

Redes neuronales aplicadas en la detección de acordes de guitarra acústica

Andrés Rondón, Camilo Rueda e Iván Rojas¹

I. RESUMEN

El reconocimiento de acordes ha sido desde el inicio tema de discusión para los artistas en cuanto a las dificultades generadas como la exactitud del acorde y el tiempo excesivo tomado para realizar esta tarea. Este proyecto planea realizar un algoritmo que, por medio de redes neuronales, facilite la detección de acordes de guitarra acústica y los muestre por medio de tablatura para mayor entendimiento del artista.

Palabras Clave—Acordes, redes neuronales, tablatura.

II. INTRODUCCIÓN

La identificación de los acordes de una canción por parte de un artista es muchas veces, un proceso de ensayo y error. El cual consiste en adivinar el acorde según su experiencia y posteriormente, probar si su ensayo fue acertado. El objetivo de este trabajo es implementar una herramienta de software para facilitar el trabajo del artista a la hora de identificar los acordes de una canción. Para ello, el enfoque de solución para este problema se valdrá de la técnica de redes neuronales. Con ella, podremos realizar clasificación de tramos de un archivo de audio a fin de identificar el acorde que se está tocando en dicho tramo.

¿Qué son los acordes? En teoría musical, un acorde es una serie de 3 o más notas tocadas simultáneamente y que constituyen una unidad armónica [1]. Esto quiere decir, que son notas que cuando son tocadas juntas tienen un balance y equilibrio adecuado para que suene *bien* para el oído humano.

La notación usada para representar las notas en este trabajo, será la notación americana. C, D, E, F, G, A, B (Do, Re, Mi, Fa, Sol, La, Si). Ejemplos de esta notación incluyen # (Sostenidos) y b (bemoles). De tal manera que C b significa Do mayor bemol y D# Significa Re mayor sostenido. Además, los acordes se diferencian por su tonalidad que puede ser mayor o menor. Cuando la notación no incluye una 'm', entonces el acorde es mayor, y cuando sí se incluye, el acorde es menor. Por ejemplo, C#m (Do sostenido menor) y C# (Do sostenido mayor). Un acorde se representa como una serie de letras y símbolos que diferencian un acorde de otro. A Continuación, se mostrarán más ejemplos de esta notación.

El uso de las redes neuronales en los problemas de clasificación de datos ha demostrado tener un potencial enorme como se evidencia en los trabajos de [2] y [3]. Por esta razón, los autores de este trabajo tomaron la decisión de utilizar las redes neuronales para un problema de clasificación, esta vez, de audio.

Sin embargo, debe tenerse en cuenta que el alcance de este proyecto está limitado a progresiones de acordes y/o canciones que tengan poco o ningún ruido, es decir, donde el sonido emitido por la guitarra sea el predominante. Esto se hace debido a que el problema de clasificación de instrumentos, voz, distintos tipos de guitarra, es un muy complejo y amerita su propia investigación. Sin embargo, se validará si el software es capaz de clasificar acordes de guitarra eléctrica usando datos de estudio de guitarra acústica. Esto nos servirá para verificar si la red neuronal artificial es capaz de extraer lo más importante en un acorde por ejemplo la nota, la frecuencia, tonalidad, y se basa menos en cómo el sonido es 'realmente'.

III. PLANTEAMIENTO DEL PROBLEMA

Dado el archivo de audio digital correspondiente a una progresión de acordes en guitarra acústica, el problema consiste en dividir el archivo de audio en tramos de acordes y posteriormente identificar cada uno de dichos acordes.

Los archivos de audio para la primera parte del proyecto serán audios cuyo contenido solo sea *una* guitarra acústica. Esta decisión se toma, debido a que el objetivo primal es la identificación de acordes en guitarra acústica, entonces, es importante identificar si el problema tiene una solución viable por medio de redes neuronales y si no se adoptara esta convención, sería muy difícil identificar si en realidad la responsabilidad es del software o si es causado por la presencia de otros sonidos en el audio analizado. Una vez conseguidos resultados satisfactorios bajo estas

¹ Andrés Rondón: amrondonp@unal.edu.co, estudiante de Ingeniería de sistemas y computación, Universidad Nacional de Colombia.

