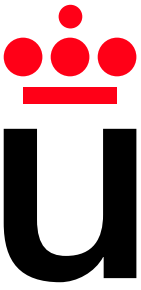


Universidad
Rey Juan Carlos

Escuela Técnica Superior
de Ingeniería Informática

Grado en Matemáticas

Curso 2023-2024

Trabajo Fin de Grado

REDES NEURONALES PARA LA CLASIFICACIÓN DE INSTRUMENTOS MUSICALES A PARTIR DEL ANÁLISIS DE ESPECTROGRAMAS

Autora: Irene Casas López
Tutores: Clara Simón De Blas[1] y Emanuele Schiavi[2].

[1]Departamento de Informática y Estadística.
[2]Departamento de Matemática Aplicada, Ciencia e Ingeniería de los Materiales y Tecnología Electrónica.

"¿No debería describirse la música como las matemáticas del sentimiento y las matemáticas como la música de la razón?"

James Joseph Sylvester, *On Newton's Rule for the Discovery of Imaginary Roots; Collected Mathematical Papers, Vol. 2*, p. 419.

Agradecimientos

Ha llegado finalmente el día en que concluyo esta etapa universitaria de mi vida. Por tanto, no me gustaría dejar pasar la oportunidad de mencionar a todas aquellas personas que han estado tan cerca de mí durante estos años y que me han infundido el valor, la energía y el cariño necesarios para afrontar cada uno de los retos. Gracias a vosotros, las Matemáticas se han convertido en la profesión que me acompañará a partir de ahora fuera de las aulas... o tal vez continúe dentro.

Antes que nada, quiero expresar mi más profundo agradecimiento a mis padres, Juani y Fernando. Por sus sabios consejos y sacrificios, por su fuente de amor inagotable y por ayudarme a no tirar la toalla aún cuando las cosas se volvían difíciles. Todo lo que soy, se lo debo a ellos.

Recordando con nostalgia a mis profesores de infancia y adolescencia, en especial a Don Ángel, Justo, Julio, Gerardo y El.Lodia, sin olvidar tampoco a Elisa, mi profesora de piano. Gracias a cada uno de ellos por su dedicación y por transmitirme su pasión por las matemáticas, la física y la música, despertando en mí la curiosidad de querer profundizar más en estas materias.

A lo largo de mi estancia en la URJC, me gustaría dar las gracias a mis tutores de este TFG, Clara y Emanuele, por haber estado pendientes y guiarme durante estos meses de trabajo. También quiero destacar a Miguel y Ángel, otros profesores del Grado de Matemáticas, de los que he adquirido grandes conocimientos.

Y por supuesto, esto no habría sido posible sin la ayuda de Sergi, mi otro yo, mi confidente, a quien le estoy eternamente agradecida. Por su inmensa paciencia, por creer siempre en mí y por transmitirme paz en todo momento. Gracias por su colaboración en la grabación de sonidos musicales y por compartir conmigo esta experiencia.

También me gustaría mencionar a mi abuela Consuelo, desaparecida en plena pandemia. Probablemente su alegría al verme finalizando este proyecto académico e ir progresando poco a poco habría sido comparable a sus ganas de vivir.

Gracias a mis amigas de Sigüenza de toda la vida, por su cercanía y apoyo. Gracias también a mis amigos de Madrid, de la parroquia de San Juan Crisóstomo y compañeras de residencia. Por acompañarme y permitirme algún rato de

desconexión de la gran cantidad de horas que he dedicado a este trabajo.

Por último, no quiero terminar sin agradecer a Santa Cecilia, la patrona de la música, a San Buenaventura y a San Huberto, patronos de las matemáticas, y a Santo Tomás de Aquino, patrono de los estudiantes. Gracias por interceder por mí e iluminar mi camino a seguir. Gracias a María y al Señor desde las alturas, por haber sido mi guía y fuente de inspiración y fortaleza en mi vida académica y espiritual, ayudándome a afrontar cada uno de los retos con confianza y esperanza.

Muchas gracias a todos y cada uno de ellos. Me llevo la satisfacción del deber cumplido, de la enseñanza de que cada esfuerzo tiene su recompensa y de que la madurez personal debe ir asociada siempre al trabajo, esfuerzo y dedicación.

