelos utilizando datos nuevos no vistos [1].

- Clasificación de etiquetas de audio. Esta tarea compara las capacidades de varios algoritmos para asociar etiquetas descriptivas con clips de audio de canciones de 10 segundos. Se usan dos conjuntos de datos para implementar un par de subtareas, basadas en los conjuntos de datos de etiquetas MajorMiner y Mood. Las etiquetasMajorMinerse obtienen del juego MajorMiner [2]. El objetivo del juego es etiquetar canciones con palabras relevantes con las que otros jugadores están de acuerdo. Estas descripciones se usan posteriormente para enseñar a ordenadores a recomendar música que suene como la música que ya le gusta. Las canciones usadas son clips de 10 segundos que representan 1400 pistas diferentesde 800 álbumes distintos de 500 artistas diferentes. La etiqueta Moodhace referencia a etiquetas relacionadas con el estado de ánimo. Todas las etiquetas de este conjunto están identificadas por expertos humanos. Las etiquetas similares se agrupan para definir un grupo de etiquetas Moody cada canción puede pertenecera varios grupos. Hay 18 grupos de etiquetas Mood que contienen 135 etiquetas únicas. El conjunto de datos contiene 3469 canciones. Esta tarea está muy relacionada con otras tareas de clasificación de audio, sin embargo, se pueden aplicar múltiples etiquetas a cada ejemplo en lugar de la clasificación de una sola etiqueta [3].
- Detección de música. Esta tarea hace referencia a encontrar segmentos de música en un archivo de audio. Las dos aplicaciones principales de los algoritmos de detección de música son: la indexación automática y la recuperación de información auditiva basada en su contenido de audio, y el monitoreo de música para la gestión de derechos de autor. Además, la detección de música se puede aplicar como un paso intermedio para mejorar el rendimiento de algoritmos diseñados para otros fines [4].
- Detección de música y/o voz. En muchas tareas de procesamiento de audio es necesaria la detección de música y/o voz. La segregación de la señal en segmentos de voz y música es un primer paso antes de aplicar algoritmos específicos de voz o música [5].

Se ha observado que las señales de audio de música que pertenecen al mismo género comparten ciertas características porque están compuestas por tipos de instrumentos semejantes, tienen patrones rítmicos parecidos y distribuciones de tono similares.

Para la clasificación de géneros musicales se pueden utilizar muchas características diferentes. Por ejemplo, se pueden usar características de referencia como título y compositor;características acústicas basadas en contenido que incluyen tonalidad, tono y ritmo; carácterísticas simbólicas extraídas de las partituras, características basadas en texto extraídas de las letras de las canciones.

Las características acústicas basadas en contenido se clasifican en características de textura tímbrica, características de contenido rítmico y características de contenido tonal [6]. En este trabajo se han usado características de textura tímbrica como energía, centroide espectral, flujo espectral, cruces por cero, coeficientes cepstrales, roll-off espectral.

Logan [7] examinó los coeficientes cepstrales (MFCC) para el modelado de música y la discrimación entre música y voz. Las características de contenido rítmico contienen información sobre el ritmo, la regularidad de éste y la información del tempo. El seguimiento de tempo y ritmo de señales músicales acústicas se ha explorado en [8, 9, 10].

Foote [11] propuso el uso de coeficientes cepstrales (MFCC) para constuir un cuantificador de vector de árbol de aprendizaje. Los histogramas de las frecuencias relativas de los vectores de características en cada bin de cuantificación se utilizan posteriormente para la recuperación.

Ezzaidi y Rouat [12] propusieron dos métodos. Dividieron las piezas musicales en frames y posteriormente obtuvieron los coeficientes cepstrales (MFCC) de energías espectrales promediadas. Finalmente, para fines de comparación, utilizaron modelos de mezclas gaussianas (GMM) [13], obteniendo un máximo del 99% de reconocimiento. Se ha usado la base de datos RWC (Real World Computing), una base de datos disponible para investigadores. Está compuesta por 100 piezas musicales, 73 de ellas originales y 27 de dominio público.

Lambrou et al. [14] utilizaron características estadísticas en el dominio temporal, así como tres dominios de transformación diferentes para la clasificación musical en rock, piano y jazz. Estudiaron cuatro clasificadores

estadísticos: clasificador de mínima distancia (MDC) usando la mínima distancia euclídea, clasificador k-nearestneighbor (KNN), clasificador leastsquareminimumdistance (LSMDC) y clasificador de cuadratura (QC). Se usaron 12 muestras, 4 de cada género musical. Se alcanzó una precisión del 91.67% mediante el clasificador LSMDC, consiguiendo clasificar 11 de las 12 señales.

Deshpande et al. [15] clasificó en rock, clásico y jazz usando características de textura tímbrica mediante modelos de mezclas gaussianas, SVM (support vector machines) y algoritmos KNN. Se usó como base de datos 157 muestras de canciones, cada una de 20 segundos. De las muestras, 52 eran de rock, 53 de música clásica y 52 de jazz. Se usaron como entrenamiento 17 canciones de cada género seleccionadas de manera aleatoria. El mejor resultado obtenido fue del 75% usando KNN.

Tsanetakis y Cook [16] usaron características de textura tímbrica, ritmo y contenido de tono al proponer tres conjuntos de características para entrenar clasificadores de reconocimiento de patrones estadísticos como simple Gaussian (SG), Gaussian mixture model (GMM) y k-nearestneighbors (KNN) para la clasificación automática de géneros musicales. La precisión de la clasificación mediante el uso de estas características fue del 61% para un conjunto de datos de 10 géneros musicales diferentes. La base de datos usada constaba de 100 extractos de 30 segundos cada uno.

Silla et al. [17] adoptaron múltiples vectores de características, los mismos propuestos por Tsanetakis y Cook [16], seleccionados de diferentes segmentos de tiempo de partes del principio, medio y final de la música, y el enfoque de conjunto de reconomiento de patrones de acuerdo con una dimensión de descomposición espacio-tiempo. La mejor precisión obtenida fue del 65,06%. Se usa una base de datos conocida como LatinMusicDatabase, compuesta por 3160 piezas musicales pertenecientes a 10 géneros musicales distintos.

Li [18] usó el mismo conjunto de datos para comparar varios métodos de clasificación y conjuntos de características, y propuso el uso del método de clasificación de patrón de línea característica más cercano (NFL, siendo las siglas de NearestFeature Line). El método NFL supone que hay disponible un conjunto de sonidos prototipo (entrenamiento) y que existe más de un prototipo (punto de característica) para cada clase de sonido. NFL hace uso de la información proporcionada por múltiples prototipos por clase, en contraste con el método NN (nearestneighbor) en el cual la clasificación se realiza comparando la consulta con cada prototipo individualmente.

C. Xu et al. [19] utilizaron múltiples capas de SVM para lograr una precisión superior al 90% en un conjunto de datos que solo contiene cuatro géneros musicales.

Tao Li [6] propuso un nuevo método de extracción de características basado en el histograma de coeficientes wavelet, llamado DWCH (Daubechies Wavelet Coefficient Histogram). Usando DWCH con técnicas avanzadas de machine learning, se alcanza una precisión de casi el 80% en la clasificación de géneros musicales con el mismo conjunto de datos usado en [16].

Guohui Li et al. [20] propone una técnica de indexación y recuperación de audio basada en la transformada wavelet discreta (DWT). Usa una base de datos formada por 418 pistas de audio, que incluyen instrumentos musicales, máquinas, animales, habla. Este sistema alcanza el 70%.

Como referencia, la precisión promedia humana es alrededor del 70% [21] para una clasificación entre 10 géneros musicales distintos.Mingwen Dong [21] propuso un nuevo método que combina el conocimientodel estudio de la percepción humana en la clasificación del género musical y la neurofisiología del sistema auditivo. El método funciona entrenando una red neuronal convolucional simple (CNN) para clasificar un segmento corto de la señal de música. Luego, el género musical se determina dividiéndolo en segmentos cortos y posteriormente combinando las predicciones de CNN de todos los segmentos cortos. Después del entrenamiento, este método logra una precisión a nivel humano (70%) y los filtros aprendidos en la CNN se parecen al campo receptivo espectrotemporal (STRF) en el sistema auditivo. El campo receptivo de una neurona representa qué tipo de estímulos excitan o inhiben esa neurona. Espectrotemporal hace referencia a la audición, donde la respuesta de la neurona depende de la frecuencia frente al tiempo.

Panagakis y Kotropoulos [22] propusieron un marco de clasificación de géneros musicales que considera las propiedades del sistema auditivo de percepción humana, es decir, modulaciones temporales auditivas 2D que son usadas para la representación musical. Las características son extraídas mediante técnicas de reducción de dimensionalidad, se ha usado NMF (Non-negativeMatrixFactorization) para la base de datos GTZAN y PCA (Principal ComponentAnalysis) para la base de datos ISMIR2004. Se utiliza la clasificación SRC (SparseRepresentation-basedclassification). Las precisiones que obtuvieron para los conjuntos de datos GTZAN

e ISMIR2004 fueron 91% y 93,56%, respectivamente.

Paradzinets et al. [23] exploraron información acústica, características relacionadas con ritmo y timbre. Para obtener la información acústica, usaron características de modelado gaussiano por partes (PGM) mejoradas por el modelado de filtro auditivo humano. Para hacerlo, obtuvieron las características PGM y posteriormente aplicaron filtro de bandas críticas, igual volumen y sensación de volumen específico. Para extraer las características relacionadas con el ritmo, usaron transformaciones wavelet, obteniendo los histogramas de ritmo 2D. Para las características de timbre, recolectaron todas las notas detectadas con una amplitud relativa de sus armónicos y después calcularon sus histogramas.La base de datos usada cuenta con 1873 canciones de 822 artistas diferentes clasificadas manualmente en 6 géneros musicales. La precisión obtenida usando la combinación de las tres tipos de características fue del 66.7%.

A lo largo de los años se han realizado estudios de múltiples vectores de características para la clasificación de géneros musicales. Dichas características propuestas abarcan desde características musicales como ritmo, tono,timbre,etc hasta basadas en el histograma de las señales musicales. Sin embargo, en este trabajo se va a estudiar también el uso de descriptores de texturas propios del procesamiento de imágenes.

La clasificación de textura tiene un papel importante en el procesamiento de imágenes y en aplicaciones de visión por computadora, como el reconocimiento de caracteres [24], la detección de rostros [25], la clasificación de telas [26], la segmentación geográfica del paisaje [27].

1.2 Géneros Musicales

Los géneros musicales en los que se va a clasificar son los siguientes:

- Blues. Este género está basado en la utilización de las notas de blues y de un patrón repetitivo, que suele seguir una estructura de doce compases. En este género es común utilizar técnicas vocales como el melisma (sostener una sola sílaba a través de varios lanzamientos), técnicas rítmicas como la sincopación y técnicas instrumentales como ahogar o doblar cuerdas de guitarra en el cuello o aplicar una diapositiva metálica o cuello de botella a las cuerdas de la guitarra para crear un sonido de gemido [28].

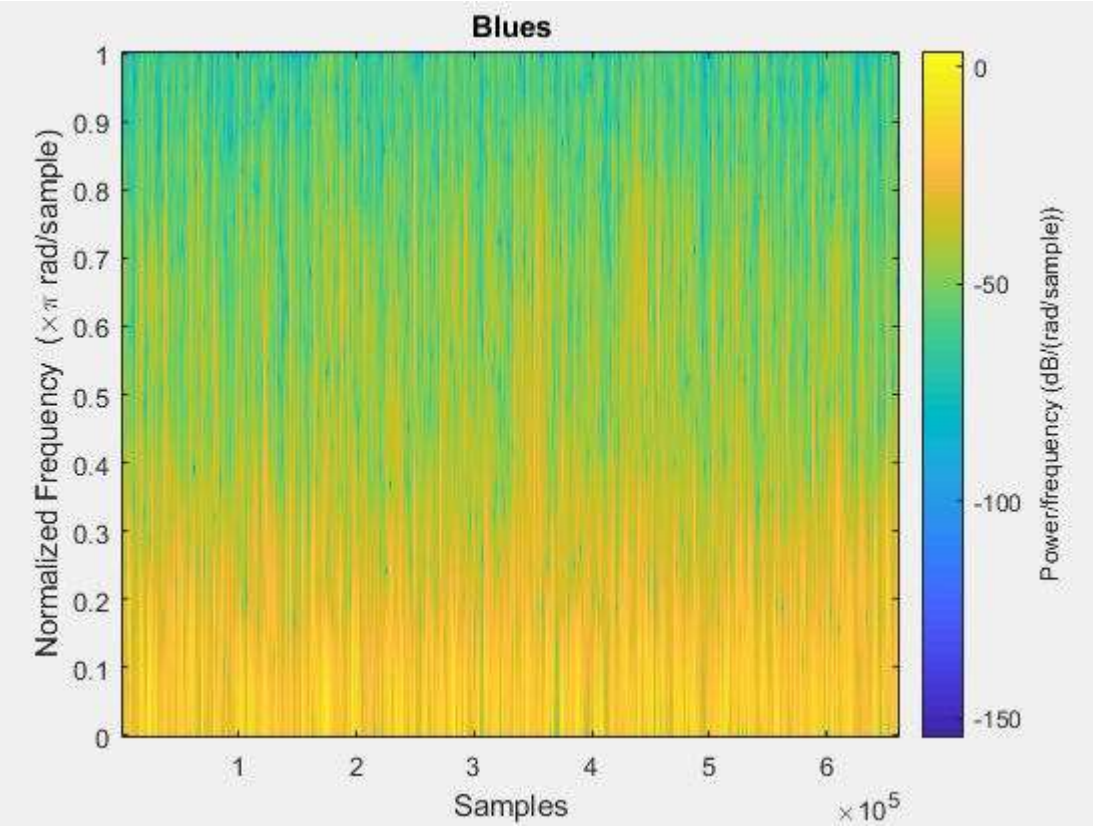

Figura 1-1. Espectrograma de Blues.

- Clásico. Los instrumentos usados en este género son los encontrados en la orquesta sinfónica. Los instrumentos electrónicos no juegan ningún papel en este género musical. Dada la amplia gama de estilos de la música clásica, desde la cantera medieval hasta las sinfonías clásicas y románticas, es difícil enumerar características que se puedan atribuira todas las obras de este género. Sin embargo, se podría decir que la música clásica se distingue por su característica notación musical simbólica. Dicha notación permite a los compositores prescribir de forma detallada el tempo, la métrica, el ritmo, la altura y la ejecución precisa de cada pieza musical. Esto limita el espacio para la improvisación, característica destacada de otros géneros musicales [29].

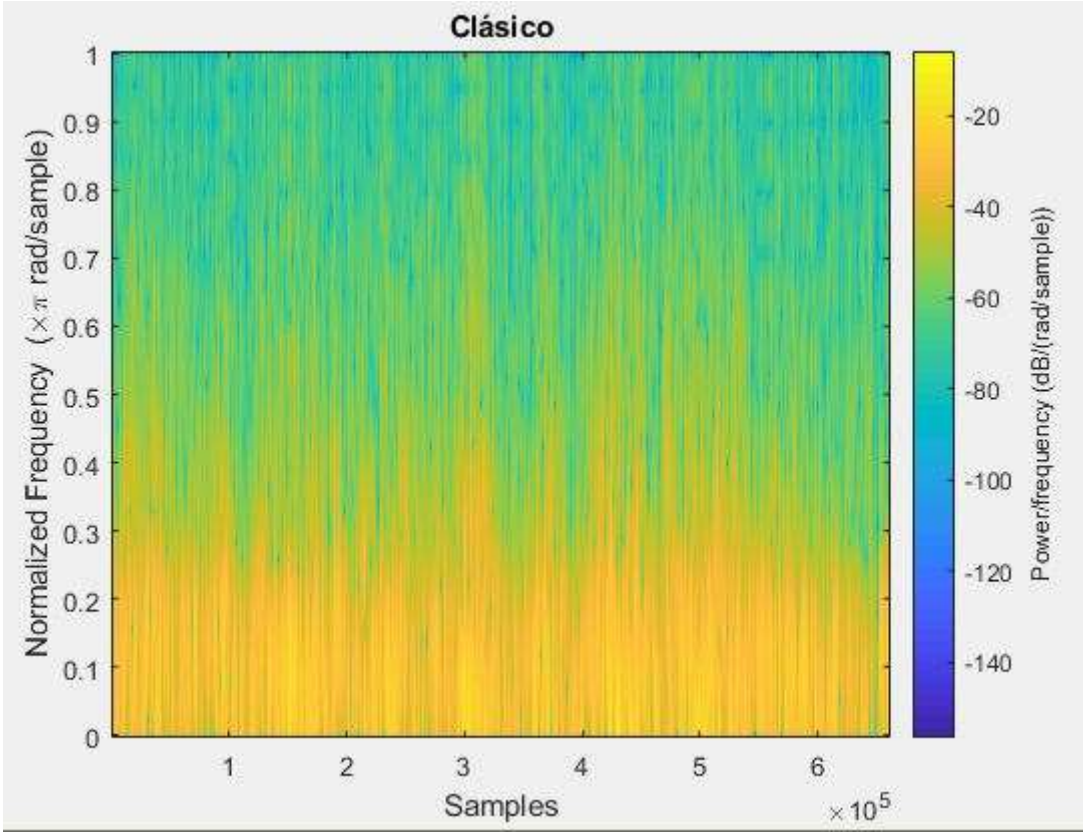

Figura 1-2. Espectrograma de Clásico.

- Country. Este género es principalmente acústico. En el country tradicional, los instrumentos usados esencialmente eran instrumentos de cuerda como la guitarra, el banjo, el violín y el contrabajo, aunque también se utilizaba frecuentemente la armónica y el acordeón. En el country moderno, se utilizan sobre todo los instrumentos electrónicos como la guitarra eléctrica, el bajo eléctrico, el teclado.Este género mantiene una pulsación muy constante en un ritmo binario y ligero o en un ritmo ternario más tranquilo.Se utilizan las escalas modales del folk y un fraseo clásico.Las características que definen a este género pueden varias notablemente en función de la localización geográfica, pero la fórmula básica de la música country consiste en una progresion de acordes sencillos yun coro resonante [30].