Camilo Rueda: crcruedaal@unal.edu.co estudiante de Ingeniería de sistemas y computación, Universidad Nacional de Colombia.

Ivan Rojas: idrojasf@unal.edu.co estudiante de Ingeniería de sistemas y computación, Universidad Nacional de Colombia.

restricciones, procederemos a hacer pruebas con audios más complejos, aplicando los filtros necesarios.

IV. HIPÓTESIS

1. Las redes neuronales son una herramienta útil y viable para la detección de acordes en audios digitales de guitarra acústica.
2. Un software de detección automática de acordes facilitara el trabajo de los artistas.
3. La detección automática de acordes es una herramienta útil a la hora de clasificar grandes cantidades de archivos de audio.

V. ANTECEDENTES

La guía para este proyecto serán principalmente dos artículos: el primero es “*clasificación de acordes con redes neuronales*” de Russo y Oliva [5] en el cual proponen la clasificación de 24 acordes mayores y menores por medio de una red multicapa hacia adelante (feed-foward) controlando la salida de sus neuronas con una función sigmoidea.

En cuanto a la entrada consiste de un vector de 12 elementos correspondiente al PCP de la muestra que contienen potencia normalizada cada nota y una salida que corresponde a un vector de 24 elementos que indican una medida de proximidad a cada uno de los acordes.

Cabe anotar que “*clasificación de acordes con redes neuronales*” también plantean una solución por medio de mapas auto organizados sin embargo para nuestro propósito sólo tendremos en cuenta la metodología y resultados de la solución por medio de redes neuronales

En segundo lugar el trabajo más robusto encontrado hasta el momento “*Neural networks for musical chords recognition*” Osmalskyj [4], en este artículo, al igual que en el anterior, se realizó un reconocimiento de acordes musicales. Sin embargo en este se tomaran los diez principales (A, Am, Bm, C, D, Dm, E, Em, F, G) y por medio de una red multicapa hacia adelante (feed-foward) se hizo el reconocimiento de dichos acordes. La metodología a seguir fue clara, de un dataset de audios previamente clasificados se tomó el audio y mediante el método llamado PCP (Pitch Class Profile) obtenemos una caracterización del sonido a través de un vector de 12 componentes que corresponden a los valores de los 12 semitonos.

A grandes rasgos el algoritmo computacional generalmente usado en este tipo de casos es primero transformar la entrada sonora en una transformada de Fourier discreta (DFT), acto seguido mediante el algoritmo PCP

caracterizar la señal de frecuencia y generar los posibles grupos de los diez acordes, al final da una probabilidad de suceso que identifica que tan probable es que el acorde de la pista sea el descrito por el algoritmo mostrando una gráfica como la siguiente.

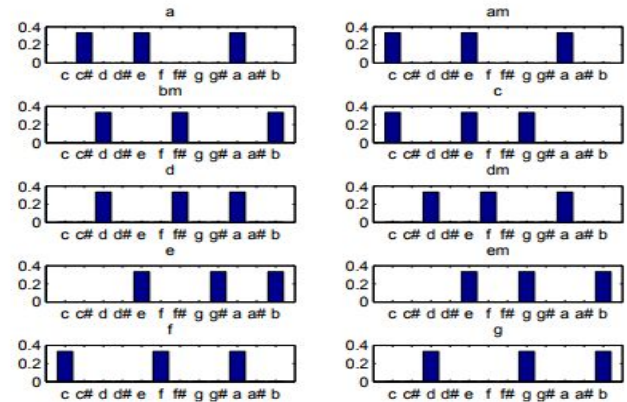


Figura 1 Representación ideal de diez acordes

Tanto para el entrenamiento de la red como para su prueba se tuvieron en cuenta dos tipos de sonido, unos con ruido en la muestra y otras libre de ruido. Ahora en cuanto a resultados el artículo arrojó los siguientes resultados.