Resumen

Este proyecto tiene como propósito ofrecer un modelo de predicción para identificar diferentes instrumentos musicales mediante un conjunto de espectrogramas generados a partir de la grabación de 136 acordes musicales con un sintetizador.

En primer lugar, se presenta una breve introducción sobre la evolución del estudio de los armónicos de un sonido, desde las contribuciones iniciales de Pitágoras hasta los posteriores descubrimientos de Fourier. Acto seguido, se utiliza el método de La Transformada de Fourier de Tiempo Reducido (STFT) para descomponer las señales de audio en espectrogramas. Estas imágenes contendrán información de la intensidad de las frecuencias a lo largo del tiempo y formarán la base de datos para el estudio.

Por último, se analiza la clasificación de los diversos espectrogramas generados para determinar a qué instrumento pertenecen, empleando la arquitectura de GoogLeNet, una red neuronal convolucional de aprendizaje supervisado. Se describirán también los diferentes experimentos realizados, se expondrán las conclusiones obtenidas y se mencionarán las líneas de investigación futuras.

Palabras clave:

- Espectrograma
- Armónicos
- Transformada de Fourier
- GoogLeNet
- Instrumentos Musicales
- Red Neuronal Convolucional

Abstract

This project aims to provide a prediction model to identify different musical instruments by using a set of spectrograms generated from the recording of 136 musical chords with a synthesizer.

Firstly, a brief introduction is given on the evolution of the study of sound harmonics, from the initial contributions of Pythagoras to the later discoveries by Fourier. Next, the Short-Time Fourier Transform (STFT) method is employed to decompose audio signals into spectrograms. These images will contain information about the intensity of the frequencies over time and will form the database for the study.

Finally, the classification of the various generated spectrograms is analyzed to determine which instrument they belong to, using the architecture of GoogLeNet, a supervised learning convolutional neural network. The different experiments carried out will also be described, the conclusions obtained will be presented, and future lines of research will be mentioned.

Palabras clave:

- Spectrogram
- Harmonics
- Fourier Transform
- GoogLeNet
- Musical Instruments
- Convolutional Neural Network

Índice de contenidos

Índice de figuras

Índice de códigos

1

Desarrollo científico de la Teoría Musical y de los Armónicos

1.1. Pitágoras y las primeras aportaciones.

Los primeros descubrimientos en el campo de la Teoría Musical y las Proporciones Armónicas fueron realizados por Pitágoras de Samos (500 a.C.), que experimentó con las vibraciones sonoras producidas a partir de utensilios cotidianos y construyó un instrumento musical de una sola cuerda tensada, llamado monocordio. Así, fue comparando los sonidos de dos en dos a partir de distintas variaciones en la longitud de la cuerda.

De esta forma, llegó a la conclusión de que los sonidos generados al hacer vibrar una cuerda en toda su extensión y al presionar posteriormente a la mitad de su longitud congeniaban de forma armónica. Tiempo después se averiguará que esta proporción 1:2 equivale a lo que hoy en día conocemos como un intervalo de octava. Además, observó que al dividir la cuerda a la tercera parte de su longitud y después a la cuarta parte, se obtenían los denominados intervalos de quinta y cuarta, respectivamente, asociados a las proporciones 2:3 y 3:4.

A continuación, se muestran de forma gráfica las anteriores consonancias perfectas que ayudarán con el tiempo a la elaboración de una escala musical ([21], [2]).

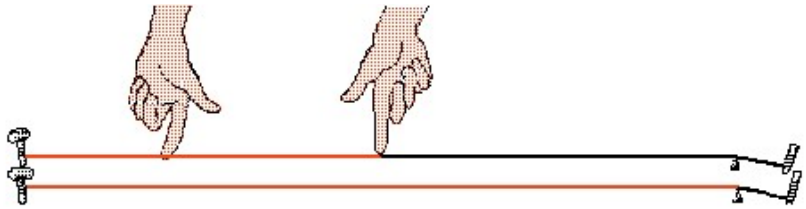


Figura 1.1: Intervalo de octava (1:2) (Imagen extraída de [1])