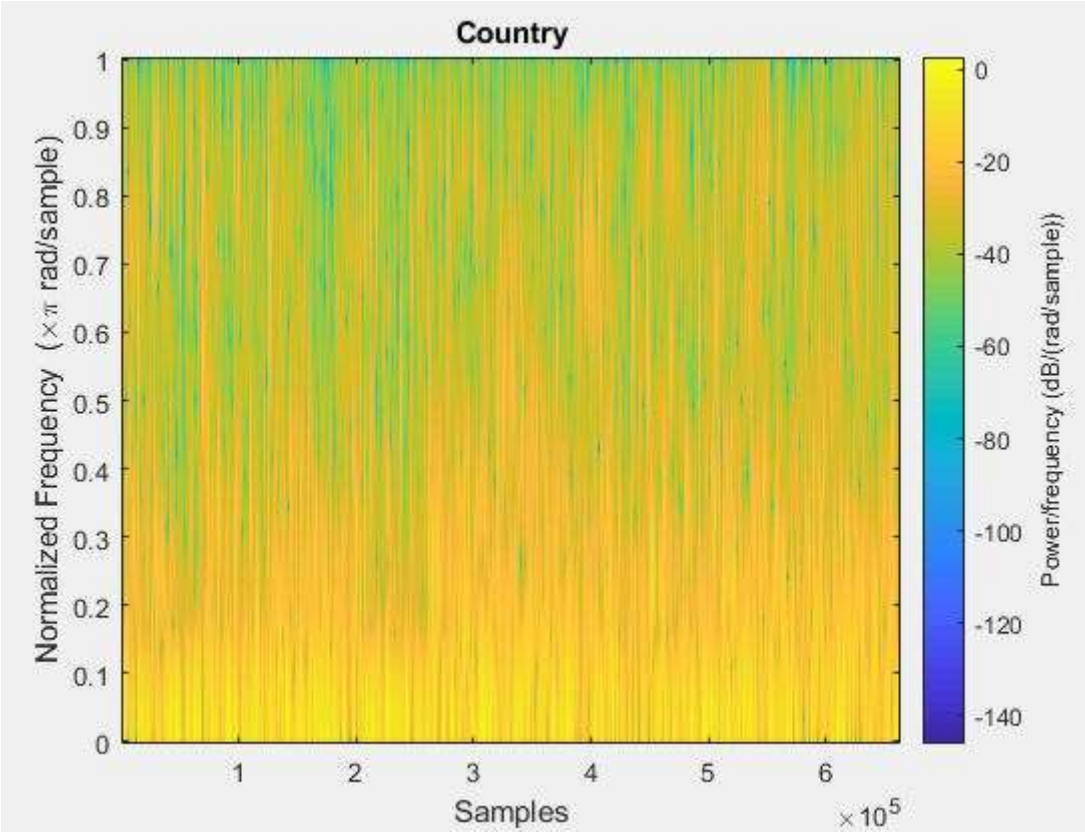

Figura 1-3. Espectrograma de Country.

- Disco. Las canciones de este género normalmente están estructuradas sobre un repetitivo compás de 4/4, marcado por una figura de hi hat, de ocho o dieciséis tiempos abierto en los tiempos libres, y una línea predominante de bajo sincopado, con voces fuertemente reverberadas. Son fácilmente reconocibles por sus ritmos repetitivos (generalmente entre 110 y 136 pulsaciones por minuto) y pegadizos. El sonido orquestal, usualmente conocido como sonido disco, se fundamenta en la presencia de secciones de cuerda y metales, que desarrollan frases lineales en unísono, tras la base instrumental formada por el piano y la guitarra eléctricos. Al contrario que en el rock, la guitarra solista es inusual [31].

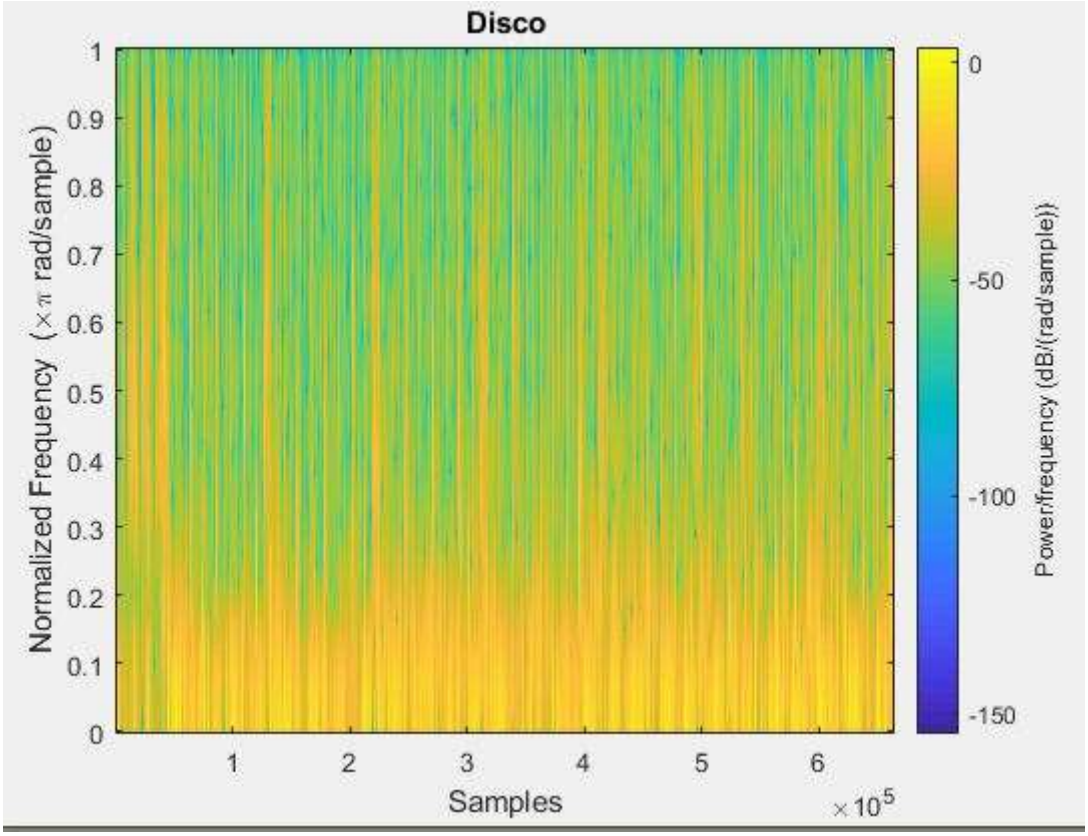

Figura 1-4. Espectrograma de Disco.

- Hip hop. Los instrumentos comunes de este género son: tocadiscos, sintetizador, DAW, caja de ritmos, sampler, beatboxing y teclado. Este género se caracteriza principalmente por el uso del rap y por los DJs.En el hip hop, el DJ selecciona los breaks, es decir, el ritmo base, y el MC o rapero inventa las rimas sobre el ritmo base del break [32].

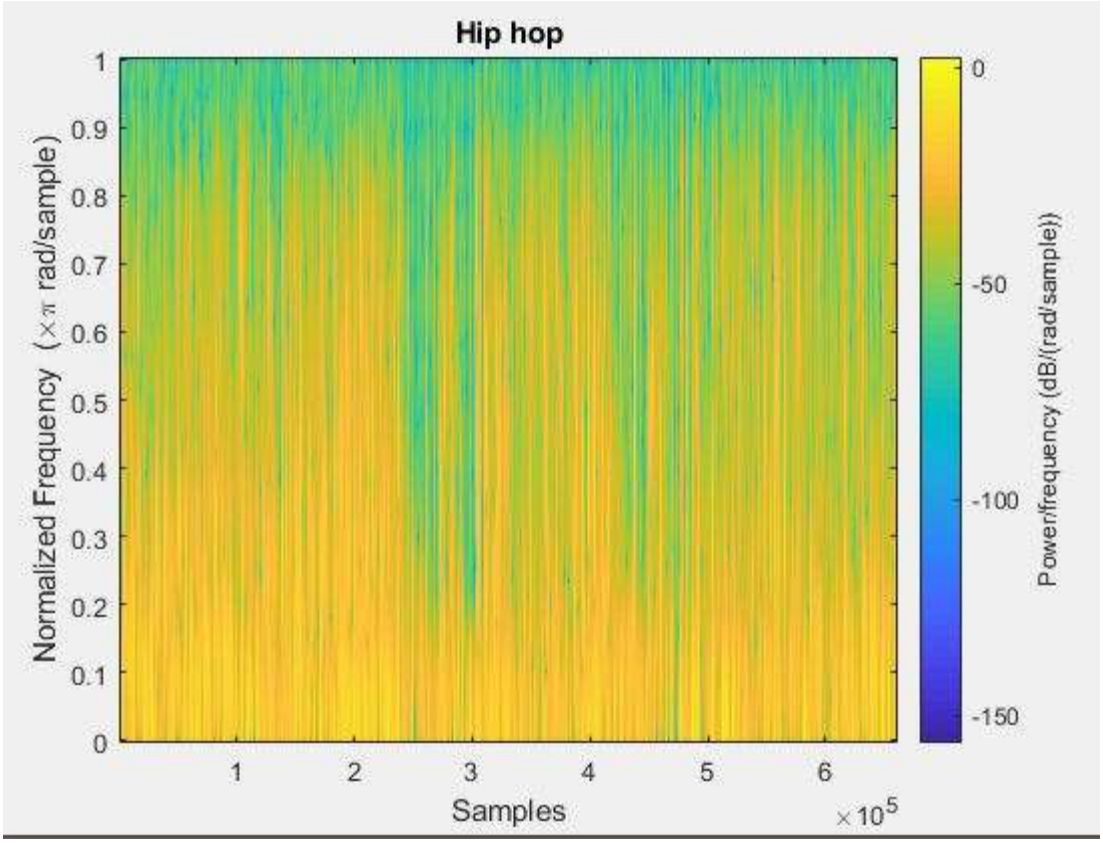

Figura 1-5. Espectrograma de Hip hop.

- Jazz. Hay tres elementos básicos que distinguen a este género: una cualidad rítmica especial conocida como swing, la improvisación y un sonido y un fraseo que reflejan la personalidad de los músicos ejecutantes. Una de las mayores diferencias entre el jazz y la música clásica tiene que ver con la formación del tono. Mientras que los integrantes de una sección de instrumentos en una orquesta clásica aspiran a obtener el mismo sonido de sus instrumentos, de forma que puedan ejecutar los pasajes del modo más homogéneo posible, los músicos de jazz aspiran a lograr un sonido propio que los distinga del resto. Toda agrupación de jazz dispone de una sección melódica y una sección rítmica. La primera está compuesta de instrumentos melódicos como el saxofón, la trompeta o el trombón, mientras que la segunda la componen instrumentos como la batería, la guitarra, el contrabajo y el piano. En general, la sección rítmica es la encargada de proveer el ritmo sobre el que tiene lugar la ejecución de melodías y solos [33].

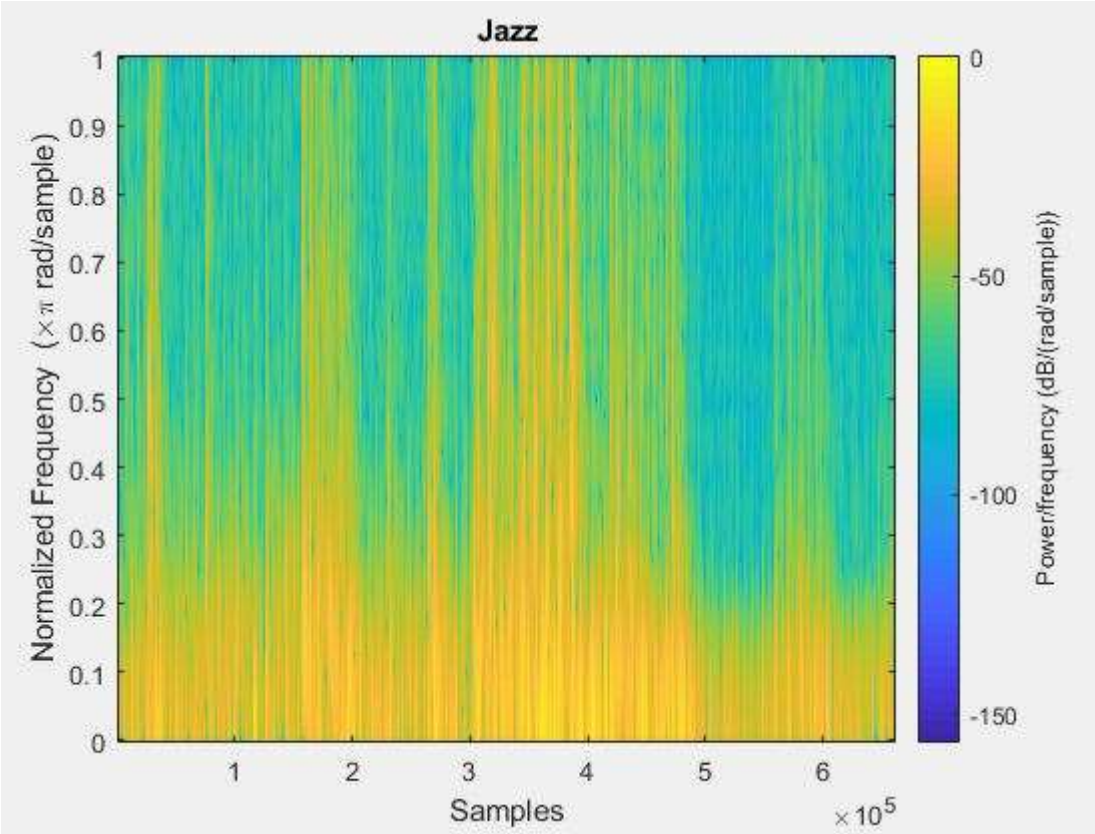

Figura 1-6. Espectrograma de Jazz.

- Metal. Este género se caracteriza principalmente por sus guitarras fuertes y distorsionadas, sus ritmos enfáticos, los sonidos del bajo y la batería más densos de lo habitual y por voces generalmente agudas. Los instrumentos comunes son la guitarra eléctrica, el bajo eléctrico y la batería. La guitarra eléctrica y la potencia que proyecta a través de la amplificación ha sido históricamente el elemento clave de este género, cuyo sonido proviene de un uso combinado de altos volúmenes y una gran distorsión. La voz se caracterizó en un principio por ser aguda con un gran uso del vibrato y una amplitud enorme de octavas. Otra de las técnicas que se utiliza es el falsete. Con el pasar de los años, algunos vocalistas emplearon un tono más rudo y más alejado de la agudeza propia de este género. La batería crea un ritmo fuerte y constante basándose en la velocidad, potencia y precisión. Una de las características propias del batería de este género es el uso del cymbalchoke, que es tocar los platillos y silenciarlos rápidamente con la mano [34].

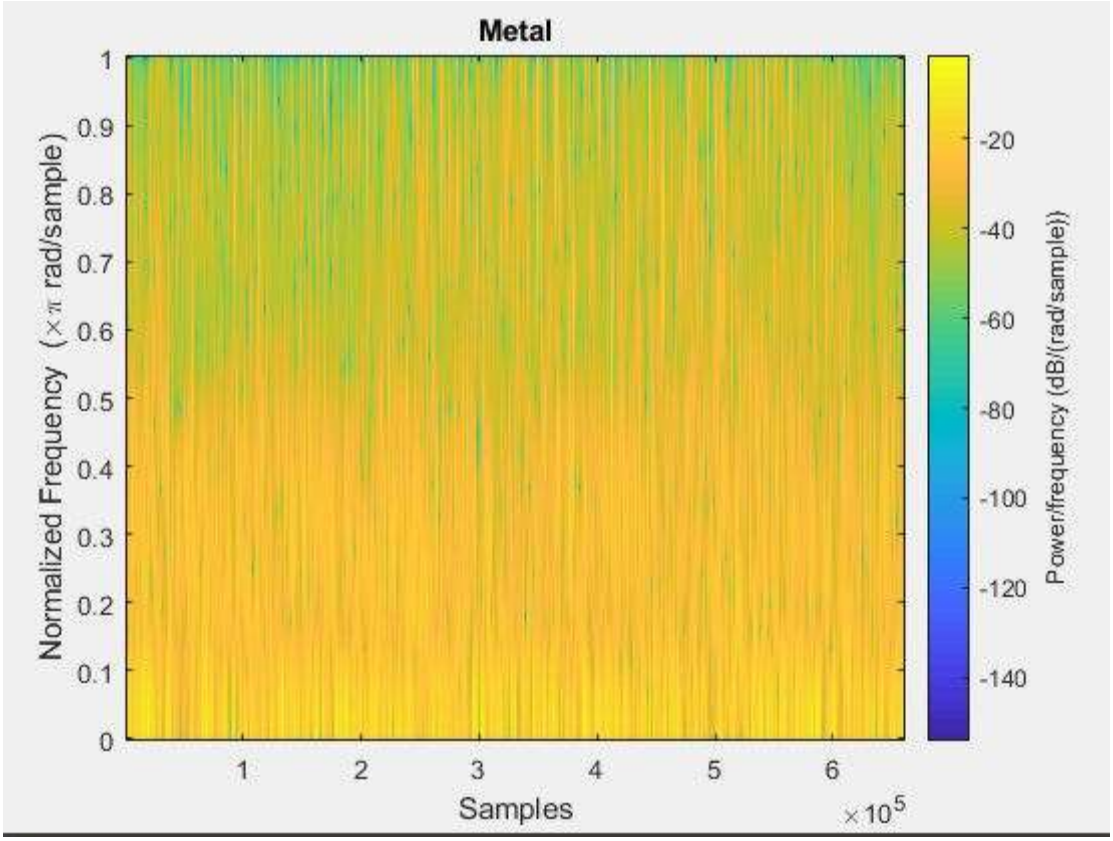

Figura 1-7. Espectrograma de Metal.

- Pop. Los elementos esenciales que definen este género son las canciones de corta a media duración (a menudo entre tres y cinco minutos) escritas en un formato básico, el uso habitual de estribillos repetidos y temas melódicos. El ritmo y las melodías tienden a ser sencillos, con un acompañamiento armónico limitado [35].