TS / LS	Noise-free	Noisy	Mixed
Noise-free	4.0 %	5.0 %	4.0 %
Noisy	11.7 %	6.0 %	7.3 %

Tabla 1 Resultados de validación con muestras libres de ruido, con ruido y mezcladas tanto para el conjunto de entrenamiento (Learning set) como para el conjunto de prueba (Test Set).

En esa tabla podemos ver como el error aumenta considerablemente cuando se analizaron las muestras con ruido, sin embargo, cuando se usan muestras mixtas para entrenar la red podemos ver como disminuye drásticamente el error.

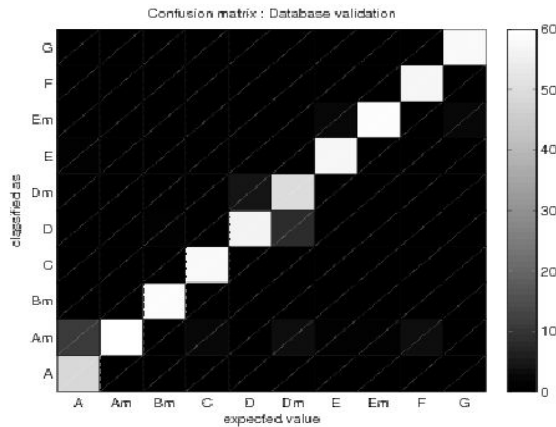


Figura 2 Matriz de confusión para una red entrenada con aprendizaje mixto de 1400 muestras(140 acordes por clase) la prueba contiene 600 muestras (60 por clase) con una tasa de error de 6.5%

Para final mostramos una gráfica que nos permite ver la diferencia entre el valor esperado y los valores arrojados por la red, y podemos ver que son muy parecidos y donde hay errores no son abruptos, son parecidos a los acordes originales.

VI. METODOLOGÍA

En este trabajo, sólo se tomarán en cuenta inicialmente 10 acordes los cuales son.

A, Am, Bm, C, D, Dm, E, Em, F, G

Esto se hace debido a que el dataset solo contiene esos 10 acordes. Sin embargo, a lo largo del trabajo se planea agregar un nuevo acorde (C#) el cual es bastante usado en melodías de música popular.

Para dar solución al problema, se utilizará el algoritmo de Pitch class profile (PCP) que es uno de los algoritmos más usados para la identificación de acordes musicales. En el trabajo de Osmalsky[4]. Se muestra que por sí solo el algoritmo no sirve para hacer clasificación directa.

El algoritmo PCP en su versión básica, toma como entrada un vector representando la señal sonido en el dominio del tiempo y retorna un vector de 12 componentes representando el contenido de frecuencia de la señal en cada una de las notas de la escala musical (también está disponible en 24 y 36 componentes). Aplicando este algoritmo logramos caracterizar archivos de audio en vectores de componente real lo cual permite la aplicación de los conceptos de aprendizaje automático en este

contexto, en particular, las redes neuronales de aprendizaje supervisado.

Posterior a la transformación, comienza la fase de entrenamiento de la red neuronal. La red neuronal propuesta tiene 12 neuronas de entrada que reciben el vector representativo de la muestra de audio, y 10 neuronas de salida que representan la probabilidad de que el audio de entrada sea un acorde de los 10 seleccionados. Se planea usar una capa oculta dado que mostró resultados positivos en los trabajos relacionados. El número ideal de neuronas se determinará realizando la experimentación pertinente.

Los datos de entrenamiento son 2000 audios grabados con guitarra y la validación se hará con 100 audios y con la posibilidad de incluir más instrumentos. En la sección VII se explica más a detalle el dataset usado para entrenamiento y validación.

Hasta este momento tenemos planteado un sistema de identificación de acordes de guitarra en audios con un uno solamente. Para continuar con el objetivo de caracterizar archivos con múltiples acordes, se plantea el un algoritmo para la separación por acordes de una canción.