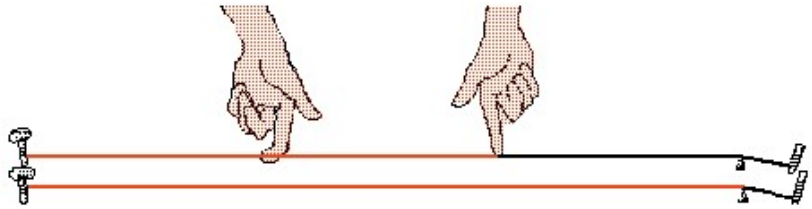


Figura 1.2: Intervalo de quinta (2:3) (Imagen extraída de [1])

Figura 1.3: Intervalo de cuarta (3:4) (Imagen extraída de [1])

Cuando la longitud de la cuerda era más grande, se apreciaba un sonido con un tono más grave. En cambio, al disminuir su tamaño, el tono percibido era más agudo [22].

Se estima que, a partir del año 1600 aproximadamente, se sustituirá la notación de las proporciones en la longitud de cuerda por relaciones de **frecuencia** a la hora de caracterizar un intervalo musical [21].

En el siguiente apartado se explicará con más detalle.

1.2. Las contribuciones de Sauveur, la serie armónica y las cualidades del sonido.

Joseph Sauveur (1653-1716) fue un científico francés que se dedicó en mayor parte a las Matemáticas, a la Anatomía y a la Botánica, y a pesar de su discapacidad auditiva y del habla, realizó grandes contribuciones en el ámbito de la

Música. "No tenía ni voz ni oído, pero no pensaba más que en la música. Se vio reducido a tomar prestada la voz y el oído de los demás y a cambio les hacía demostraciones hasta entonces desconocidas a los músicos"[23] fue una descripción evocadora realizada por el escritor y filósofo Bernard le Bovier de Fontenelle [21].

En su línea de investigación, Sauveur hizo un hallazgo significativo situando pequeñas tiras de papel a lo largo de una cuerda, cuya vibración generaba movimientos independientes de ascenso y descenso en secciones diferentes de la misma. Por tanto, pudo señalar que el comportamiento de dichas vibraciones se producía de manera muy desordenada y particular [21].

Definición 1.2.1. *Al escuchar una nota producida por un instrumento musical, el oído no la percibe como un tono puro, sino que también se mezcla con una serie infinita de notas más agudas. Esta combinación, conocida como* ***timbre*** *o* ***color****, contribuye a la caracterización única del sonido y permite reconocer qué instrumento se está tocando a partir del sonido emitido ([21], [2]).*

Por ejemplo, un mismo tono interpretado en un trombón y en un piano genera sensaciones auditivas totalmente diferentes, tal y como aparece ilustrado en la Figura 6.2.

Definición 1.2.2. *La secuencia de notas o tonos armónicos descrita anteriormente da lugar a la siguiente serie divergente, también llamada* ***serie armónica*** *[21]:*

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + ... = \sum_{n=1}^{\infty} \frac{1}{n} \tag{1.1}$$

Definición 1.2.3. *La* ***altura tonal*** *de un sonido se encuentra determinada por la frecuencia, que es el número de oscilaciones producidas en cada segundo, permitiendo discernir entre sonidos graves o agudos. La frecuencia es inversamente proporcional a la longitud de la cuerda en instrumentos musicales de cuerda pulsada o frotada y su unidad en el Sistema Internacional se mide en Herzios (Hz).*

El rango de frecuencias audible se encuentra comprendido entre 20 Hz y 20000 Hz, aproximadamente. Este intervalo equivale a unas diez octavas, aunque puede variar en función de la persona y tiende a disminuir con la edad. A diferencia de la vista, que presenta dificultades para distinguir los colores que componen una mezcla, el oído tiene la habilidad de reconocer las diferentes frecuencias combinadas en un mismo sonido [21].

Definición 1.2.4. *La* ***intensidad*** *se refiere a la cantidad de energía presente en una onda sonora, la cual se encuentra influenciada por su amplitud, la sensibilidad auditiva del individuo y la distancia desde la cual se emite el sonido. Esta medida permite diferenciar entre sonidos fuertes y débiles y se expresa comúnmente en decibeles (dB) mediante un sonómetro ([24], [25]).*

La amplitud representa la magnitud del desplazamiento máximo de las partículas de aire en una onda sonora con respecto a su posición de reposo en cada ciclo de vibración; a mayor amplitud de onda, más fuerte será la intensidad del sonido percibido. Sin embargo, la percepción humana de la intensidad del sonido muestra una relación logarítmica en lugar de una relación lineal, debido a que un sonido con el doble de intensidad no se percibe necesariamente como el doble de fuerte por el oído. Los sonidos audibles deben estar por encima del nivel de sonido mínimo perceptible (0 dB) pero no sobrepasar el umbral de dolor (140 dB), ya que podrían causar daños auditivos severos ([24], [25], [26]).