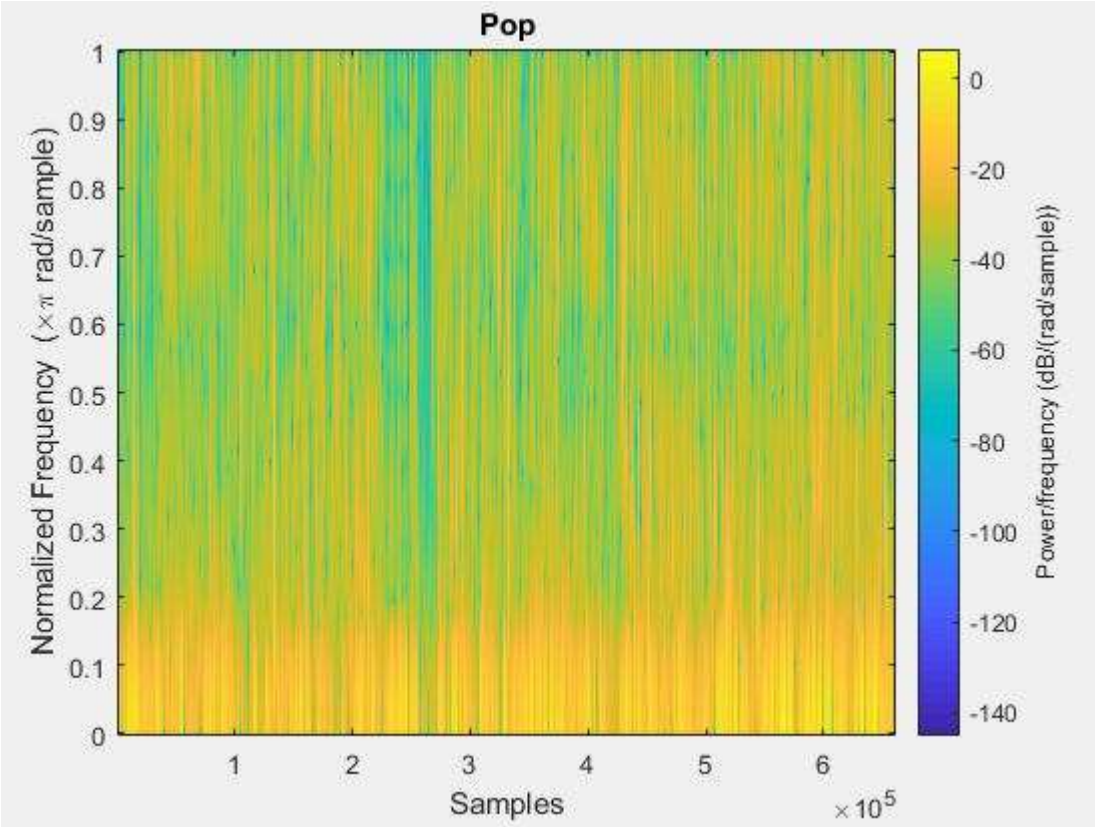

Figura 1-8. Espectrograma de Pop.

- Reggae. Este género se caracteriza rítmicamente por un tipo de acentuación del segundo y cuarto pulso de cada compás, y utiliza la guitarra para poner o bien énfasis en el tercer pulso o para mantener el acorde del segundo hasta el cuarto. El tempo de este género es normalmente lento. Está liderado por la batería y el bajo. El sonido del bajo es grueso y pesado, y está ecualizado, por lo que se eliminan las frecuentas altas y se enfatizan las frecuencias más bajas. El patrón rítmico simétrico de este género no se presta a otras formas de tiempo diferentes al 4/4. Uno de los elementos más fácilmente reconocibles son los acordes staccato tocados por una guitarra o piano (o ambos), así como el uso de líneas de bajo melódicas y sincopadas [36].

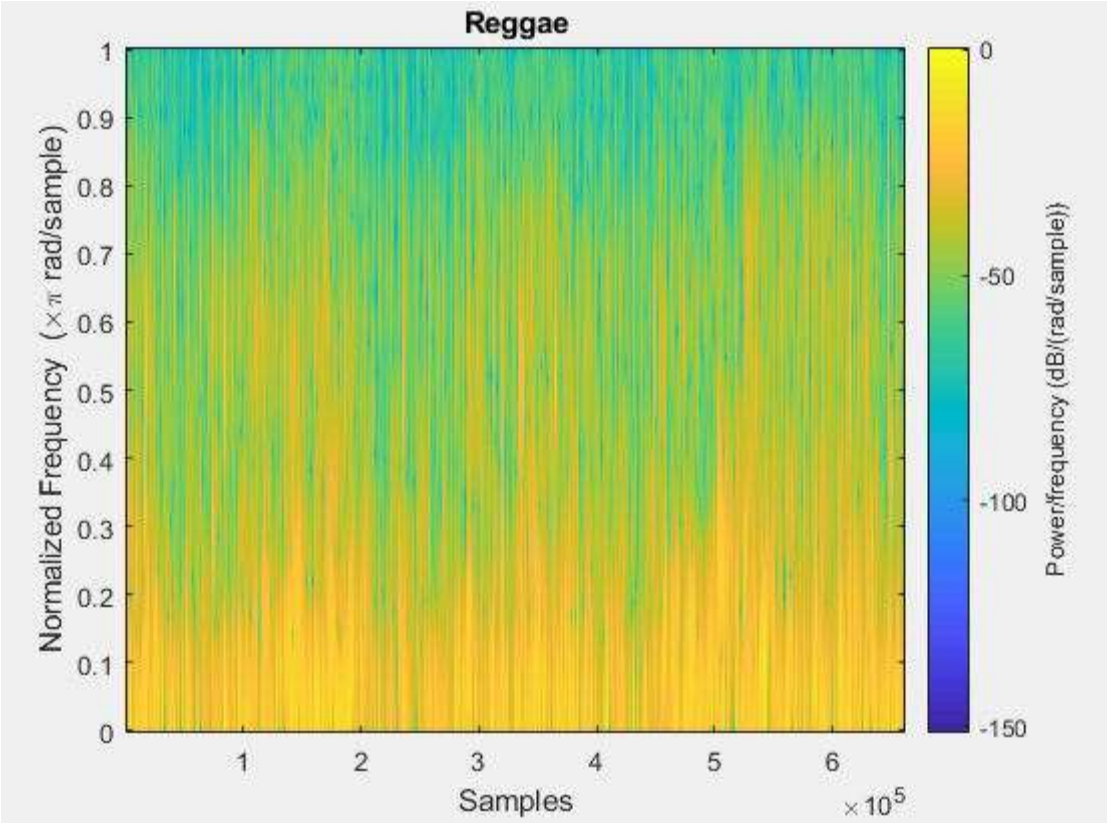

Figura 1-9. Espectrograma de Reggae.

- Rock. Los instrumentos típicos usados en este género son: guitarra eléctrica, bajo, batería y teclado. El rock se construye tradicionalmente sobre una base de ritmos simples no sincopados en compases de 4/4, con un ritmo repetitivo de tambor en los tiempos 2 y 4. Este género hace uso de los recursos de amplificación sonora a la máxima potencia, de una marcada acentuación rítmica y de efectos de distorsión [37].

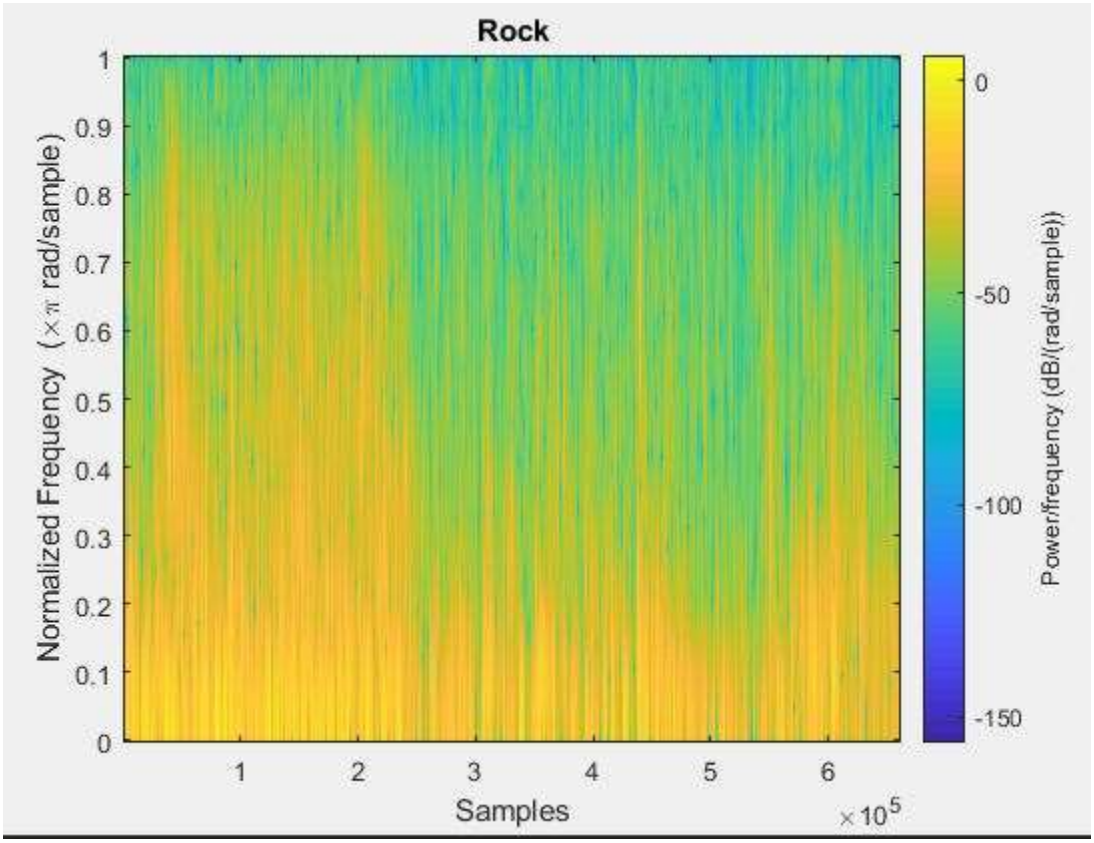

Figura 1-100. Espectrograma de Rock.

2 TIPOS DE CARACTERÍSTICAS

La educación no es aprender hechos, sino entrenar la mente a pensar.

- Albert Einstein -

En este capítulo se van a detallar los diferentes tipos de características usadas para la clasificación de géneros musicales.

En primer lugar, se va a usar un tipo de descriptor de textura para el que se realiza el espectrograma de cada canción y así trabajar con las muestras como si se trataran de imágenes, buscando diferentes patrones para cada género musical. Este tipo de característica no se ha usado antes para clasificación de géneros musicales.

En segundo lugar, se va a utilizar una variante del descriptor de textura anterior que presenta ventajas frente al ruido denominada AELBP.

En tercer lugar, se va a usar la Transforma Discreta de Fourier (DFT).

Por último, se van a usar un conjunto de características espectrales detalladas más adelante.

2.1 LBP

LBP (Local BinaryPattern) es un descriptor de textura para imágenes que limita los píxeles vecinos en función del valor del píxel actual.Los descriptores LBP capturan los patrones espaciales locales y el contraste de la escala de grises en una imagen [38].

El cálculo del descriptor LBP de una imagen es un proceso de cuatro pasos, explicados a continuación.

1. Por cada píxel (x,y) en una imagen, se elige P píxeles vecinos en un radio R.
2. Se calcula la diferencia de intensidad del píxel actual (x,y) con los P píxeles adyacentes.
3. Se umbraliza la diferencia de intensidad, de modo que a todas las diferencias negativas se asigna 0 y a todas las positivas, 1. Formando así un vector de bits.
4. Se convierte el vector P-bit a su valor decimal correspondiente y se reemplaza el valor de intensidad en (x,y) con ese valor decimal.

A continuación, se muestra un ejemplo de los pasos anteriores.

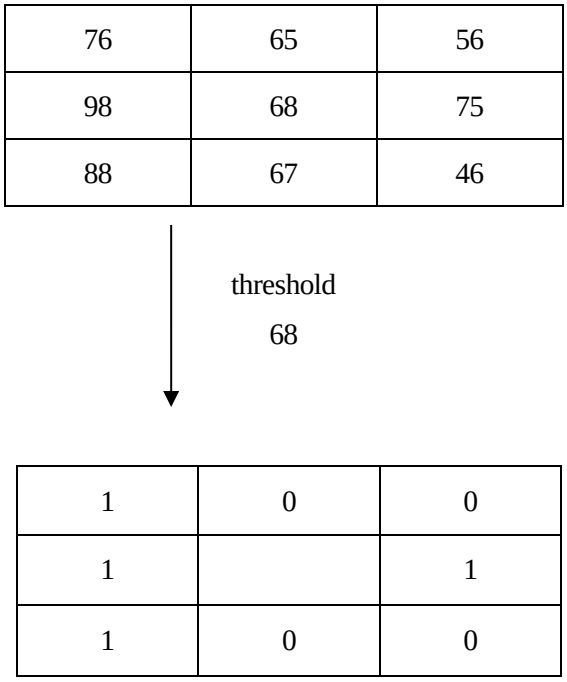

Figura 2-1.Operador LBP.

Por lo tanto, el descriptor LBP para cada píxel se da como

$$LBP(P,R) = \sum_{p=0}^{P-1} f(g_p - g_c)2^p \qquad (2-1)$$

donde g_c yg_p denota la intensidad de los píxeles actuales y vecinos, respectivamente; P es el número de píxeles vecinos elegidos a un radio R.

Para usar este tipo de descriptor como vector de característica, es necesario tratar las muestras de audio como imágenes. Para ello se debe realizar el espectrograma de cada muestra, que es una representación visual del espectro de frecuencias de una señal que varía en el tiempo.

El espectrograma se puede crear a partir de una señal en el dominio del tiempo de dos formas: aproximado como un banco de filtros que resulta de una serie de filtros de paso de banda o calculado a partir de la señal de tiempo utilizando la transformada de Fourier. En este trabajo se usa el segundo método, que corresponde esencialmente a calcular la magnitud al cuadrado de la transformada de Fourier de tiempo reducido (STFT) de la señal.

$$X(f,t) = |STFT(f,t)|^2 \qquad (2-2)$$

donde f es el bin de frecuencia y t el bin de tiempo o número de frame.

En este trabajo se va a usar para crear el vector de características el histograma de los códigos LBP calculados a partir del espectrograma de cada muestra de audio. En la sección 3 se detallará el código necesario para extraer dicho vector de características.

2.2 AELBP

AELBP (AdjacentEvaluation Local BinaryPattern) es un descriptor de textura,una variante de LBP [39]. El operador convencional LBP es sensible al ruido ya que los valores de los vecinos pueden ser modificados fácilmente por ruido aleatorio, lo que hace a este descriptor inestable. AELBP intenta reducir la interferencia del ruido mediante la construcción de una ventana de evaluación adyacente que rodea a los vecinos.

Para el cálculo del descriptor AELBP se construye una ventana de evaluación adyacente para modificar el esquema de umbral de LBP. Los vecinos del centro vecinal g_c se configuran como el centro de evaluación a_p. Rodeando el centro de evaluación, configuramos una ventana de evaluación y calculamos el valor de a_p. Luego, se extraen los códigos binarios locales comparando el valor de a_p con el valor del centro de vecindad g_c.

Se define de la siguiente manera

$$AELBP(P,R) = \sum_{p=0}^{P-1} x(a_p - g_c)2^p \qquad (2-3)$$

$$x(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

donde g_c denota la intensidad de los píxeles vecinos, P es el número de píxeles vecinos y a_p es el valor medio de la p^{th} ventana de evaluación excluyendo el valor del centro de evaluación, y R es el radio.

En la siguiente figura se muestra un ejemplo del cálculo de AELBP siendo los valores de P y R, 8 y 1, respectivamente.

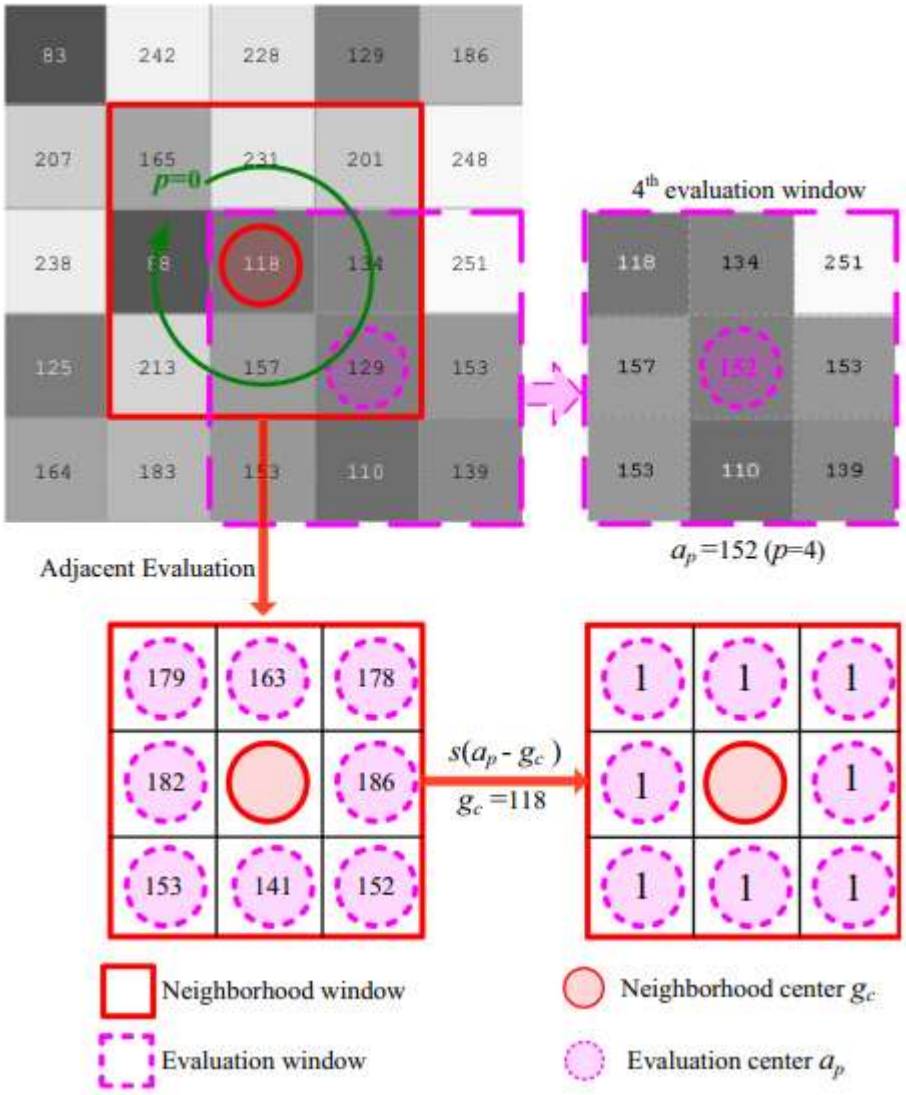

Figura 2-2. Operador AELBP [39].

En este trabajo se va a usar para crear el vector de características el histograma de los códigos AELBP calculados a partir del espectrograma de cada muestra de audio. Al igual que en el caso anterior del descriptor LBP. En ambos casos se usan los mismos valores de radio y número de píxeles vecinos. En la sección 3 se detallará el código necesario para extraer dicho vector de características.

2.3 DFT

DFT (Discrete Fourier Transform) es un tipo de transformada discreta usada en el análisis de Fourier. Transforma una función matemática en otra, obteniendo una representación en el dominio de la frecuencia, siendo la función original una función en el dominio del tiempo.

Esta transformada modifica una secuencia de N números complejos $\{x_n\} := x_0, x_1, \ldots, x_{N-1}$ en otra secuencia de números complejos $\{X_k\} := X_0, X_1, \ldots, X_{N-1}$, que es definida por

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N}kn\right) - i \cdot \sin\left(\frac{2\pi}{N}kn\right)\right] \qquad (2-4)$$

donde la última expresión se obtiene de la primera mediante la fórmula de Euler.