Definimos entonces un algoritmo de separación de acordes en muestras de audio con muchos acordes seguidos. Este algoritmo se basa en la idea del trabajo de Kurnia Muludi [6] el cual consiste en definir una 'ventana' que es una porción de audio a analizar al tiempo e ir iterando sobre el archivo de audio analizando una ventana a la vez y clasificando el acorde de acuerdo con el algoritmo de clasificación definido.

Algoritmo: Identificación y separación de acordes

Entrada: Señal de audio

Salida: Lista de acordes, con una cantidad de tiempo asociada a cada valor de la lista

```
def split_song( song_signal ):
    window = 0.5 seconds

    partition_signal in partitions of
    . 'window' duration each

    chords = []

    for each portion in partitioned:
        add identify_chord(portion) to
    . chords

    return chords
```

En el algoritmo anterior, la función ‘identify_chord’ hace referencia a un llamado a la red neuronal. En palabras, lo que realiza el anterior algoritmo es dividir el audio en tramos de 0,5 segundos, obtener el acorde de cada tramo aplicando el algoritmo PCP y utilizando la red neuronal. Los problemas a esperarse en este tipo de algoritmos es que en un tramo de medio segundo puede haber más de un acorde, en cuyo caso la cantidad de frecuencia analizada en el PCP sería errónea dando una imposibilidad de clasificación correcta. Sin embargo, tampoco es viable volver muy pequeña la ventana ya que, si se tienen muy pocas muestras de audio, la representación de la frecuencia de la transformada de Fourier podría no ser adecuada y por tanto llegando al mismo problema. En este trabajo optamos por usar la ventana de 0,5 segundos para los experimentos.

XI. DATASETS

Una de las dificultades con las que se inicia al desarrollar este tipo de proyectos es el conjunto de datos con los que se planea realizar el entrenamiento adecuado del sistema. Se hace uso de un dataset propuesto en anteriores proyectos con el fin de ampliar la usabilidad y funcionalidad de estos. En esta base de datos, se guardan archivos de audio (grabados en formato WAV, muestreados en $f_s = 44100$ Hz y cuantificados con 16 bits), y los correspondientes diagramas PCP calculados. Los vectores PCP se calcularon en ventanas que comprenden cada 16384 muestras, que corresponden a 0,37 segundos. Debido a que nuestro objetivo final es el reconocimiento de acordes en un archivo de audio digital, inicialmente se limitará al manejo de los diez acordes más frecuentes descritos en el anterior apartado [4].

Estos acordes están representados por un identificador que varía entre 0 a 9. Se espera que la detección sea solo para un acorde en específico, si otros acordes aparecen en la reproducción, los acordes principales podrán ser insuficientes. Los archivos PCP serán restaurados en un archivo único que está organizado de acuerdo a los doce elementos principales y un elemento más para el identificador del acorde.

La base de datos cuenta con dos subconjuntos, el primero subconjunto contiene una gran cantidad de muestras de los diez acordes más frecuentes de guitarra, mientras que en el segundo subconjunto contiene un conjunto pequeño de acordes en las cuales se mostraran la misma cantidad de acordes frecuentes, pero esta vez tocados en diferentes instrumentos. Esto con el fin de hacer un uso principalmente del primer subconjunto de datos como un conjunto de aprendizaje y entrenamiento, mientras que el segundo, compuesto por diferentes instrumentos, estará

siendo usado para validar si el resultado correspondiente a cada acorde es el indicado según el anterior subconjunto.

Para los acordes del primer subconjunto, se hizo la grabación de la mitad de los acordes en una cámara anecoica con un micrófono de banda ancha (01dB MCE320). La otra mitad se grabó en un ambiente ruidoso, con un solo micrófono en vivo (Shure SM58). Ya que se deduce que el sistema funcionará mejor si está siendo entrenado por una combinación de acordes sin ruido y ruidosos. Cabe aclarar que los acordes también fueron grabados usando varios estilos (arpeggio, staccato, legato, etc.). En la gráfica de la figura 2 podemos observar las representaciones de PCP de un acorde D grabado en la cámara anecoica y en la habitación ruidosa con la misma guitarra.