Definición 1.2.5. *La **duración** de un sonido se define como el tiempo de un cuerpo en emitir un movimiento vibratorio, siendo posible distinguir entre sonidos breves o más prolongados [24].*

En instrumentos musicales como el órgano o el violín, se puede mantener la duración del sonido de forma prolongada. Sin embargo, en la mayoría de los instrumentos de viento, la duración depende de la capacidad pulmonar del músico. En el caso del piano, su duración también es limitada, pero cuenta con un pedal de resonancia que permite extender las notas por más tiempo de lo común.

Después de esta breve introducción de las cualidades básicas del sonido, se va a proseguir con la explicación de formación de los armónicos.

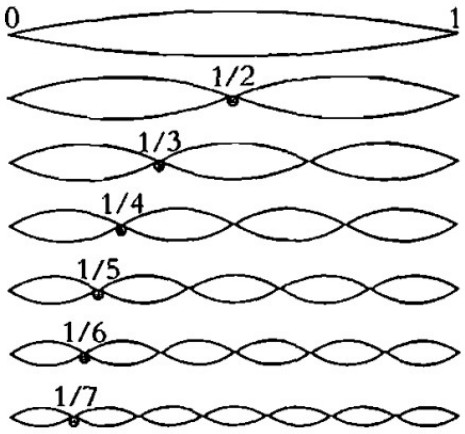


Figura 1.4: Secuencia de armónicos en una cuerda vibrante [2])

Al hacer resonar en toda su longitud una cuerda tensada, sujeta en sus extremos a dos puntos fijos, se generan *ondas estacionarias* (ver Definición 1.3.5) El sonido producido es conocido como **nota fundamental** o **primer armónico**, asociado al primer término de la serie 1.1 y, a su vez, es el que va a indicar el tono principal.

De modo simultáneo, surgen otras vibraciones independientes en distintas fracciones de la cuerda. Cuando la onda se divide en mitades idénticas, aparece

el **segundo armónico**, y en el centro de la cuerda se forma un punto estático o *nodo* (ver Definición 1.3.6). Al dividirse en tres partes iguales a partir de dos nodos intermedios, se manifiesta el **tercer armónico**. Si se forman cuatro ondas separadas por tres nodos a partir de la onda inicial, nos encontramos ante el **cuarto armónico**. Estos corresponden a los términos segundo, tercero y cuarto de la serie armónica, respectivamente. De esta forma, se obtiene el resto de la lista infinita de armónicos sucesivos ([27], [28], [29], [21]). Para una mejor comprensión, se puede observar visualmente en la Figura 1.4 cómo cada armónico representa un *modo* específico de oscilación de la cuerda (ver Observación 1.3.1).

A partir de estos experimentos, Sauveur pudo concluir también que las frecuencias de aquellas vibraciones correspondían a múltiplos enteros de la frecuencia más grave producida por la nota fundamental. Es decir, la frecuencia producida por el segundo armónico es el doble que la de la fundamental, la del tercer armónico es el triple y así sucesivamente [21].

A medida que la secuencia avanza, la amplitud e intensidad de los armónicos procedentes de la nota fundamental son cada vez menores, volviéndose más tenues e imperceptibles por el oído humano. Además, estos sonidos se mantuvieron escondidos hasta que Sauveur pudo demostrar su existencia casi dos mil años después [21].

En la Figura 1.5 aparecen representados los primeros 16 armónicos de la nota fundamental Do grave (Do_1), o primer armónico, ubicada en la primera octava de un piano. Se puede apreciar que cuatro de ellos se encuentran representados con un relleno negro; esto es debido a que sus afinaciones no coinciden exactamente con las notas descritas en el pentagrama, pero se aproximan bastante [30].