La DFT tiene diversas aplicaciones como son:

- Análisis espectral.
- Convolución rápida.
- Síntesis de sonido por modelado espectral.
- Compresión de audio.
- Filtrado.

2.4 Características espectrales

A continuación se presentan las características acústicas más importantes de las señales musicales para llevar a cabo la clasificación en géneros musicales.

Las características tímbricas se usan para diferenciar mezclas de sonidos que tienen posiblemente similares ritmos y tonos. El uso de estas características se origina en el reconocimiento de voz. Para extraer las características tímbricas, las señales de sonido se dividen primero en frames que son estadísticamente estacionarios, generalmente aplicando una función de ventana a intervalos fijos. La función de ventana, típicamente una ventana de Hamming, elimina los efectos de borde. Las características tímbricas se calculan para cada frame, y se calculan los valores estadísticos (como la media y la varianza) de estas características.

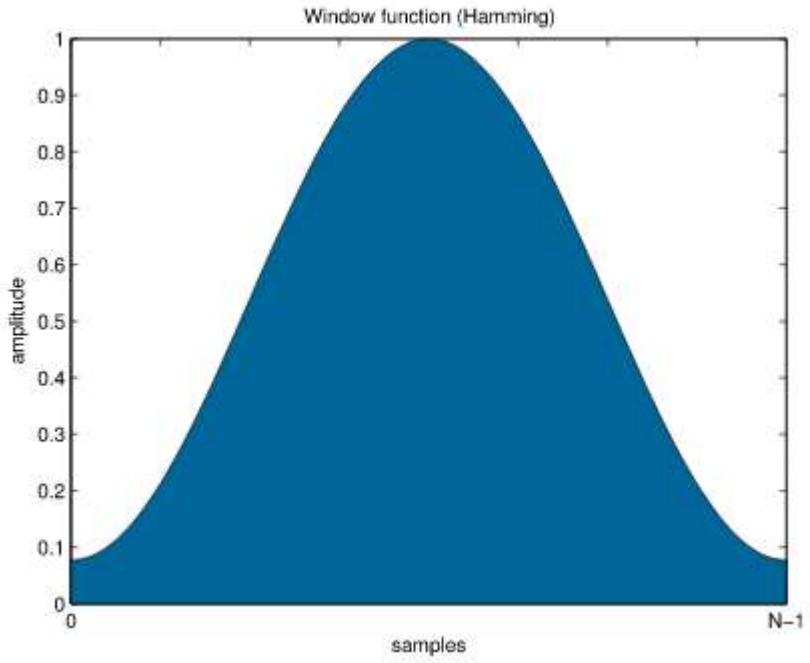

Figura 2-3. Ventana de Hamming.

A partir de la representación de una muestra de una canción en función del tiempo se puede obtener información sobre características importantes como la energía y los cruces por cero, que facilitan su estudio y análisis.

Por ello, se han recopilado una serie de características a estudiar para la clasificación de canciones. A continuación, se listan dichas características:

- cruces por cero: indican el número de veces que una señal continua toma el valor cero en el dominio del tiempo. Las señales de mayor frecuencia presentan un mayor valor de esta característica. Mide el ruido de la señal. La fórmula es la siguiente:

$$ZCR = \frac{1}{N}\sum_{n=1}^{N} |sgn[x(n)] - sgn[x(n-1)]| \qquad (2-5)$$

  donde $x(n)$ es una señal de audio finita y N es el número total de muestras. Se usan la media y la desviación típica como características.

- energía: permite distinguir entre segmentos sonoros y sordos en la señal de voz, debido a que los valores de esta característica aumentan en los sonidos sonoros respecto a los sordos. Para el cálculo de la energía de una señal discreta se usa la siguiente fórmula:

$$E = \sum_{n=0}^{N} |x(n)|^2 \qquad (2-6)$$

  donde $x(n)$ es una señal de audio finita y N es el número total de muestras. En el vector de características se usa la media y la desviación típica de la energía.

- entropía de la energía: aporta información sobre si la señal contiene o no picos predominantes. La fórmula usada para su cálculo es la siguiente:

$$H(X) = -\sum_{n=1}^{N} x(n) \cdot \log_2(x(n)) \qquad (2-7)$$

  En este trabajo se usa como características la media y la desviación típica de la entropía de la energía.

- centroide espectral: es una medida usada en el procesamiento de señales digitales para caracterizar un espectro. Indica dónde se encuentra el centro de masa del espectro. Debido a que es un buen predictor del "brillo" de un sonido, se usa como medida automática del timbre musical. La fórmula usada para su cálculo es la siguiente:

$$C_t = \frac{\sum_{n=1}^{N} M_t(n) * n}{\sum_{n=1}^{N} M_t(n)} \qquad (2-8)$$

  donde $M_t(n)$ es la magnitud de la transformada de Fourier en el frame t y el bin de frecuencia n. Se usa en el vector de características la media y la desviación típica del centroide.

- entropía espectral: es una medida de la distribución de potencia espectral de una señal. Se utiliza como característica en el reconocimiento de voz. Se ha usado la siguiente fórmula para su cálculo:

$$H(X) = -\sum_{n=1}^{N} p(x(n)) \cdot \log_2\big(p(x(n))\big) \qquad (2-9)$$

  donde $p(x(n))$ es la distribución de probabilidad de la señal. Como se puede observar, su cálculo es similar al de la entropía de la energía, pero en este caso en el dominio de la frecuencia. Al igual que en las características anteriores, se usa su media y su desviación típica.

- flujo espectral: es una medida de la rapidez con que cambia el espectro de potencia de una señal. Esta característica se puede usar para determinar el timbre de una señal de audio o en la detección de inicio,

entre otras cosas. Es la diferencia al cuadrado entre las magnitudes normalizadas de distribuciones espectrales sucesivas. Se calcula mediante la siguiente fórmula:

$$F_t = \sum_{n=1}^{N} (N_t(n) - N_{t-1}(n))^2 \qquad (2-10)$$

donde $N_t(n)$ y $N_{t-1}(n)$ son las magnitudes normalizadas de la transformada de Fourier en el frame actual t y el frame anterior $t-1$, respectivamente. Se usa la media y la desviación típica del flujo espectral como características.

- centroide del flujo espectral. La fórmula usada es la siguiente:

$$FC_t = \frac{\sum_{n=1}^{N} f(t)|P_t(n) - P_{t-1}(n)|}{\sum_{n=1}^{N}|P_t(n) - P_{t-1}(n)|} \qquad (2-11)$$

En este trabajo se utiliza la media y la desviación típica de este centroide para formar el vector de características.

- roll-off espectral: es una medida de la forma espectral. Se define como el intervalo de frecuencia M por debajo del cual se concentra el 85% de la distribución de magnitud. Se calcula mediante la siguiente ecuación, donde $x_n$es el valor espectral en el bin n, b_1 y b_2 son los bordes de banda, en bins, sobre los que se calcula la dispersión espectral, y k es el porcentaje de energía total contenido entre b_1y i:

$$\sum_{n=b_1}^{i} x_n = k \sum_{n=b_1}^{b_2} x_n \qquad (2-12)$$

En este trabajo se usa como porcentaje de energía total el 90% y como características la media y la desviación típica del roll-off espectral.

- coeficientes cepstrales de frecuencia de Mel o MFCC (MelFrequencyCepstralCoefficient): son coeficientes para la representación del habla basados en la percepción auditiva humana. Están diseñados para capturar características basadas en el espectro a corto plazo. Después de realizar el logaritmo del espectro de amplitud basado en STFT (transformada de Fourier de tiempo reducido) para cada frame, los bins de frecuencia se agrupan y suavizan de acuerdo con la escala de frecuencia de Mel. Los MFCC se generan decorrelacionando los vectores espectrales de Mel usando una transformada de coseno discreta (DCT).

Con la señal $x(n)$, la ventana $w(n)$, el eje de frecuencia w y el desplazamiento m , se calcula mediante:

$$STFT\{x(n)\}(m,w) = \sum_{n} x(n)w(n-m)e^{-jwn} \qquad (2-13)$$

Los pasos a seguir para el cálculo de estos coeficientes serían, por tanto, los siguientes:

1- Se separa la señal en pequeños tramos.

2- A cada tramo, se le aplica la DFT y se obtiene la potencia espectral de la señal.

$$X_k = \sum_{n=o}^{N-1} x_k e^{-\frac{2\pi i}{N}kn} \qquad k = 0, \dots, N-1 \qquad (2-14)$$

3- Se aplica el banco de filtros correspondiente a la escala Mel al espectro obtenido en el paso anterior y se suman las energías en cada uno de ellos.

4- Se toma el logaritmo de todas las energías de cada frecuencia Mel.

5- Se aplica la DCT a esos logaritmos.

$$X_k = \sum_{n=0}^{N-1} x_n cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, \dots, N-1 \qquad (2-15)$$

En este trabajo se usa la media y la desviación típica de los trece primeros coeficientes. También se utiliza la media y la desviación típica de los trece primeros coeficientes cepstrales fraccionales de frecuencia de Mel (FrCC). Estos coeficientes son calculados de la misma forma que los MFCCs pero usando la transformada fraccional de Fourier (FRFT).

- HR (Harmonic Ratio): mide la cantidad de energía en la parte tonal de la señal en comparación con la cantidad de energía de la señal total. Se calcula mediante la siguiente ecuación, donde x es una única muestra de audio con N elementos y m es el retraso máximo:

$$\Gamma(m) = \frac{\sum_{n=1}^{N} x(n)x(n-m)}{\sqrt{\sum_{n=1}^{N} x(n)^2 \sum_{n=0}^{N} x(n-m)^2}} \; para\ (1 \le m \le M) \qquad (2-16)$$

  Se usa como características su media y su desviación típica.

- frecuencia fundamental: es la frecuencia de vibración, es decir, es el número de repeticiones por unidad de tiempo. Se calcula a partir de la siguiente fórmula:

$$f_0 = {}^{F_s}/_{\gamma} \qquad (2-17)$$

  donde F_s es la frecuencia de muestreo y γ es el índice que corresponde al valor máximo de la autocorrelación normalizada.

  En este trabajo se utiliza la media y la desviación típica de dicha frecuencia.

- vector de croma: está relacionado con las doce diferentes clases de tonos (pitch). Para obtener este vector se agrupan los coeficientes DFT de una ventana de corto plazo en 12 bins. Cada bin representa una de las doce clases de tonos [40]. Se obtiene mediante la siguiente ecuación:

$$v_k = \sum_{n \in S_k} \frac{X_t(n)}{N_k}, \quad k \in 0, 1, \dots, 11 \qquad (2-18)$$

  donde S_k es un subconjunto de las frecuencias que corresponden a los coeficientes DFT y N_k es el número de elementos que contiene S_k. Se usa la media y la desviación típica de este vector como características.

- bpm (beats per minute): es una medida del tiempo en música. Se calcula de la siguiente forma:

$$BPM = \frac{60}{\frac{\sum_{n=2}^{N} x_n - x_{n-1}}{N}} \qquad (2-19)$$

  donde x es un vector de N pulsos consecutivos. Este valor es utilizado como característica.

3 ALGORITMO IMPLEMENTADO

El arte es la ciencia de la belleza, y las matemáticas son la ciencia de la verdad.

- Oscar Wilde -

En este capítulo se va a explicar, en primer lugar, los pasos para extraer las diferentes características explicadas en el capítulo anterior. Para ello se mostrará el código de Matlab utilizado para obtener los vectores de características que serán usados para realizar la clasificación. Posteriormente, se comentará el tipo de clasificador utilizado.

3.1 Código

A continuación, se van a explicar en las siguientes subsecciones el código utilizado para obtener los vectores de características que se usarán posteriormente para la clasificación de los géneros musicales.

Todo el código se ha realizado en Matlab con la versión R2018a. Algunas de las funciones utilizadas para la extracción de características han sido realizadas por Kamil Wojcicki.

A continuación se muestra un diagrama de flujo del código empleado para llevar a cabo la obtención de los vectores de características necesarios para la clasificación.

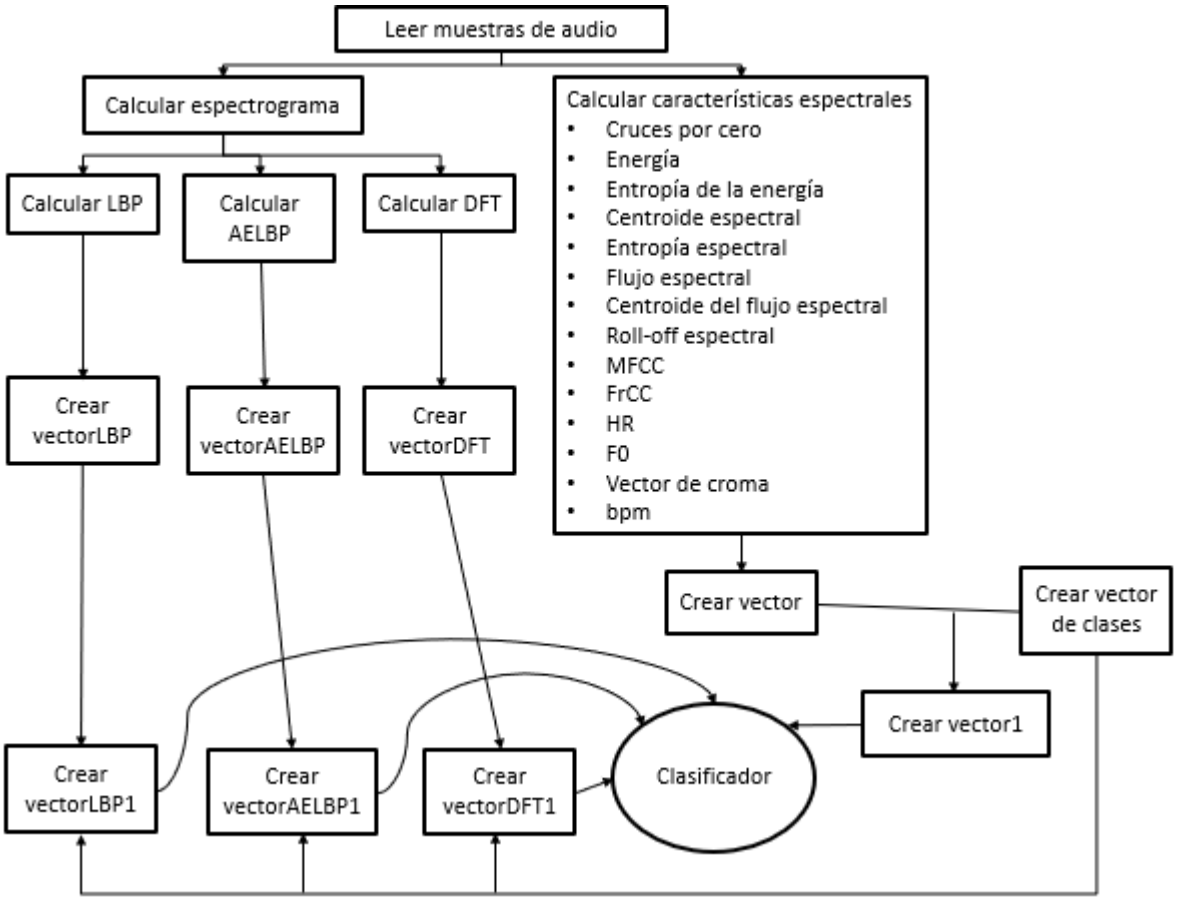

Figura 3-1.Diagrama de flujo.

Antes de obtener los vectores de características es necesario leer los archivos de audio de la base de datos que se va a utilizar para este trabajo.

Las canciones están separadas en carpetas según el género musical al que pertenecen, por lo que se ha creado un bucle para leer las 1000 canciones de la base de datos escogida.

En la siguiente imagen se muestra el código utilizado para recorrer las diferentes carpetas.

```
dataset_folder = 'genres';
subsets_names = {'blues','classical','country','disco','hiphop','jazz','metal','pop','reggae','rock'};
for subsetnum = 1:numel(subsets_names)
    genre_folder = fullfile(dataset_folder,subsets_names{subsetnum});
    genre_dir = dir(genre_folder);
    n = numel(genre_dir)-2;
    results_set = cell(numel(subsets_names),n,96);
    for singnum = 101:n
        file_name = fullfile(dataset_folder,subsets_names{subsetnum},genre_dir(singnum+2).name);
```

Figura 3-2. Lectura de audio.

Una vez indicada la carpeta en la que se encuentra la muestra de audio, se utiliza la función *audioread(filename)* de Matlab para su lectura.

```
[signal,fs] = audioread(file_name);
```

Figura 3-3. Lectura de audio.

Una vez que tenemos las señales de audio, ya se pueden extraer las características deseadas para formar los vectores de características y realizar la clasificación.

3.1.1 LBP

Debido a que el descriptor de textura LBP está pensado para trabajar con imágenes en vez de con señales de audio, es necesario hallar el espectrograma de las diferentes señales de audio a clasificar. Para ello se utiliza la función *spectrogram()* de Matlab.

```
[S,F,T,P] = spectrogram(signal, Window, Overlap, nFFT,
```

Figura 3-4. Espectrograma.

Los valores usados como parámetros de la función *spectrogram()* son los siguientes:

- signal es la señal de audio obtenida tras la lectura.
- Window es la ventana usada para dividir la señal en segmentos. Se ha usado 2048.
- Overlap es el número de muestras superpuestas. Se ha utilizado 512.
- nFFT es el número de puntos DFT. Se ha usado 2048.

Una vez obtenido el espectrograma de las señales de audio (P), se puede calcular el LBP de éstos. Aunque es necesario primero inicializar los valores de radio y vecindad que se van a usar.

En la siguiente imagen se muestran los valores utilizados para este trabajo.

```
% Radio y vecindad
R=3; PL=16;
```

Figura 3-5. Radio y vecindad.

Una vez inicializados los valores necesarios, se procede a calcular el descriptor LBP de cada muestra mediante la función *lbp()*, como se puede observar en la siguiente imagen. Esta función está realizada por Marko Heikkil y Timo Ahonen.

```
LBP=lbp(10*log10(abs(P)),R,PL,mapping,'h');
```

Figura 3-6. LBP.

La función *lbp()* tiene como parámetros el espectrograma, el radio, la vecindad, el mapping usado y 'h' que hace referencia a la obtención del histograma de los códigos LBP.

El mapping usado se ha obtenigo mediante la función *getmapping()*. El tipo de mapping es LBP uniforme invariante en rotación.

Se ha obtenido por tanto un vector LBP de dimensión 1x18 double con el histograma LBP del espectrograma de una muestra de audio.

Cuando ya se ha calculado el descriptor LBP se almacena su valor en el vector donde se van a recoger todos los valores LBP de cada una de las 1000 muestras de audio.