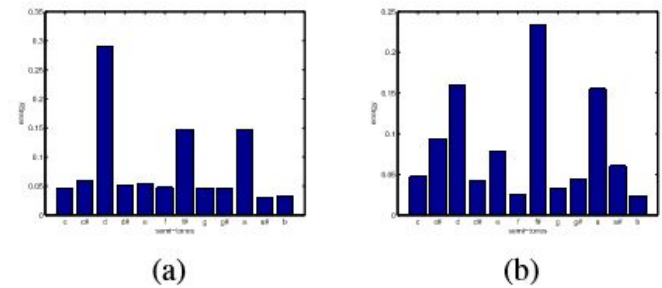


Figura 3: Representaciones PCP de un acorde D grabado con la misma guitarra en (a) cámara anecoica, y (b) una habitación ruidosa. Tenga en cuenta que los tres semitonos principales aún son visibles en (b).

Como muchas canciones reales se reproducen en un ambiente ruidoso, es importante incluir acordes ruidosos en la base de datos.

Para cada acorde, se registraron 100 muestras en la cámara anecoica y se registraron 100 muestras en una habitación ruidosa. Para cada ambiente, las muestras se grabaron con cuatro guitarras diferentes: una guitarra clásica con cuerdas de nylon y tres guitarras acústicas que producen tres sonidos diferentes. Se espera que la variedad del conjunto de datos con respecto a las guitarras mejorará la robustez del sistema y ampliará su aplicabilidad, ya que hay muchos sonidos de guitarra diferentes disponibles en todo el mundo [4].

Para el segundo subconjunto se creó una base de datos más pequeña que contiene acordes grabados en una guitarra y otros tres instrumentos, piano, violín y acordeón. Esa base de datos está destinada a proporcionar un conjunto de pruebas por lo cual no se usará para el entrenamiento del modelo. Contiene 100 acordes para cada instrumento. Esos

100 acordes se distribuyen por igual entre los diez acordes mencionados anteriormente, por lo tanto, hay 10 muestras por acorde respectivo a cada instrumento.

Dataset de audios con más de un acorde

Para validar los resultados del trabajo, también se usarán pistas de audio con múltiples acordes. El set de datos está dividido en los siguientes grupos.

1. Guitarra acústica solamente
2. Versión karaoke (todos los instrumentos, pero sin la voz del cantante)
3. Canción original

Desafortunadamente, armar una base de datos con las anteriores características no es trabajo fácil ya que se requiere de programas de edición de audio y de bastante tiempo para su construcción. Es por esto, que las pruebas se realizarán solamente en 3 canciones distintas, lo que da un total de 9 pistas de prueba.

VIII. RESULTADOS

La red neuronal se implementó en el lenguaje de programación Python 3 utilizando la librería de aprendizaje automático Keras, el código fuente se encuentra disponible en GitHub².

Se corrieron experimentos con distintos tipos de redes neuronales, variando la cantidad de capas ocultas, la cantidad de neuronas, las funciones de activación, el número de muestras vistas a la vez. Entre todas las redes neuronales probadas, la que obtuvo mejores resultados se describe a continuación.

Número de neuronas en la capa de entrada	12
Capas ocultas	1
Número de neuronas en la capa oculta 1	30
Número de neuronas en la capa de salida	10

Tabla 2: Características de la red neuronal con mejores resultados

También se realizaron experimentos sobre el optimizador y las funciones de activación. Sobre la red neuronal especificada de la tabla anterior, se experimentó con diferentes funciones de pérdida (loss functions) y los resultados fueron los siguientes.