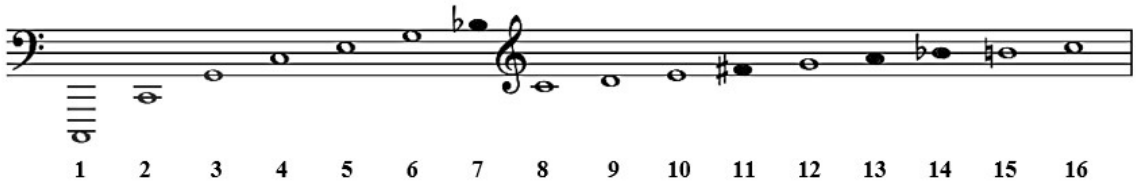


Figura 1.5: Los 16 primeros armónicos de Do_1 representados en un pentagrama (Imagen de creación propia inspirada en [2])

Las relaciones de frecuencia y longitud de cuerda para los primeros armónicos de Do_1, con una frecuencia aproximada de $f = 32{,}7$ Hz [31], se presentan en la Tabla 1.1. Aquí, L denota la longitud de la cuerda cuando se toca la nota fundamental. Las proporciones indican las subdivisiones sucesivas de la cuerda donde se forman los nodos que darán lugar a los armónicos, siguiendo la serie armónica (ver Ecuación 1.1).

La última columna contiene información sobre el intervalo existente entre el tono fundamental y su correspondiente armónico.

Tabla 1.1: Relación entre las frecuencias y longitudes de cuerda de los 16 primeros armónicos de Do_1

Nº armónico	**Nota**	**Frecuencia**	**Proporción de la cuerda**	**Intervalo**
1	Do_1	$f = 32{,}7$ Hz	L	Tono fundamental
2	Do_2	$2f = 65{,}4$ Hz	$\frac{L}{2}$	Octava justa
3	Sol_2	$3f = 98{,}1$ Hz	$\frac{L}{3}$	Quinta justa
4	Do_3	$4f = 130{,}8$ Hz	$\frac{L}{4}$	Octava justa
5	Mi_3	$5f = 163{,}5$ Hz	$\frac{L}{5}$	Tercera mayor
6	Sol_3	$6f = 196{,}2$ Hz	$\frac{L}{6}$	Quinta justa
7	$Si\flat_3$	$7f = 228{,}9$ Hz	$\frac{L}{7}$	Séptima menor
8	Do_4	$8f = 261{,}6$ Hz	$\frac{L}{8}$	Octava justa
9	Re_4	$9f = 294{,}3$ Hz	$\frac{L}{9}$	Segunda mayor
10	Mi_4	$10f = 327{,}0$ Hz	$\frac{L}{10}$	Tercera mayor
11	$Fa\sharp_4$	$11f = 359{,}7$ Hz	$\frac{L}{11}$	Cuarta aumentada
12	Sol_4	$12f = 392{,}4$ Hz	$\frac{L}{12}$	Quinta justa
13	La_4	$13f = 425{,}1$ Hz	$\frac{L}{13}$	Sexta mayor
14	$Si\flat_4$	$14f = 457{,}8$ Hz	$\frac{L}{14}$	Séptima menor
15	$Si\natural_4$	$15f = 490{,}5$ Hz	$\frac{L}{15}$	Séptima mayor
16	Do_5	$16f = 523{,}2$ Hz	$\frac{L}{16}$	Octava justa

1.3. Superposición de ondas

1.3.1. Movimiento ondulatorio armónico y clasificación de ondas

Para lograr una mejor comprensión del concepto de superposición de ondas, se van a explicar unos conceptos previos en este apartado y se mencionarán los diferentes tipos de ondas que pueden aparecer.

Definición 1.3.1. *Un **movimiento periódico** es aquel que se repite en intervalos de tiempo constantes. Es decir, para una función $f(x)$ dada, se cumple que $f(x) = f(x+T)$, donde x es cualquier valor del dominio y T representa el periodo. El periodo es el tiempo mínimo que tarda la onda en completar un ciclo y regresar al mismo estado de perturbación ([21], [32]).*

Definición 1.3.2. *Un **movimiento ondulatorio** es el proceso mediante el cual se genera una perturbación que se desplaza a través de un medio, transmitiendo únicamente energía y cantidad de movimiento, sin transferir materia.*

Dependiendo de la naturaleza y del medio en el que se propagan, las ondas pueden ser *mecánicas*, si requieren de un medio material para su propagación, como el sonido producido por una cuerda vibrante; y *electromagnéticas*, pudiendo generarse en el vacío a partir de un campo eléctrico o magnético.