En la siguiente imagen se muestra la línea de código que hace posible la creación del vector, donde subsetnum hace referencia a los distintos géneros musicales y singnum a las pistas de audio de cada género.

```
result_set_LBP(subsetnum,singnum-100,:)=[LBP];
```

Figura 3-7.Vector LBP.

El vector result_set_LBP tiene una dimensión de 10x100x18 double.

3.1.2 AELBP

En el caso del descriptor AELBP se va a hacer uso de parte del código usado para el cálculo del LBP, puesto que es una variante de éste y tienen muchos pasos en común para su cálculo.

Al igual que en el caso del descriptor LBP, lo primero sería calcular el espectrograma (P) ya que trabaja con imágenes. Pero no va a ser necesario, así como inicializar valores como el radio y la vecindad, porque ya se realizó para el cálculo de LBP.

En las siguientes imágenes se muestra el código usado para calcular el descriptor AELBP de una muestra de audio.

```
%AELBP
[AECLBP_S,AECLBP_M,AECLBP_C] = AE_clbp(10*log10(abs(P)),R,PL,...
    patternMappingriu2,'x');
```

Figura 3-8. AELBP.

La función tiene como parámetros el espectrograma, el radio, la vecindad, el mapping usado y 'x' que hace referencia a que no se obtiene el histograma de los códigos AELBP.

El mapping usado se ha obtenigo mediante la función *getmapping()*, al igual que en el caso del descriptor LBP. El tipo de mapping es LBP uniforme invariante en rotación.

```
%Generate histogram of AECLBP_S
AECLBP_SH = hist(AECLBP_S(:),0:patternMappingriu2.num-1);
```

Figura 3-9. AELBP_SH.

```
%Generate histogram of AECLBP_M
AECLBP_MH= hist(AECLBP_M(:),0:patternMappingriu2.num-1);
```

Figura 3-10. AELBP_MH.

Tras generar el histograma se obtiene un vector AECLBP_SH y AECLBP_MH, cada uno de dimensión 1x18 double correspondiente a un muestra de audio.

Una vez calculado el AELBP de una muestra de audio se recopila en el vector usado para la clasificación que recogerá los resultados de todas las muestras. Como se ha comentado anteriormente, subsetnum hace referencia a los distintos géneros musicales y singnum a las pistas de audio de cada género.

```
resultset_AELBP(subsetnum,singnum-100,:)=[AECLBP_SH  AECLBP_MH];
```

Figura 3-11.Vector AELBP.

El vector resultset_AELBP tiene una dimensión de 10x100x36 double. La dimensión 36 se obtiene tras unificar los dos vectores de 18 elementos.

3.1.3 DFT

Para el cálculo de la DFT se va a utilizar la variable S devuelta por la función *spectrogram()* de Matlab. Esta variable es la transformada de Fourier a corto plazo, devuelta como una matriz. La variable S tiene una dimensión de 1025x430 complex double.

En la siguiente imagen se muestra la línea de código que permite obtener el valor de la DFT para cada muestra de audio.

```
DFT_coeff=mean(abs(S),2)./sum(mean(abs(S),2));
```

Figura 3-12.DFT.

En la siguiente imagen se muestra el vector que recogerá todos los valores DFT de todas las muestras de audio, donde subsetnum hace referencia a los distintos géneros musicales y singnum a las pistas de audio de cada género.

```
result_set_DFT(subsetnum,singnum-100,:)=[DFT_coeff];
```

Figura 3-13. Vector DFT.

El vector result_set_DFT tiene una dimensión de 10x100x1025 double.

3.1.4 Características Espectrales

Antes de calcular las diferentes variables para este vector, se divide la señal de audio en pequeños frames.

A cada uno de estos frames se le hallarán las características enumeradas en el capítulo anterior.

El primer paso para la obtención del vector de características es inicializar todas las variables que se van a utilizar para la clasificación. Estas variables están listadas en el capítulo anterior.

La variable numOfFrames es el número de frames en los que se ha dividido cada muestra de audio. Tiene un valor de 430 obtenido a partir de la longitud de la señal de audio y la ventana de 2048 escogida

En la siguiente imagen se muestra la inicialización de todas las variables a usar.

```
ZCR = zeros(numOfFrames, 1);
Energy = zeros(numOfFrames, 1);
Entropyenergy = zeros(numOfFrames, 1);
ceps=zeros(numOfFrames,13);
Fracionalceps=zeros(numOfFrames,13);
H0=zeros(numOfFrames, 1);
F0=zeros(numOfFrames, 1);
CromaVector=zeros(numOfFrames, 12);
```

```
SpectralRollOff = zeros(numOfFrames, 1);
SpectraCentroid = zeros(numOfFrames, 1);
SpectralSpread = zeros(numOfFrames, 1);
SpectralEntropy = zeros(numOfFrames, 1);
SpectralFlux = zeros(numOfFrames, 1);
SpectralFluxCentroid = zeros(numOfFrames, 1);
```

Figura 3-14. Variables espectrales.

Una vez que están inicializadas todas las variables, se pueden empezar a calcular. En la siguiente imagen se muestra el código usado para su cálculo.

```
ZCR(i) = feature_zcr(frame);
Energy(i) = feature_energy(frame);
EntropyEnergy(i) = feature_energy_entropy(frame, 10);
```

```
[SpectralCentroid(i),SpectralSpread(i)] = feature_spectral_centroid(frameFFT, fs);
SpectralEntropy(i) = feature_spectral_entropy(frameFFT, 10);
SpectralFlux(i) = feature_spectral_flux(frameFFT, frameFFTPrev);
SpectralFluxCentroid(i) = feature_spectral_flux_centroid(frameFFT, PframeFFT,fs);
SpectralRollOff(i) = feature_spectral_rolloff(frameFFT, 0.90);
ceps(i,:) = feature_mfccs(frameFFT, mfccParams);
```

```
[HR(i), F0(i)] = feature_harmonic(frame, fs);
CromaVector(i,:)=feature_chroma_vector(frame, fs);
```

```
Fractionalceps(i,:) = feature_mfccs(abs(FaF(nFFT/2+1:end)), mfccParams);
```

```
%Beats per minute
[b,onsetenv,oesr,D,cumscore] = beat2(signal,fs);
bpm(singnum)=length(b)*60/b(end);
```

Figura 3-15.Características espectrales.

ZCR, Energy, EntropyEnergy, SpectralCentroid, SpectralEntropy, SpectralFlux, SpectralFluxCentroid, SpectralRollOff, HR y F0 tienen una dimensión de 430x1 double.

CromaVector tiene una dimensión de 430x12.

Fractionalceps y ceps tienen una dimensión de 430x13 double.

bpm tiene una dimensión de 1x100 double, siendo un valor por cada muestra de audio.

En la siguiente imagen se muestra el vector que recoge todas las características calculadas anteriormente.

```
resultset(subsetnum,singnum-100,:)=[mean(ZCR) std(ZCR) mean(Energy) ...
    std(Energy) mean(EntropyEnergy) std(EntropyEnergy)  mean(SpectralCentroid) ...
    std(SpectralCentroid) mean(SpectralEntropy) std(SpectralEntropy)...
    mean(SpectralFlux) std(SpectralFlux) mean(SpectralFluxCentroid) ...
    std(SpectralFluxCentroid) mean(SpectralRollOff) std(SpectralRollOff) mean(ceps) ...
    std(ceps) mean(HR) std(HR) mean(F0) std(F0) mean(CromaVector) std(CromaVector) ...
    mean(Fractionalceps) std(Fractionalceps) bpm(singnum)];
```

Figura 3-16. Vector características espectrales.

El vector resultset tiene una dimensión de 10x100x97 double.

3.2 Clasificador

Para realizar la clasificación se ha utilizado la aplicación ClassificationLearner de Matlab R2018a.

Figura 3-17. Clasificador.

Esta aplicación no acepta matrices de 3 dimensiones por los que los vectores obtenidos deben ser modificados, para que en vez de ser de 10x100 sean de 1000. Para ello, se hace uso de la función *reshape()* de Matlab. Esta función permite transformar una matriz de 3 dimensiones a un vector de 2 dimensiones.

A continuación se muestra el código utilizado para modificar las dimensiones de los vectores.

```
vectorLBP = reshape(result_set_LBP,[1000,18]);
```

```
vectorAELBP = reshape(resultset_AELBP,[1000,36]);
```

```
vectorDFT = reshape(result_set_DFT,[1000,1025]);
vector = reshape(resultset,[1000,97]);
```

Figura 3-18. Vectores de características.

Tras ejecutar el código mostrado en la imagen anterior, las dimensiones de los vectores son las siguientes:

- vectorLBP: 1000x18.
- vectorAELBP: 1000x36.
- vectorDFT: 1000x1025.
- vector: 1000x97.

Al clasificador es necesario indicarle el género musical al que pertenece cada muestra de audio. Cada fila de los vectores que se muestran en la imagen anterior corresponde a una canción diferente. Por lo que para indicar el género musical al que corresponde cada fila, se añade una columna a los vectores de características creados anteriormente. Esta columna tiene un valor entre 0 y 9, siendo cada número un género musical diferente. Así el clasificador puede entrenar el modelo de clasificación sabiendo qué canciones corresponden a qué género musical.

Para añadir dicha columna, se crea primero un vector de 10x1 con los valores de 0 a 9. Posteriormente, haciendo uso de la función *repmat()* de Matlab, se realizan copias del vector para conseguir una columna de 1000 valores que se pueda añadir a los vectores de la imagen anterior.

En la siguiente imagen se muestra el código para la creación del vector que se añadirá como la última columna de los vectores de características anteriores.

```
clase = [0 1 2 3 4 5 6 7 8 9]';
vectorclase = repmat(clase, [100 1]);
```

Figura 3-19.Vector de clases.

Una vez obtenido el vectorclase de dimensión 1000x1, se añade como la última columna de los vectores de características. El código para realizar esta operación es el siguiente:

```
vectorLBP1 = [vectorLBP vectorclase];
vectorAELBP1 = [vectorAELBP vectorclase];
vectorDFT1 = [vectorDFT vectorclase];
vector1 = [vector vectorclase];
```

Figura 3-20. Vectores de características.

Tras esta última operación, las dimensiones de los vectores de características son las siguientes:

- vectorLBP1: 1000x19.
- vectorAELBP1: 1000x37.
- vectorDFT1: 1000x1026.
- vector1: 1000x98.

Como se puede ver, los vectores poseen una nueva columna, pero manteniendo el número de filas.

Una vez obtenidos los vectores de características necesarios para utilizar la aplicación de Matlab ClassificationLearner, podemos introducir los valores en la aplicación.

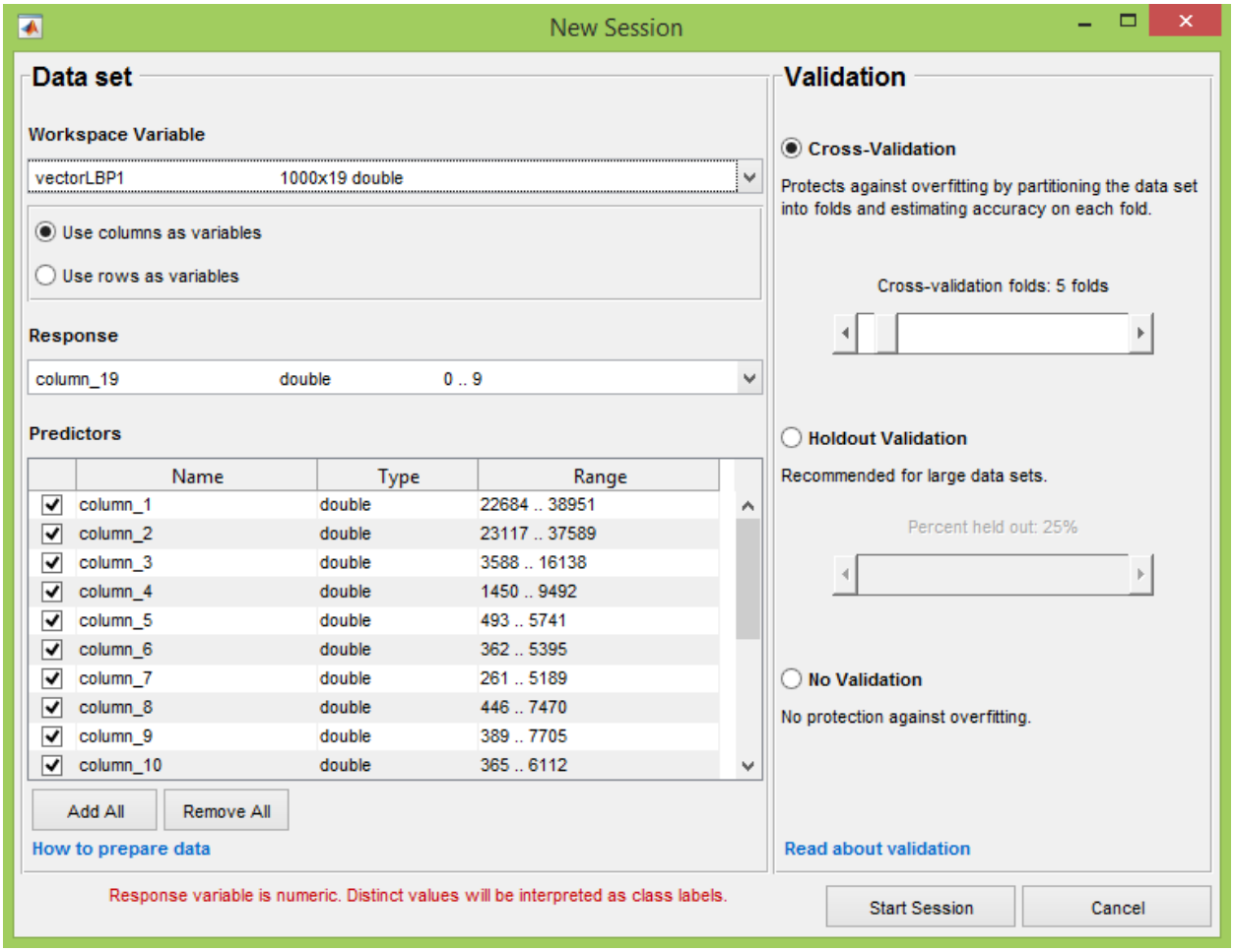

Figura 3-21. Aplicación de clasificación.

Como se muestra en la imagen anterior, se usa la última columna como la respuesta, es decir, su valor indica a qué género musical corresponde esa muestra de audio.

En este trabajo se ha decidido usar cada fila como una muestra de audio diferente y por tanto es una columna la que recoge la clase para la clasificación. Pero la aplicación permite versatilidad en cuanto al uso de filas o columnas.

En cuanto a la validación, se ha utilizado la validación cruzada (Cross-validation).

La validación cruzada es una técnica de validación de modelos para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Se utiliza principalmente en entornos donde el objetivo es la predicción, y se desea estimar con qué precisión se desempeñará en la práctica un modelo predictivo. En un problema de predicción, se proporciona generalmente un conjunto de datos conocidos a un modelo con los que se ejecuta el entrenamiento, y un conjunto de datos desconocidos con los que se prueba el modelo.

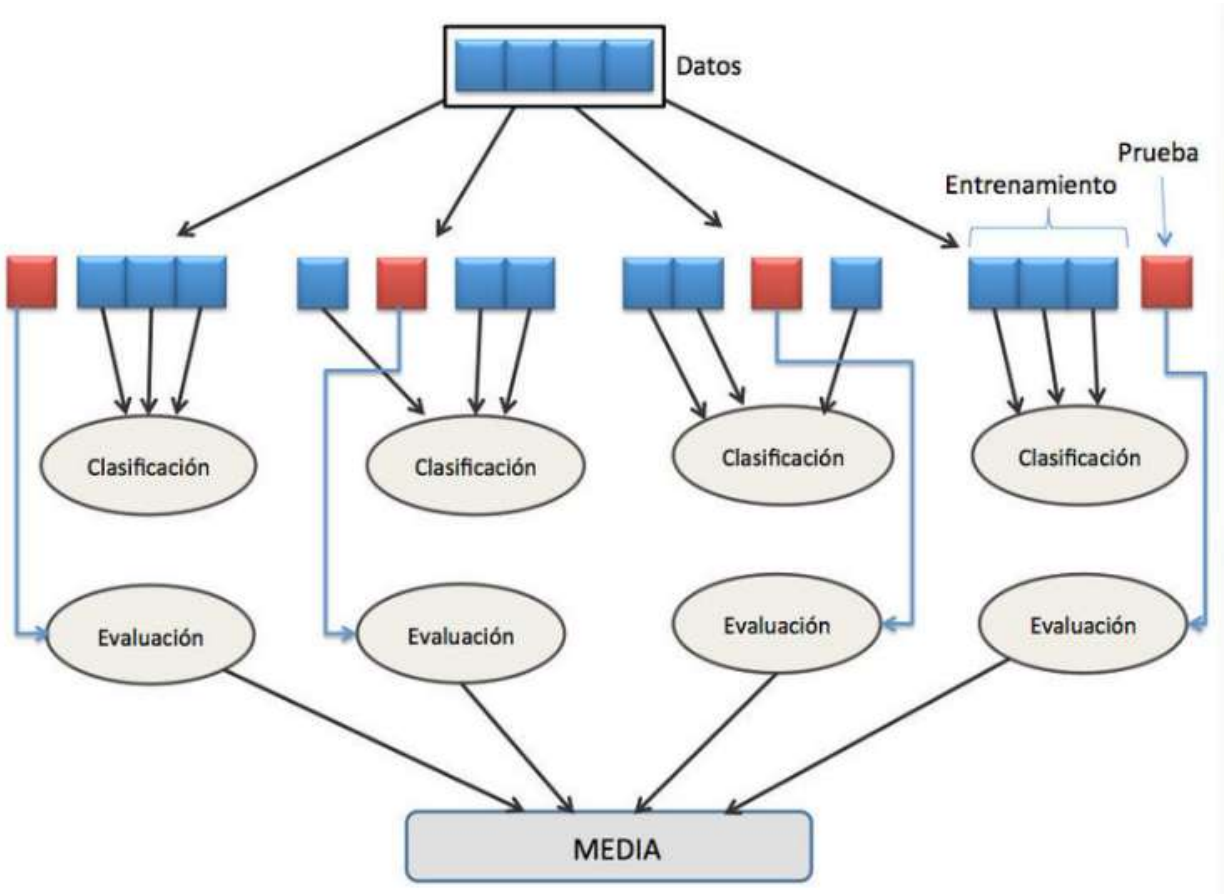

Figura 3-22. Esquema k-foldcross-validation [41].

En la imagen anterior, se muestra un esquema de la validación cruzada para un valor de k igual a 4.