Función de pérdida	Exactitud categórica en el conjunto de entrenamiento
binary crossentropy	98.70%
cosine proximity	98.65%
mean squared error	98.50%
mean squared logarithmic error	98.35%
categorical crossentropy	98.30%
logcosh	98.25%
poisson	98.20%
categorical hinge	98.05%
hinge	27.25%
mean absolute percentage error	14.40%
kullback leibler divergence	12.30%
squared hinge	11.65%
mean absolute error	5.20%

Tabla 3: Resultados de exactitud de la red neuronal utilizando distintas funciones a optimizar

La **exactitud categórica** determina si la predicción tiene el máximo valor en el mismo índice que el del máximo valor del resultado esperado. Lo cual quiere decir que, si tomamos el valor máximo de la salida como la predicción, esta sería correcta en el porcentaje que se muestra en la tabla.

Tomando entonces la red neuronal que mayor tuvo exactitud en el experimento pasado, se prueba la red neuronal con la función de exactitud categórica y los siguientes fueron los resultados.

Instrumento	Exactitud categórica
Guitarra	95.00%
Piano	91.00%
Violín	82.00%

Tabla 4: Resultados de exactitud de la red neuronal en el conjunto de datos de validación

Los resultados muestran que, en el set de 100 muestras de validación para cada instrumento, hubo una exactitud del 95% para guitarra, 91% para piano y 82% para violín.

² <https://github.com/amrondonp/chord-recognition>

Separación en archivos de audio con más de un acorde.

Dado que se ha alcanzado una exactitud del 95% en la guitarra vale la pena analizar una canción con múltiples acordes e intentar validar y medir la exactitud bajo ese caso.

Recordemos que el set de datos tiene 3 representaciones para una porción de la misma canción: la versión original, versión en karaoke y versión con solo guitarra.

Para la medición de la exactitud de nuestro algoritmo, se utilizaron dos métricas, la primera la llamamos “similitud absoluta” y la segunda “similitud por LCS”

Métrica de similitud absoluta

El resultado del algoritmo de separación es comparado directamente acorde por acorde con el resultado esperado, es decir, se cuenta el número de acordes en el cual las dos secuencias coinciden.

Métricas de similitud por LCS

Se utiliza el algoritmo de ‘Longest Common Subsequence’ para obtener la longitud de la subsecuencia común más larga.

Ambas métricas se normalizan en la longitud de las secuencias para obtener un número de 0 a 1 que indica el porcentaje de similitud por cada métrica.

Los resultados del experimento de separación se muestran en la siguiente tabla.

	Cancion	% similitud absoluta	% similitud LCS
Original	about_a_girl	3.70%	3.70%
	every_breath_you_take	0.00%	14.29%
	wind_of_change	16.67%	39.58%
Guitarra	about_a_girl	70.37%	85.19%
	every_breath_you_take	75.00%	75.00%
	wind_of_change	70.83%	95.83%
Karaoke	about_a_girl	0.00%	0.00%
	every_breath_you_take	0.00%	14.29%
	wind_of_change	47.62%	69.05%

Tabla 5: Resultados de exactitud de el algoritmo de separación por acordes con la métrica absoluta y LCS

IX. DISCUSIÓN DE LOS RESULTADOS

Los resultados que se obtuvieron para los acordes de guitarra tienen una exactitud bastante aceptable (95%), a

pesar de que, en su sonido y forma, algunos audios eran considerablemente distintos de los audios de entrenamiento. Esto nos muestra que este enfoque de clasificación por red neuronal provee una buena generalización sobre los acordes musicales.

Además, es importante notar que se presentó una tasa relativamente alta para los dos instrumentos restantes. Esto de nuevo refuerza la afirmación de que la red neuronal es una buena herramienta de generalización, ya que ningún audio de piano o violín fue utilizado para el entrenamiento.

En el ámbito de la separación de acordes en canciones, se obtuvo una precisión absoluta para las 3 canciones por encima del 70% en los audios en donde solo está presente la guitarra. Considerando los problemas de separación de acordes, este resultado puede considerarse aceptable como primera aproximación. Sin embargo, podemos observar que la exactitud decrece bastante cuando se consideran audios con más instrumentos (karaoke) y audios originales con instrumentos y voz.

De los anteriores resultados podemos inferir que la generalización del tono musical y acorde se alcanzó hasta cierta medida, pues las clasificaciones en los archivos con acorde sencillo y compuestos con instrumentos únicos arrojaron una tasa de exactitud alta. Sin embargo, al introducir varios instrumentos simultáneamente en un mismo archivo de audio no se obtuvieron resultados similares.

X. CONCLUSIONES

Se implementó un método de detección de 10 acordes en guitarra acústica basado en redes neuronales que alcanzó una precisión del 95% con resultados favorables. También se lograron buenos resultados para los instrumentos musicales piano (91%) y violín (82%).

Se implementó un algoritmo de identificación y separación de acordes que tiene una exactitud del 70% en muestras donde solo la guitarra está presente. Para los casos en los que había más instrumentos y/o voces los resultados no fueron buenos pues en promedio la exactitud estuvo por debajo del 10%. Una manera en que los resultados podrían mejorar es creando un conjunto de datos que contenga acordes, pero con otros instrumentos y voces.

Las redes neuronales son una herramienta viable en los procesos de reconocimiento de audio musical pues con un set de datos relativamente pequeño se obtuvieron resultados importantes, es decir, se obtuvo una alta exactitud en el set de entrenamiento, en el de validación y en el de acordes múltiples. Los esfuerzos para mejorar las

aplicaciones en este campo deben ir centrados a crear conjuntos de datos nuevos, mejores y más variados con otros instrumentos y voces que probablemente ayudarán a que la exactitud sea aún mayor y a hacer viable su aplicación por parte de los profesionales de la música.

XI. TRABAJO FUTURO

El principal trabajo a realizar a futuro es aumentar el set de datos para lograr una mejor generalización en los audios con múltiples instrumentos y voces. Se espera con este trabajo poder aumentar la exactitud en los sets de datos de canciones a karaoke y con voz.

Otra tarea importante es soportar más acordes de los 10 de este informe. Para ello se necesita crear un set de datos de 200 muestras para cada acorde que se desea agregar, pero el impacto que esto tendría es que el número de canciones que potencialmente podrían ser clasificadas crecería exponencialmente.

Otro trabajo que puede mejorar la exactitud del algoritmo es proveer la opción de usar ventanas de tamaño variable. Sin embargo, para ello primero se debe definir un formato de especificación de la salida esperada de tal manera que se especifique el acorde y el tiempo en milisegundos que se espera obtener dicho acorde. Una implicación de esto es que la tarea de crear un set de validación para este algoritmo se hace más compleja.

Aumentar el set de datos de validación de las canciones con acordes múltiples a por lo menos 50 canciones.

Obtener retroalimentación y consejos de expertos musicales en el ámbito de caracterización, clasificación y segmentación de pistas de audio musicales.

XII. ANEXOS

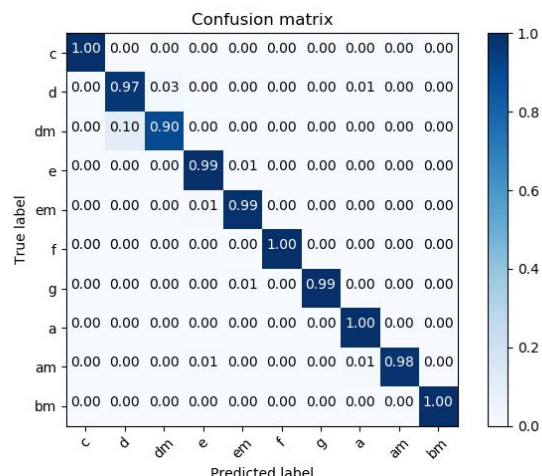


Figura 4: Matriz de confusión para el set de datos de entrenamiento.