En este tipo de validación, la muestra original se divide aleatoriamente en k submuestras de igual tamaño. De las k submuestras, se retiene una sola submuestra como datos de validación para probar el modelo y las restantes k-1 submuestras se usan como datos de entrenamiento. El proceso de validación cruzada se repite k veces, siendo usada cada una de las k submuestras una vez como datos de validación. Después, con los k resultados se calcula la media aritmética para obtener una única estimación.

En la práctica, la elección del número de interacciones, k, depende de la medida del conjunto de datos. Lo más común es utilizar una validación cruzada de 10 iteraciones.

En este trabajo se ha realizado una clasificación utilizando 5 y 10 iteraciones. En el siguiente capítulo se van a analizar los resultados obtenidos para estos valores de k y se analizará con cuál de ellos se han obtenido mejores resultados para cada vector de características.

4 CLASIFICADORES

Ninguna investigación humana puede ser denominada ciencia si no pasa a través de pruebas matemáticas.

- Leonardo Da Vinci -

En este capítulo se van a comentar los distintos clasificadores usados para la clasificación de géneros musicales. Se explicarán las características de cada uno de los modelos de clasificación utilizados. En total se han usado 17 modelos con cada vector de características.

Los modelos de clasificación son los siguientes:

- Fine Tree.
- Medium Tree.
- Coarse Tree.
- Linear Discriminant.
- Quadratic Discriminant.
- Linear SVM.
- Quadratic SVM.
- Cubic SVM.
- Fine Gaussian SVM.
- Medium Gaussian SVM.
- Coarse Gaussian SVM.
- Fine KNN.
- Medium KNN.
- Coarse KNN.
- Cosine KNN.
- Cubic KNN.
- Weighted KNN.

Estos modelos se pueden agrupar en cuatro tipos de clasificadores. A continuación, se van a explicar las características de cada uno de ellos.

4.1 Árboles de decisión

Se presentan como una alternativa a los métodos clásicos de clasificación donde lo que se infieren son reglas de decisión, que permiten realizar las clasificaciones en base al cumplimiento de premisas aprendidas y reflejadas en las reglas [42].

Existen varios tipos de árboles de decisión. Este trabajo se centra en los árboles de clasificación, en los que el resultado predicho es la clase a la que pertenecen los datos.

En estas estructuras de árbol, las hojas representan etiquetas de clases y las ramas representan las conjunciones de características que conducen a esas etiquetas de clase.

Dentro de este grupo se encuentran los siguientes tres modelos: Fine tree, Medium tree y Coarse tree. Estos clasificadores son fáciles de interpretar, rápidos para el ajuste y la predicción, y con poco uso de memoria. Pero pueden tener una precisión predictiva baja [43].

La diferencia entre estos modelos es el número de divisiones.

Un árbol de decisión toma decisiones dividiendo nodes en subnodos. Este proceso se realiza varias veces durante el proceso de entrenamiento hasta que solo quedan nodos homogéneos. Existen varias formas de realizar la división, que pueden agruparse en dos categorías dependiendo del tipo de variable de destino [44]:

- variable objetivo continua.
    - reducción de la varianza. Se utiliza para problemas de regresión. Hace uso de la varianza para calcular la homogeneidad de un nodo. Si un nodo es completamente homogéneo, entonces la varianza es nula.
- variable objetivo categórica:
    - impureza de Gini. Es la forma más popular y sencilla de dividir un árbol de decisión. Gini es la probabilidad de etiquetar correctamente un elemento elegido al azar si se etiquetó al azar de acuerdo con la distribución de etiquetas en el nodo. Cuanto menor sea la impureza de Gini, mayor es la homgeneidad del nodo. Se prefiere la impureza de Gini a la ganancia de información porque no contiene logaritmos que sean computacionalmente intensivos.
    - ganancia de información. Utiliza la entropía para calcular la pureza de un nodo. Cuanto menor es la entropía, mayor es su pureza. La entropía de un nodo homogéneo es cero.
    - chi-square. Trabaja sobre la importancia estadística de las diferencias entre el nodo padre y los nodos secundarios. Cuanto mayor sea el valor de chi-square, mayores serán las diferencias entre el nodo padre y el nodo hijo, es decir, mayor será la homogeneidad.

En este trabajo se ha utilizado la impureza de Gini como criterio de división.

El número máximo de divisiones o puntos de ramificación establecido para controlar la profundidad del árbol en los modelos usados en este trabajo ha sido:

- Fine tree: 100.
- Medium tree: 20.
- Coarse tree: 4.

4.2 Análisis discriminante

En este grupo se encuentran dos modelos: linear discriminant y quadratic discriminant. El análisis discriminante es un algoritmo popular de primera clasificación porque es rápido, preciso y fácil de interpretar. Este análisis es bueno para conjuntos de datos grandes [43].

El análisis discriminante asume que diferentes clases generan datos basados en distintas distribuciones gaussianas. Este análisis encuentra un conjunto de ecuaciones de predicción basadas en variables independientes que se usan para clasificar individuos en grupos.

El entrenamiento puede fallar si hay predictores con varianza nula o si alguna de las matrices de covarianza de los predictores es singular.

El linear discriminant es un clasificador estadístico que encuentra una combinación lineal de rasgos que separan dos o más clases.

El quadratic discriminant es un clasificador estadístico que utiliza una superficie de decisión cuadrática para separar los datos en dos o más clases.

4.3 SVM

Las máquinas de vectores de soporte (SVM) son modelos de aprendizaje supervisado con algoritmos de aprendizaje asociados que analizan los datos utilizados para el análisis de clasificación. Presenta uno de los métodos de predicción más robusto basado en el marco de aprendizaje estadístico.

Un clasificador SVM cataloga los datos al encontrar el mejor hiperplano que separa los puntos de datos de una clase de los de otra. El mejor hiperplano es el que tiene el mayor margen entre dos clases, es decir, el que tiene el máximo ancho entre dos puntos de distintas clases.

Los vectores de soporte son los puntos de datos más cercanos al hiperplano de separación.

Dentro de este grupo se hallan los siguientes modelos: linear SVM, quadratic SVM, cubic SVM, fine Gaussian SVM, médium Gaussian SVM y coarse Gaussian SVM.

Se puede entrenar SVM cuando los datos pertenecen a dos o más clases. Estos tipos de clasificadores son rápidos en la predicción en caso de dos clases y lentos cuando hay más de dos clases. El linear SVM tiene un uso de memoria medio sin importar el número de clases. Mientras que los demás tiene un uso de memoria medio para solo dos clases y grande cuando el número de clases es mayor a dos. La interpretabilidad del linear SVM es fácil, mientras que los otros modelos tienen una interpretabilidad difícil [43].

La diferencia entre los distintos modelos pertenecientes a este grupo radica en el hiperplano de separación. Por ejemplo, en el caso de linear SVM, el hiperplano es una función lineal y en el caso de quadratic SVM, el hiperplano es un polinomio cuadrático.

En el caso de clasificadores no lineales, se usan funciones kernel que permiten convertir un problema de clasificación no lineal en el espacio dimensional original en un sencillo problema de clasificación lineal en un espacio dimensional mayor. Existen varios tipos de funciones kernel.

En este trabajo se han usado la polinomial homogénea y la función gaussiana.

Los clasificadores SVM siempre toman la siguiente forma:

$$f(x) = \sum_n \alpha_n K(x, n) \qquad (4-1)$$

donde $K(x, n)$ es la matriz kernel.

La capacidad de aproximación y generalización de la SVM está determinadad por la elección del kernel [45].

4.4 Clasificadores Nearest Neighbor

El algoritmo k-nearest neighbor es un método no paramétrico utilizado para clasificación supervisada. La entrada consta de los k ejemplos de entrenamiento más cercanos en el espacio de características, del elemento que se desea clasificar y del conjunto de clases. La salida es una de las clases.

Los algoritmos basados en NN se utilizan ampliamente como reglas de aprendizaje automático de referencia.

Dado un conjunto de clases, los nuevos puntos a clasificar se asignan a aquella clase que posea un número de k vecinos más cercanos. Dicha proximidad se determina mediante alguna medida de similitud, por ejemplo, una distancia. Este tipo de clasificadores categoriza los puntos de consulta en función de su distancia a los puntos (o vecinos) en un conjunto de datos de entrenamiento. Se pueden utilizar diferentes métricas para determinar la distancia [46].

En este trabajo se ha usado distancia euclídea, distancia minkowski y uno menos el coseno del ángulo incluido entre observaciones (tratados como vectores).

La distancia euclídea es la distancia obtenida a partir del teorema de Pitágoras.

La distancia minkowski es una métrica en un espacio vectorial normalizado que puede considerarse como una generalización tanto de la distancia euclídea como de la distancia de Manhattan.

Una vez calculados los k vecinos más cercanos del elemento a clasificar, se decide a qué clase pertenece. Para ello, los k vecinos deben votar por una de las clases, siendo la clase más votada a la que pertenece el nuevo elemento. Hay distintas formas de hacerlo, el método del voto donde el voto de todos los vecinos vale lo mismo o puede ser que el voto sea ponderado. En el caso del voto ponderado, cada vecino vota con un peso diferente dependiendo de la distancia que hay entre el elemento a clasificar y el vecino. El voto será mayor cuanto menor sea la distancia.

En este trabajo se han usado los dos métodos, el del voto y el del voto ponderado. En el caso del voto ponderado, el peso correspondía $1/_{distancia^2}$.

Están en este grupo los clasificadores: fine KNN, médium KNN, coarse KNN, cosine KNN, cubic KNN y weighted KNN.

Estos clasificadores suelen tener una buena precisión predictiva en dimensiones bajas, pero es posible que no en dimensiones altas. Tienen un alto uso de memoria y no son fáciles de interpretar. La velocidad de predicción de cubic KNN es menor que en los demás modelos [43].

La diferencia entre los distintos modelos de clasificadores radica en el número de vecinos más cercanos que se deben encontrar para clasificar cada punto. Por ejemplo, fine KNN tiene un número menor de vecinos que coarse KNN.

Si el número de vecinos es muy pequeño, el resultado puede ser sensible a supuestos vecinos que en realidad son ruido. Si este valor es muy alto, se podrían incluir vecinos que pertenecen a otra clase diferente.

A continuación, se especifica los valores de k, el tipo de distancia y el peso para cada modelo utilizado:

- Fine KNN: se ha usado un vecino, distancia euclídea y sin peso.
- Medium KNN: se han utilizado 10 vecinos, distancia euclídea y sin peso.
- Coarse KNN: se han usado 100 vecinos, distancia euclídea y sin peso.
- Cosine KNN: se han utilizado 10 vecinos. como distancia se ha tomado uno menos el coseno del ángulo incluido entre observaciones (tratados como vectores) y sin peso.
- Cubic KNN: se han usado 10 vecinos. distancia minkowski de orden 2 y sin peso.
- Weighted KNN: se han utilizado 10 vecinos, distancia euclídea y peso de $1/_{distancia^2}$.

5 RESULTADOS

Un experto es una persona que ha cometido todos los errores posibles en un campo muy pequeño.

- Niels Bohr -

En este capítulo se van a exponer los resultados obtenidos tras la clasificación de las canciones mediante el uso de las diferentes características explicadas en capítulos anteriores. También se comentará la base de datos usada para la clasificación y los tipos de clasificadores utilizados.

5.1 Base de datos

Para el desarrollo de este trabajo, se han utilizado los siguientes 10 géneros musicales para la clasificación:

- Blues.
- Clásico.
- Country.
- Disco.
- Hip hop.
- Jazz.
- Metal.
- Pop.
- Reggae.
- Rock.

Cada género musical consta de 100 canciones de 30 segundos cada una. La tasa de muestreo es de 22050 muestras/s.
La base de datos es GTZAN genre collection [42].

5.2 Resultados

Para realizar la clasificación se ha utilizado la aplicación ClassificationLearner de Matlab R2018a. Aunque cabe indicar que dentro de la aplicación se han probado diferentes tipos de modelos de clasificación para hallar en cada caso con cuál de ellos se obtenían mejores resultados dependiendo de cada vector de características utilizado.

Para cada uno de los cuatro vectores de características analizados en este trabajo se han realizado diferentes pruebas de clasificación con distintos tipos de modelos de clasificación y con una validación cruzada de 5 y 10 iteraciones para cada uno de los modelos.

En total se han probado 17 tipos de modelos de clasificación para cada vector de características, como ya se ha comentado en el capítulo anterior. Los modelos de clasificación son los siguientes:

- Fine Tree.
- Medium Tree.
- CoarseTree.
- Linear Discriminant.
- QuadraticDiscriminant.
- Linear SVM.
- Quadratic SVM.
- Cubic SVM.
- Fine Gaussian SVM.
- Medium Gaussian SVM.
- Coarse Gaussian SVM.
- Fine KNN.
- Medium KNN.
- Coarse KNN.
- Cosine KNN.
- Cubic KNN.
- Weighted KNN.

En las siguientes tablas se muestran para cada vector de características los porcentajes obtenidos con cada uno de los tipos de modelo de clasificación.

Tabla 5–1. Vector LBP con 5 iteraciones.

Tipo de Modelo	Precisión
Fine tree	39,5%
Medium tree	35,1%
Coarsetree	30,0%
Linear discriminant	49,0%
QuadraticDiscriminant	Fallo
Linear SVM	53,0%
Quadratic SVM	**56,2%**
Cubic SVM	52,5%
Fine Gaussian SVM	50,0%
Medium Gaussian SVM	50,7%
CoarseGaussian SVM	42,0%
Fine KNN	45,3%

Medium KNN	47,2%
Coarse KNN	41,7%
Cosine KNN	46,1%
Cubic KNN	47,3%
Weighted KNN	50,0%

Tabla 5–2. Vector LBP con 10 iteraciones.

Tipo de Modelo	Precisión
Fine tree	39,7%
Medium tree	33,5%
Coarsetree	29,7%
Linear discriminant	49,0%
QuadraticDiscriminant	Fallo
Linear SVM	53,3%
Quadratic SVM	**56,9%**
Cubic SVM	54,2%
Fine Gaussian SVM	50,0%
Medium Gaussian SVM	51,0%
CoarseGaussian SVM	43,3%
Fine KNN	46,2%
Medium KNN	47,3%
Coarse KNN	41,4%
Cosine KNN	46,0%
Cubic KNN	47,1%
Weighted KNN	50,0%

Tabla 5–3. Vector AELBP con 5 iteraciones.

Tipo de Modelo	Precisión
Fine tree	37,4%
Medium tree	39,3%
Coarsetree	33,4%
Linear discriminant	Fallo
QuadraticDiscriminant	Fallo
Linear SVM	49,6%
Quadratic SVM	53,0%
Cubic SVM	52,6%
Fine Gaussian SVM	**55,9%**
Medium Gaussian SVM	49,8%
CoarseGaussian SVM	35,6%
Fine KNN	46,7%
Medium KNN	47,6%
Coarse KNN	43,0%

Cosine KNN	46,4%
Cubic KNN	47,8%
Weighted KNN	50,7%

Tabla 5–4. Vector AELBP con 10 iteraciones.

Tipo de Modelo	Precisión
Fine tree	42,1%
Medium tree	42,4%
Coarsetree	33,6%
Linear discriminant	Fallo
QuadraticDiscriminant	Fallo
Linear SVM	50,6%
Quadratic SVM	53,5%
Cubic SVM	53,4%
Fine Gaussian SVM	**55,5%**
Medium Gaussian SVM	51,0%
CoarseGaussian SVM	36,0%
Fine KNN	48,4%
Medium KNN	47,3%
Coarse KNN	42,3%
Cosine KNN	47,0%
Cubic KNN	46,7%
Weighted KNN	52,5%

Tabla 5–5. Vector DFT con 5 iteraciones.

Tipo de Modelo	Precisión
Fine tree	40,3%
Medium tree	36,3%
Coarsetree	26,8%
Linear discriminant	34,9%
QuadraticDiscriminant	Fallo
Linear SVM	61,6%
Quadratic SVM	**65,4%**
Cubic SVM	65,0%
Fine Gaussian SVM	35,2%
Medium Gaussian SVM	62,0%
CoarseGaussian SVM	48,2%
Fine KNN	52,3%
Medium KNN	51,1%
Coarse KNN	40,4%
Cosine KNN	49,5%
Cubic KNN	49,5%

Weighted KNN	54,0%

Tabla 5–6. Vector DFT con 10 iteraciones.

Tipo de Modelo	Precisión
Fine tree	41,2%
Medium tree	34,5%
Coarsetree	26,5%
Linear discriminant	31,5%
QuadraticDiscriminant	Fallo
Linear SVM	64,0%
Quadratic SVM	**68,7%**
Cubic SVM	68,5%
Fine Gaussian SVM	38,7%
Medium Gaussian SVM	63,8%
CoarseGaussian SVM	48,3%
Fine KNN	53,8%
Medium KNN	50,9%
Coarse KNN	41,6%
Cosine KNN	50,6%
Cubic KNN	49,9%
Weighted KNN	54,5%

Tabla 5–7. Vector espectral con 5 iteraciones.

Tipo de Modelo	Precisión
Fine tree	58,1%
Medium tree	54,8%
Coarsetree	39,0%
Linear discriminant	**85,2%**
QuadraticDiscriminant	Fallo
Linear SVM	79,7%
Quadratic SVM	81,8%
Cubic SVM	80,5%
Fine Gaussian SVM	22,6%
Medium Gaussian SVM	78,9%
CoarseGaussian SVM	68,2%
Fine KNN	67,6%
Medium KNN	68,6%
Coarse KNN	56,3%
Cosine KNN	65,9%
Cubic KNN	69,4%
Weighted KNN	71,9%

Tabla 5–8. Vector espectral con 10 iteraciones.

Tipo de Modelo	Precisión
Fine tree	60,5%
Medium tree	55,2%
Coarsetree	39,5%
Linear discriminant	**86,7%**
QuadraticDiscriminant	Fallo
Linear SVM	82,9%
Quadratic SVM	83,7%
Cubic SVM	82,3%
Fine Gaussian SVM	24,3%
Medium Gaussian SVM	81,5%
CoarseGaussian SVM	71,0%
Fine KNN	69,2%
Medium KNN	70,8%
Coarse KNN	58,9%
Cosine KNN	67,3%
Cubic KNN	70,8%
Weighted KNN	72,5%

En la siguiente tabla se muestran los mejores porcentajes obtenidos en la clasificación utilizando los diferentes vectores de características.

Tabla 5–9. Vectores de características.

Tipo de Vector	Precisión
LBP	56,9%
AELBP	55,9%
DFT	68,7%
Espectral	**86,7%**

Como se puede observar en la tabla anterior, se obtiene un porcentaje claramente superior al utilizar el vector de características espectrales.

Hay que indicar que, para los resultados de la tabla anterior, en el caso del vector LBP y DFT se ha usado Quadratic SVM con 10 iteraciones; en el caso de AELBP se ha utilizado Fine Gaussian SVM con 5 iteraciones; y en el caso de las características espectrales se ha usado Linear Discriminant con 10 iteraciones.

En la siguiente tabla se encuentran todos los resultados obtenidos para cada uno de los vectores de características utilizados, para el caso de 5 iteraciones.

Tabla 5–10. Resultados con 5 iteraciones.

Tipo de Modelo	LBP	AELBP	DFT	**Espectral**
Fine tree	39,5%	37,4%	40,3%	58,1%
Medium tree	35,1%	39,3%	36,3%	54,8%
Coarsetree	30,0%	33,4%	26,8%	39,0%
Linear discriminant	49,0%	Fallo	34,9%	**85,2%**
QuadraticDiscriminant	Fallo	Fallo	Fallo	Fallo
Linear SVM	53,0%	49,6%	61,6%	79,7%
Quadratic SVM	56,2%	53,0%	65,4%	81,8%
Cubic SVM	52,5%	52,6%	65,0%	80,5%
Fine Gaussian SVM	50,0%	55,9%	35,2%	22,6%
Medium Gaussian SVM	50,7%	49,8%	62,0%	78,9%
CoarseGaussian SVM	42,0%	35,6%	48,2%	68,2%
Fine KNN	45,3%	46,7%	52,3%	67,6%
Medium KNN	47,2%	47,6%	51,1%	68,6%
Coarse KNN	41,7%	43,0%	40,4%	56,3%
Cosine KNN	46,1%	46,4%	49,5%	65,9%
Cubic KNN	47,3%	47,8%	49,5%	69,4%
Weighted KNN	50,0%	50,7%	54,0%	71,9%

En la siguiente tabla se encuentran todos los resultados obtenidos para cada uno de los vectores de características utilizados, en el caso de 10 iteraciones.

Tabla 5–11. Resultados con 10 iteraciones.

Tipo de Modelo	LBP	AELBP	DFT	**Espectral**
Fine tree	39,7%	42,1%	41,2%	60,5%
Medium tree	33,5%	42,4%	34,5%	55,2%
Coarsetree	29,7%	33,6%	26,5%	39,5%
Linear discriminant	49,0%	Fallo	31,5%	**86,7%**
QuadraticDiscriminant	Fallo	Fallo	Fallo	Fallo
Linear SVM	53,3%	50,6%	64,0%	82,9%
Quadratic SVM	56,9%	53,5%	68,7%	83,7%
Cubic SVM	54,2%	53,4%	68,5%	82,3%
Fine Gaussian SVM	50,0%	55,5%	38,7%	24,3%
Medium Gaussian SVM	51,0%	51,0%	63,8%	81,5%
CoarseGaussian SVM	43,3%	36,0%	48,3%	71,0%
Fine KNN	46,2%	48,4%	53,8%	69,2%
Medium KNN	47,3%	47,3%	50,9%	70,8%
Coarse KNN	41,4%	42,3%	41,6%	58,9%
Cosine KNN	46,0%	47,0%	50,6%	67,3%
Cubic KNN	47,1%	46,7%	49,9%	70,8%
Weighted KNN	50,0%	52,5%	54,5%	72,5%

En la siguiente imagen se pueden observar los datos que aporta la aplicación de Matlab.Se muestra el tipo de modelo de clasificación utilizado junto al porcentaje de precisión obtenido. Además se muestra el tamaño del vector de características, compuesto por 1000 canciones (100 por cada uno de los 10 géneros), así como el número de características utilizadas. También se puede ver la velocidad de predicción y el tiempo de entrenamiento.

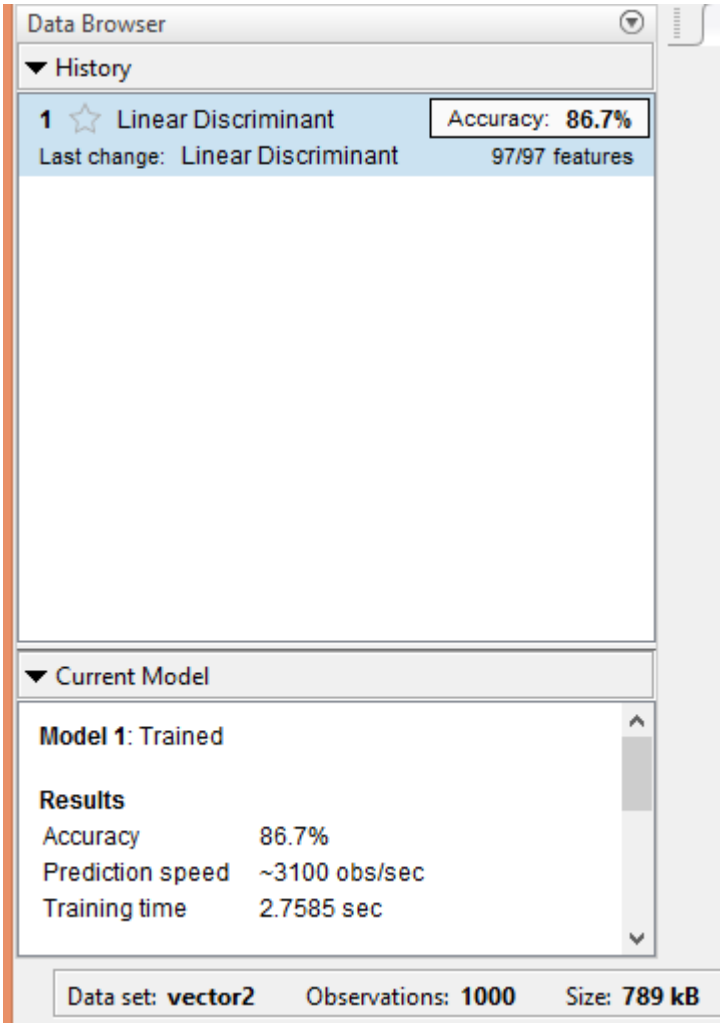

Figura 5-1. Resultado de características espectrales.

La aplicación, además de los datos anteriores, también aporta un gráfico de dispersión teniendo como ejes las columnas del vector de características.

Como en este trabajo se ha considerado formar vectores de características donde las filas corresponden a las diferentes canciones, el gráfico se realiza tomando como ejes las columnas. Pero en el caso de introducir en la aplicación las filas como variables, los ejes para el gráfico serían las filas del vector.

En el gráfico se puede apreciar representados mediante un círculo los datos correctos y mediante una cruz los incorrectos. Cada color simboliza una clase diferente.

Aparte del gráfico de dispersión, también permite visualizar la matriz de confusión, la curva ROC y las coordenadas paralelas.

La matriz de confusión es una herramienta que permite la visualización del desempeño de un algoritmo que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias de una clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases. Las celdas diagonales muestran

donde coinciden las clases reales y las predicciones. Si estas celdas son verdes, el clasificador ha funcionado bien y ha clasificado las observaciones en la clase real correctamente.

Matlab permite tres tipos de visualización:

- Number of observations: muestra el número de observaciones en cada celda. Es el usado por defecto.
- True Positive Rates False NegativeRates: muestra cómo el clasificador actúa por clase. En las últimas dos columnas de la derecha se muestran resúmenes por clase real.
- Positive PredictiveValues False Discovery Rates: muestra resultados de las predicciones. Debajo de la tabla aparecen dos filas como resúmenes. Los valores predictivos positivos se muestran en verde para los puntos correctamente predichos en cada clase, y las tasas de descubrimiento falsas se muestran en rojo para los puntos incorrectamente predichos en cada clase.

La curva ROCes la representación de la razón de verdaderos positivos (VPR) frente a la razón de falsos positivos (FPR) según se varía el umbral de discriminación (valor a partir del cual se decide que un caso es positivo).

Las coordenadas paralelas son una forma común de visualizar geometría de alta dimensión y analizar datos multivariados. Para mostrar un conjunto de puntos en un espacio n-dimensional, se dibuja un fondo que consta de n líneas paralelas, típicamente verticales e igualmente espaciadas. Un punto en el espacio n-dimensional se representa como una polilínea con vértices en los ejes paralelos. La posición de los vértices sobre el eje i-ésimo corresponde a la coordenada i-ésima del punto.

Otros datos que se pueden observar son el tipo de validación utilizada en la esquina inferior derecha, así como la columna del vector de características usada como indicación de la clase a la que pertenece cada muestra de audio.

En la siguiente imagen se muestra la matriz de confusión para el vector de características LBP. Se puede observar que la mejor clase clasificada con un 85% es música clásica y la peor con un 40% es rock que se confunde con country. También se confunde bastante blues con rock.

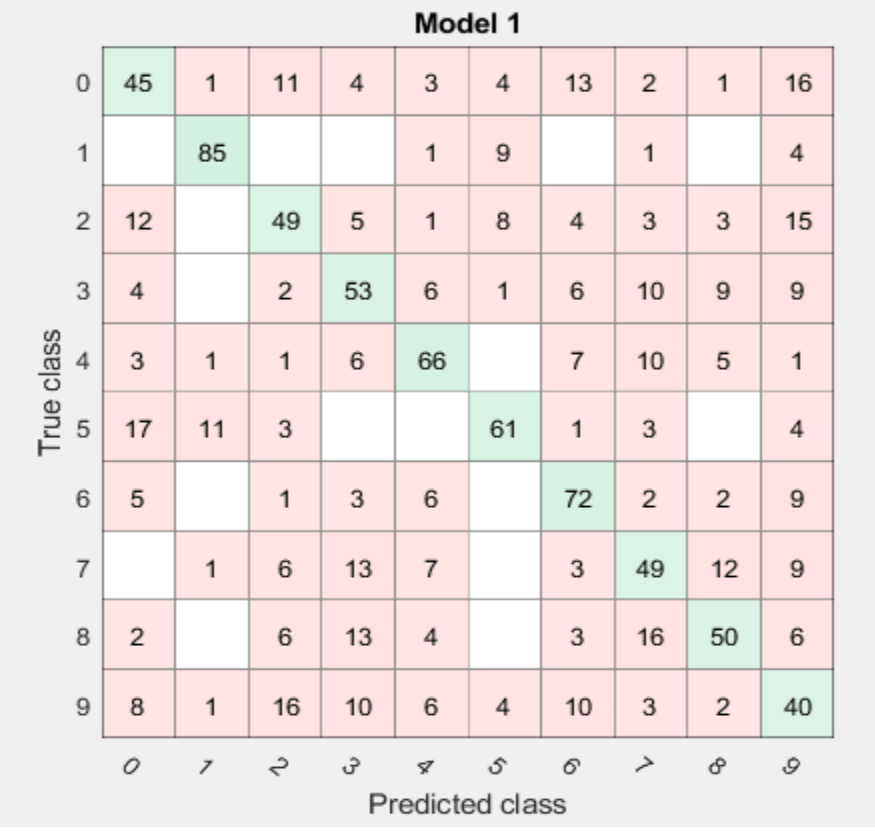

Figura 5-2. Resultado LBP.

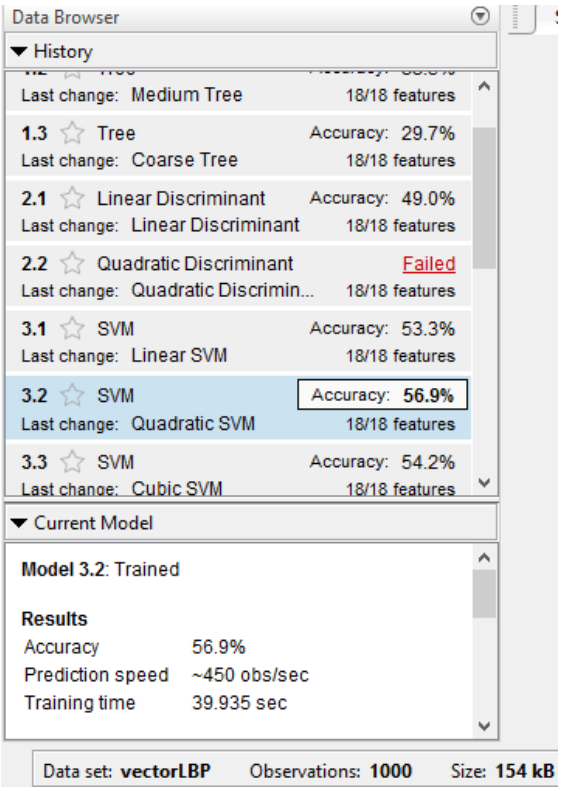

Figura 5-3. Resultado LBP.

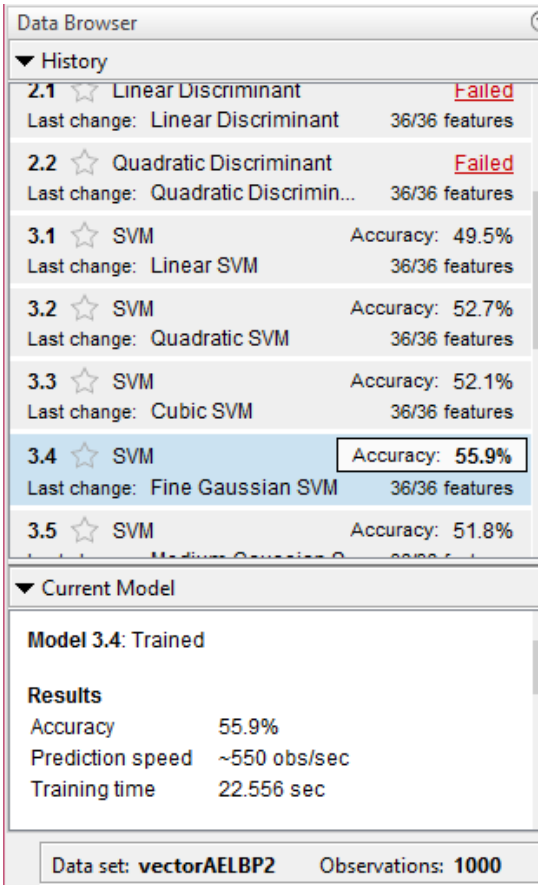

Figura 5-4. Resultado AELBP.

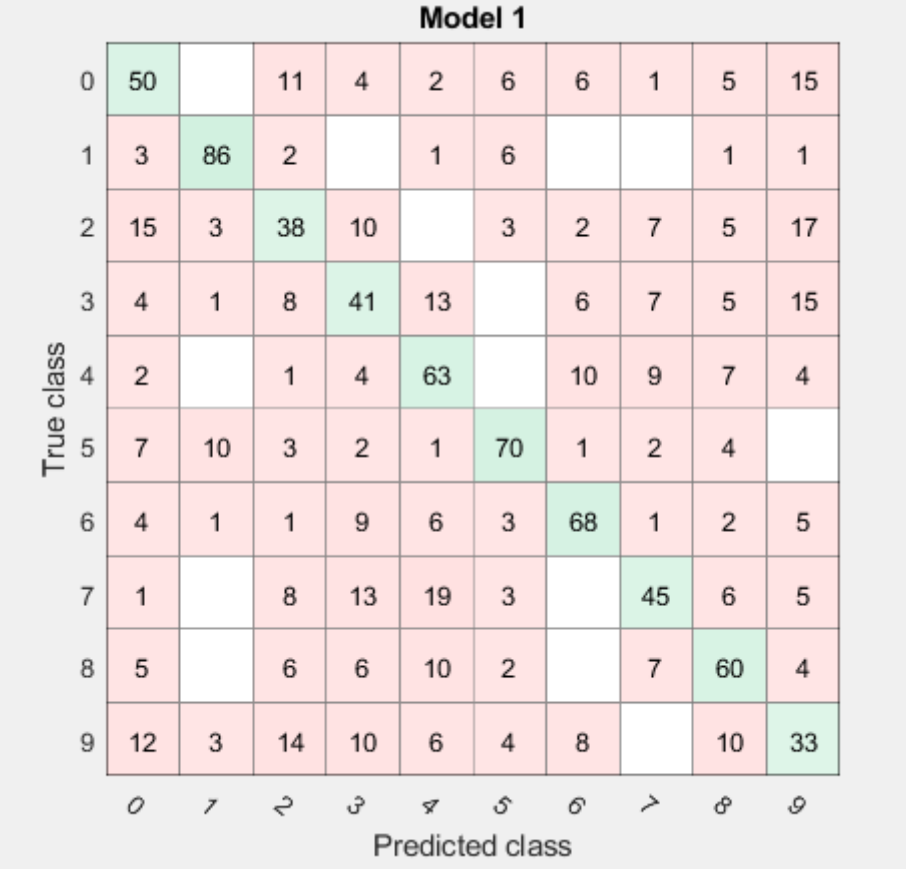

Figura 5-5. Resultado AELBP.

En la imagen anterior se observa la matriz de confusión para el vector de características AELBP. Como se puede ver, el mejor género musical clasificado es música clásica con un 86% y el peor es el rock, que se confunde con country, blues y disco.

Se puede comprobar que los resultados para este vector de características son parecidos a los obtenidos para el vector LBP.

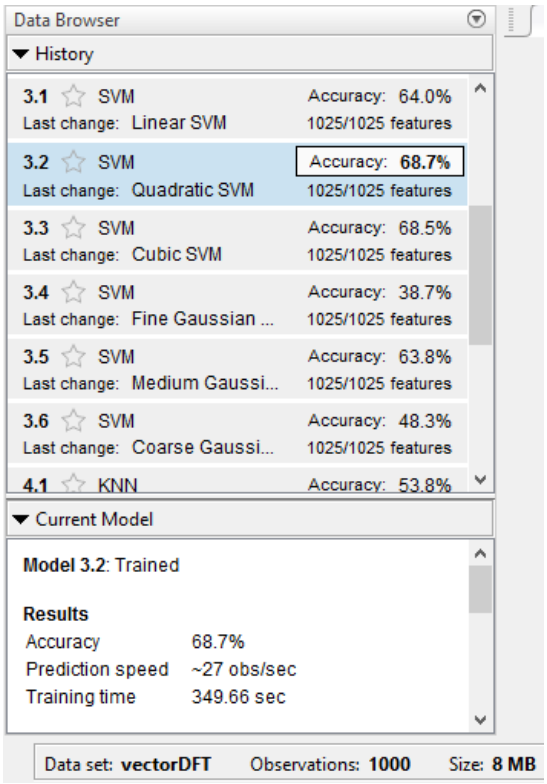

Figura 5-6. Resultado DFT.