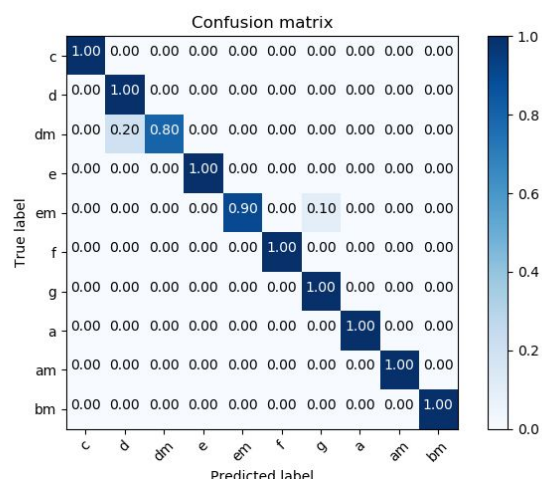


Figura 5: Matriz de confusión para guitarra del set de datos de validación

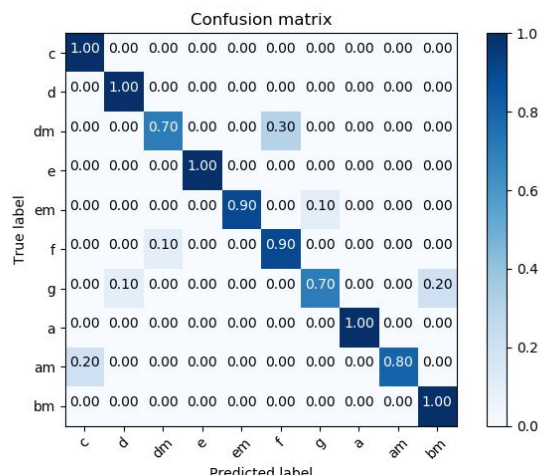


Figura 6: Matriz de confusión para piano en el set de datos de validación

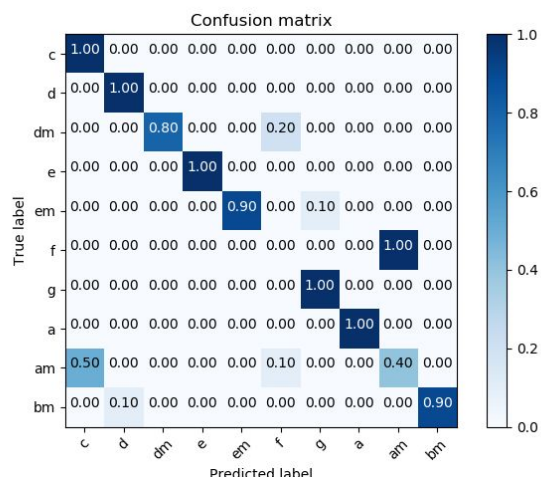


Figura 7: Matriz de confusión para violín en el set de datos de validación

XIII. REFERENCIAS

1. Joaquín Zamacois: Tratado de armonía (I). 2006. ISBN 84-8236-232-1. Pág. 28, ap. 28a.
2. Artificial Neural Network based Web Application Firewall for SQL Injection, International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:4, No:4, 2010.
3. Artificial Neural Networks that Classify Musical Chords, Int'l Journal of Cognitive Informatics and Natural Intelligence, 2(3), 22-30, July-September 2008
4. Neural networks for musical chords recognition, J. Osmalskyj, J.-J. Embrechts, S. Piérard, M. Van Droogenbroeck, Actes des Journées d'Informatique Musicale (JIM 2012), Mons, Belgique, 9-11 mai 2012.
5. Clasificación de acordes con redes neuronales, Rodrigo Russo y Andrés Oliva Dra. Ing. Lucia Isabel Passoni, CC Ana Lucia Dai Pra, Dr. Ing. Gustavo Meschino.
6. Chord Identification Chord IdentificationUsing Pitch Class Profile Method with Using Pitch Class Profile Method with Fast Fourier

Transform Feature Extraction, Kurnia Muludi, Aristoteles, Abe
Frank SFB Loupatty, 2014