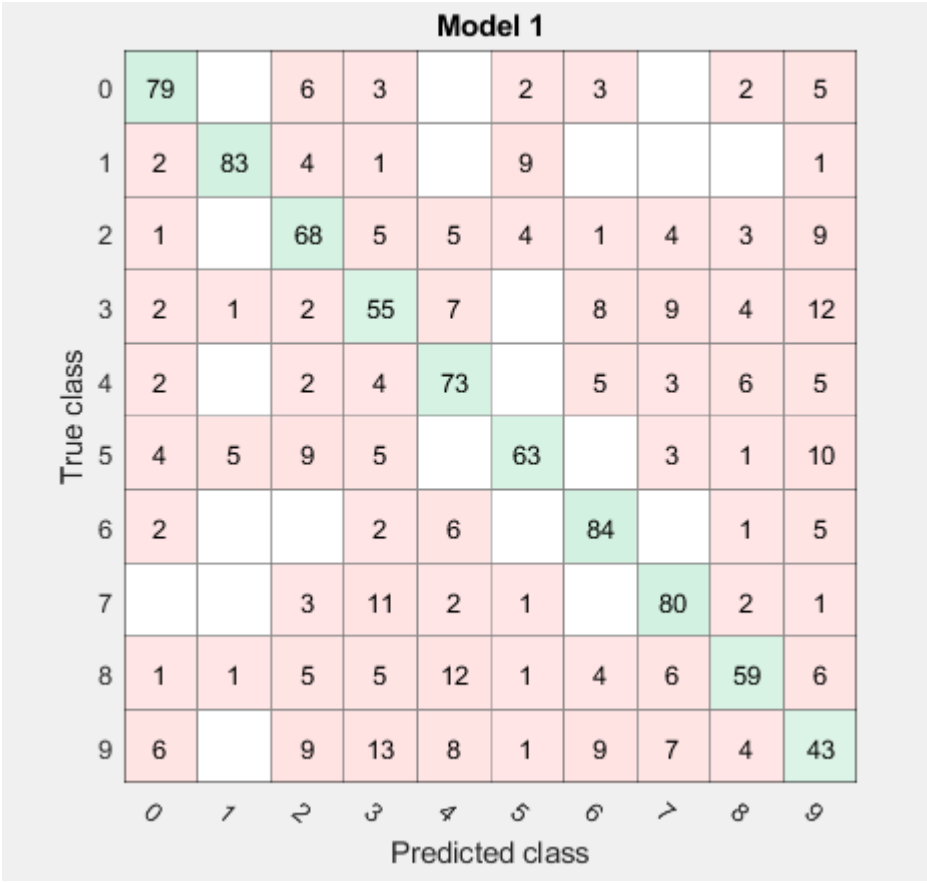

Figura 5-7. Resultado DFT.

En la imagen anterior se observa la matriz de confusión para el vector de características DFT. Se han obtenido buenos resultados para música clásica (83%), metal (84%) y pop (80%). Mientras que los peores resultados se alcanzan para rock (43%), confundiéndolo con disco.

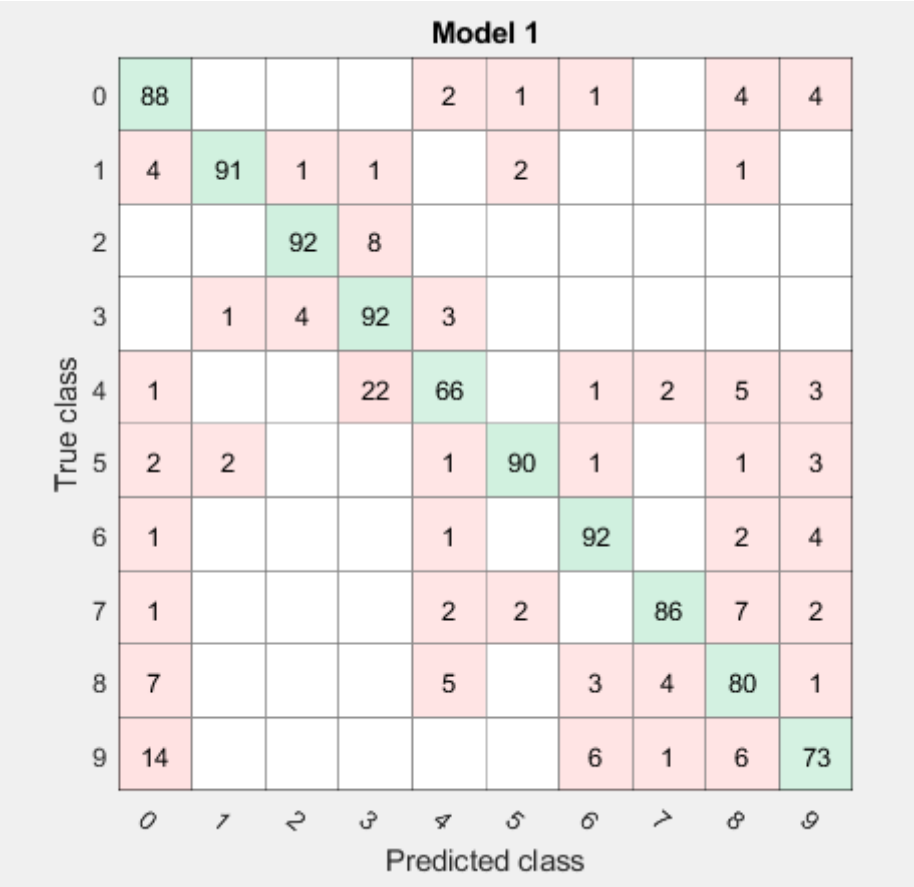

Figura 5-8. Resultado de características espectrales.

En la imagen de arriba se observa la matriz de confusión para el vector de características espectrales. Los mejores resultados con un 92% se obtienen para country, disco y metal. También se alcanza un 91% con música clásica y un 90% con jazz. El peor resultado se obtiene con hip hop (66%), confundiéndolo con disco, aunque sigue siendo mayor que los peores resultados de los demás vectores de características.

6 CONCLUSIÓN Y LÍNEAS FUTURAS

Si no cometes errores es porque no trabajas en problemas suficientemente difíciles. Y eso es un error.

- Frank Wilczek -

En este último capítulo, tras el desarrollo del trabajo, se realiza una conclusión y se aportan algunas posibles líneas de investigación futuras. En este trabajo se han estudiado diferentes vectores de características para la clasificación de 10 géneros musicales, así como distintos modelos de clasificación.

6.1 Conclusiones

Se puede concluir que el vector de características espectrales es el que proporciona mayor precisión frente a descriptores de textura como son LBP y AELBP, y frente al vector DCT para la clasificación en géneros musicales.

También se puede deducir que los descriptores de textura no trabajan bien con muestras de audio, ya que los resultados obtenidos en la clasificación para estos vectores de características rondan el 50%.

Además, se puede afirmar que el modelo de clasificación Quadratic discriminant no es buena elección debido a que una o más de las clases tiene matrices de covarianza singulares para sus valores predictores, es decir, el determinante de la matriz de covarianza es nulo. Esto hace que este tipo de entrenamiento de error.

A partir de las matrices de confusión se observa que los mejores resultados se alcanzan con el vector de características espectrales, obteniendo un 92% en country, disco y metal, un 91% en música clásica y un 90% en jazz. Con los descriptores de textura LBP y AELBP se consigue el mejor resultado para música clásica con un 85% y 86%, respectivamente. Por último, con el vector de características DFT se obtiene un 84% en metal, un 83% en música clásica y un 80% en pop.

Se puede concluir por tanto, que el mejor vector es el de características espectrales y que la música clásica es la que obtiene los mejores resultados para todos los vectores de características estudiados.

Los géneros musicales más confundidos son country, blues y disco utilizando los descriptores de textura. El vector de características DFT confunde rock con disco, y el vector de características espectrales confunde hip hop con disco.

6.2 Líneas futuras

En este trabajo no se han obtenido muy buenos resultados usando descriptores de textura para clasificación de géneros musicales. Sin embargo, se podría estudiar el uso de estos descriptores modificando los valores de radio y vecindad, así como el mapping utilizado para concluir si es o no una buena técnica para el análisis de audio y podría llegar a ofrecer resultados tan relevantes como en su uso en visión artificial.

Puede ser interesante trabajar con el espectrograma para el cálculo de los descriptores de textura, en vez de utilizar la densidad espectral de potencia (PSD).

Todos los resultados se han obtenido utilizando modelos de clasificación de la app Classification Learner de Matlab con la configuración por defecto de cada modelo. Luego se podrían modificar ciertos parámetros con el objetivo de averigurar si se ven afectados favorablemente estos resultados.

REFERENCIAS

[1] MIREX: https://www.music-ir.org/mirex/wiki/2020:Audio_Classification_(Train/Test)_Tasks. Úlimo acceso: 21/08/2020.

[2] Enlace: http://majorminer.org/info/intro. Úlimo acceso: 21/08/2020.

[3] MIREX: https://www.music-ir.org/mirex/wiki/2020:Audio_Tag_Classification. Úlimo acceso: 21/08/2020.

[4] MIREX: https://www.music-ir.org/mirex/wiki/2020:Music_Detection. Úlimo acceso: 21/08/2020.

[5] MIREX: https://www.music-ir.org/mirex/wiki/2018:Music_and/or_Speech_Detection. Úlimo acceso: 21/08/2020.

[6] Tao Lin, Mitsunori Ogihara, Qi Li. (Julio 2003). *A comparative study on content-based music genre classification*. SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval, pp. 282-289.

[7] B. Logan. (2000). *Mel frequency cepstral coefficient for music modeling*. Proceedings of the 1st international symposium on music information retrieval (ISMIR).

[8] M. Goto, Y. Muraoka. (1994). *A beat tracking system for acoustic signals of music*. Multimedia '94: Proceedings of the seoond ACM international conference on Multimedia, pp. 365-372.

[9] J. Laroche. (2001). *Estimating tempo, swing and beat locations in audio recording*. Proceedings of the 2001 IEEE Workshop on the applications of signal proceesing to audio and acoustics, pp. 135-138.

[10] E. Sheirer. (1998). *Tempo and beat analysis of acoustic music signals*. Journal of the acoustical society of america, vol. 103, nº 1, pp.588-601.

[11] J. Foote. (1997). *Content-based retrieval of music and audio*. Multimedia storage and archiving systems II, pp. 138-147.

[12] H. Ezzaidi, J. Rouat. (2006). *Automatic musical genre classification using divergence and average information measures*. Research report of the world academy of science, engineering and technology.

[13] L. Deng, D. O'shaugnessy. (2003). *Speech processing: a dynamic and optimization-oriented approach*.

[14] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, A. Linney. (1998). *Classification of audio signals using statistical features on time and wavelet transform domains*. Proceedings of the 1998 IEEE international conference on acoustics, speech and signal processing (ICASSP '98), vol. 6, pp. 3621-3624.

[15] H. Deshpande, R. Singh, U. Nam. (2001). *Classification of music signals in the visual domain*. Proceedings of the COST-G6 conference on digital audio effects.

[16] George Tzanetakis, Perry Cook. (Julio 2002). *Musical genre classification of audio signals*. IEEE Transactions on speech and audio processing, vol. 10, nº 5, pp. 293-302.

[17] C. Silla, A. Koerich, C. Kaestner. (2008). *A machine learning approach to automatic music genre classification*. Journal of the brazilian computer sociery, vol. 14, nº 3, pp. 7-18.

[18] S. Li. (septiembre 2000). *Content-based classification and retrieval of audio using the nearest feature line method*. IEEE transactions on speech and audio processing, vol. 8, nº 5, pp. 619-625.

[19] C. Xu, N. Maddage, X. Shao, F. Cao, Q. Tian. (abril 2003). *Musical genre classification using suport vector machines*. IEEE international conference on acoustics, speech and signal processing (ICASSP '03), vol. 5, pp. 429.

[20] G. Li, A. Khokar. (2000). *Content-based indexing and retrieval of audio data using wavelets*. IEEE Xplore conference multimedia and expo, vol. 2, pp. 885-888.

[21] M. Dong. (febrero 2018). *Convolutional neural network achieves human-level accuracy in music genre classification*. ArXiv:1802.09697.

[22] Y. Panagakis, C. Kotropoulos. (2009). *Music genre classification via sparse representations of auditory temporal modulations*. 17th european signal processing conference (EUSIPCO).

[23] A. Paradzinets, H. Harb, L. Chen. (2009). *Multiexpert system for automatic music genre classification*. Research report.

[24] M. Tuceryan, A. Jain. (1993). *Texture analysis*. Handbook of pattern recognition and computer visión, pp. 235-276.

[25] W. Yap, M. Khalid, R. Yusof.(2007). *Face verification with gabor representation and support vector machines*. AMS (American Mathematical Society), pp. 451-459.

[26] Y. Salem, S. Nasri. (2009). *Texture classification of woven fabric based on a GLCM method and using multiclass support vector machines*. 6th international multi-conference on systems, signals and devices (SSD), pp. 1-8.

[27] J. Recio, L. Fernández, A. Fernández-Sarria. (2005). *Use of gabor filters for texture classification of digital images*. Física de la tierra, nº 17, pp. 47-56.

[28] Wikipedia: https://en.wikipedia.org/wiki/Blues. Úlimo acceso: 03/09/2020.

[29] Wikipedia: https://en.wikipedia.org/wiki/Classical_music. Úlimo acceso: 03/09/2020.

[30] Wikipedia: https://en.wikipedia.org/wiki/Country_music. Úlimo acceso: 03/09/2020.

[31] Wikipedia: https://en.wikipedia.org/wiki/Disco. Úlimo acceso: 03/09/2020.

[32] Wikipedia: https://en.wikipedia.org/wiki/Hip_hop_music. Úlimo acceso: 03/09/2020.

[33] Wikipedia: https://en.wikipedia.org/wiki/Jazz. Úlimo acceso: 03/09/2020.

[34] Wikipedia: https://en.wikipedia.org/wiki/Heavy_metal_music. Úlimo acceso: 03/09/2020.

[35] Wikipedia: https://en.wikipedia.org/wiki/Pop_music. Úlimo acceso: 03/09/2020.

[36] Wikipedia: https://en.wikipedia.org/wiki/Reggae. Úlimo acceso: 03/09/2020.

[37] Wikipedia: https://en.wikipedia.org/wiki/Rock_music. Úlimo acceso: 03/09/2020.

[38] Timo Ojala, Matti Pietikäinen, Topi Mäenpää. (2000). *Gray escale and rotation invariant texture classification with local binary patterns.* Lecture notes in computer science, vol. 1842, pp. 404-420.

[39] K echen Song, Yunhui Yan, Yongjie Zhao, Changsheng Liu. (noviembre 2015). *Adjacent evaluation of local binary pattern for texture classification.* Journal of visual communication and image representation, vol. 33, pp. 323-339.

[40] Theodoros Giannakopoulos, Aggelos Pikrakis. (2014). *Introduction to audio analysis: a Matlab approach*. pp. 91.

[41] Wikipedia: https://en.wikipedia.org/wiki/Cross-validation_(statistics) . Úlimo acceso: 21/08/2020.

[42] M. Guijarro, G. Pajares, L. Garmendia. (2010). *Combinación de clasificadores para identificación de texturas en imágenes naturales: nuevas estrategias locacles y globales.*

[43] Mathworks: https://www.mathworks.com/help/stats/choose-a-classifier.html. Último acceso: 11/09/2020.

[44] Enlace:https://www.analyticsvidhya.com/blog/2020/06/4-ways-split-decision-tree/#:~:text=A%20decision%20tree%20makes%20decisions,concept%20that%20everyone%20should%20know.. Último acceso: 11/09/2020.

[45] Isaac Martín de Diego, Javier Moguerza, Alberto Muñoz. (2004). *Combining kernel information for support vector classification.* Multiple classifier systems: 5th international workshop, pp. 102-111.

[46] Mashmoud Parsian. (2015). *Data algorithms.* O'Reilly

[47] Kaggle: https://www.kaggle.com/carlthome/gtzan-genre-collection . Úlimo acceso: 09/09/2020.

[48] Zenhua Guo, L. Zhang, D. Zhang. (enero 2012). *A completed modeling of local binary pattern operator for texture classification*. IEEE Transactions on image processing, vol. 19, pp. 3844-3852.

[49] C Rouzbeh Maani, Sanjay Kalra, Yee-Hong Yang. (2013). *Noise robust rotation invariant features for texture classification*. Pattern Recongnition, vol. 46, nº 8, pp. 2103-2116.

[50] DaulappaGurannaBhalke, Betsy Rajesh, Dattatraya Shankar Bormane. (2017). *Automatic genre*

classification using fractional fourier transform based Mel frequency cepstral coefficient and timbral features. Archives of acoustics, vol. 42, nº 2, pp. 213-222.

[51] G. Paolini, A. Hernández, V. Pereyra. (octubre 2018). F*recuencia fundamental de habla de voz normal según sexo en la provincia de Córdoba, Argentina*. Revista de la facultad de ciencias médicas de Córdoba, pp. 99-100.

[52] Maria De Marsico, Michele Nappi, Hugo Proença. (2017). Human recognition in unconstrained environments. Recuperado de https://www.sciencedirect.com/topics/engineering/local-binary-pattern.

[53] MathWorks: https://es.mathworks.com/help/matlab/ref/audioread.html?s_tid=srchtitle. Úlimo acceso: 17/06/2020.

[54] MathWorks: https://es.mathworks.com/help/symbolic/reshape.html?s_tid=srchtitle. Úlimo acceso: 17/06/2020.

[55] MathWorks: https://es.mathworks.com/help/matlab/ref/repmat.html?s_tid=srchtitle. Úlimo acceso: 17/06/2020.

[56] MathWorks: https://es.mathworks.com/help/signal/ref/spectrogram.html?s_tid=doc_ta. Úlimo acceso: 17/06/2020.

[57] ScienceDirect: https://www.sciencedirect.com/topics/engineering/spectral-centroid. Úlimo acceso: 17/06/2020.

[58] ScienceDirect: https://www.sciencedirect.com/topics/engineering/spectral-flux. Úlimo acceso: 17/06/2020.

[59] ScienceDirect: https://www.sciencedirect.com/science/article/pii/B9780123984999000121. Úlimo acceso: 17/06/2020.

[60] MathWorks: https://es.mathworks.com/help/audio/ug/spectral-descriptors.html#SpectralDescriptorsExample-11. Úlimo acceso: 17/06/2020.

[61] Ch. Sudha Sree, M. V. P. Chandra Sekhara Rao. (2017). *Performance analysis of local binary pattern variants in texture classification.* International journal of signal system control and engineering application, vol. 12, pp. 74-84.

[62] M. Esfahanian, H. Zhuang, N. Erdol. (2013). *Using local binary patterns as features for classification of dolphin calls*. The journal of the acoustical society of america,vol. 134.

GLOSARIO

AELBP: Adjacent Evaluation Local Binary Pattern 4

bpm: beats per minute 6

DCT: Discrete Cosine Transform 6

DFT: Discrete Fourier Transform 5

HR: Harmonic Ratio 6

KNN: K Nearest Neighbours 18

LBP: Local Binary Pattern 3

MFCC: Mel Frequency Cepstral Coefficient 6

SVM: Support Vector Machine 18