Además, las ondas también se clasifican según la dirección en la que vibran las partículas que la componen. Una *onda transversal* es aquella en la que las partículas vibran perpendicularmente a la dirección de propagación de la onda. Un ejemplo es la vibración de una cuerda tensada, donde las partículas se mueven hacia arriba y hacia abajo mientras la onda se propaga de manera horizontal a lo largo de la cuerda. Por otro lado, en una *onda longitudinal*, la vibración de las partículas es paralela a la dirección en que la onda se propaga. Un ejemplo común son las ondas sonoras ([32], [33]).

Definición 1.3.3. *Se dice que un movimiento ondulatorio periódico es también **movimiento armónico** si las partículas que lo componen describen una forma de onda sinusoidal que oscilan en torno a un punto de equilibrio central.*

De este modo, la onda armónica se expresa en función de cada punto x del espacio y en cualquier instante t de tiempo de la siguiente manera:

$$f(x,t) = A\sin(kx - \omega t + \phi) \tag{1.2}$$

donde además:

- A es la amplitud máxima de la onda desde su posición de equilibrio.
- ϕ es la fase inicial cuando $t = 0$ y $x = 0$.
- k es el número de onda y es inversamente proporcional a la longitud de onda, denotada como λ, mediante la relación: $k = \frac{2\pi}{\lambda}$.
- ω es la frecuencia angular y se relaciona con el periodo T a partir de la expresión: $\omega = \frac{2\pi}{T}$; y como el periodo es la inversa de la frecuencia f, el número de ciclos por segundo, también se tiene: $\omega = 2\pi f$ [34].

Nota 1.3.1. *Las señales coseno son un tipo de señales sinusoidales, pero presentan una diferencia de fase de $\frac{\pi}{2}$ con respecto a las señales seno [35].*

1.3.2. Ecuación de Onda y Principio de Superposición. Bernoulli y D'Alembert

Daniel Bernoulli (1700-1782) provenía de una familia de matemáticos en la que se respiraba un ambiente tenso y lleno de rivalidad y conflictos. En uno de sus

notables descubrimientos, logró identificar ciertos nodos en el movimiento vibratorio de una cuerda y determinar sus frecuencias de oscilación. Como resultado, concluyó que las vibraciones armónicas de cada modo se encuentran presentes al mismo tiempo y de manera independiente, conformando así el movimiento fundamental de la cuerda. Este hallazgo lo documentó en un artículo titulado "Reflexions et éclaircissements sur les nouvelles vibrations des cordes"(1747-1748), lo que le llevó a reconocer el Principio de Superposición ([36], [21], [22], [37]).

Definición 1.3.4. *El **Principio de Superposición** establece que si varias ondas coinciden en un mismo punto del espacio, el desplazamiento resultante es equivalente a la suma de los desplazamientos de cada onda individualmente, generándose así una nueva onda con una forma más compleja. En otras palabras, sean tres funciones $a(x,t)$, $b(x,t)$ y $c(x,t)$ donde cada una describe el movimiento de una onda diferente, entonces la superposición de ellas se representa mediante otra fución, denotada como $r(x,t)$, de la siguiente manera [34]:*

$$r(x,t) = a(x,t) + b(x,t) + c(x,t)$$

En sus experimentos de cuerda vibrante, Bernoulli situó una serie finita de masas conectadas mediante la fuerza de tensión a los dos puntos adyacentes de la misma. Esto condujo a un sistema complejo de n ecuaciones diferenciales ordinarias, una por cada masa.

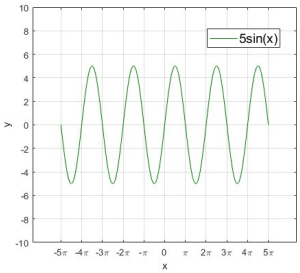
Más adelante, D'Alembert logró simplificar el problema como una única ecuación diferencial. Jean D'Alembert (1717-1783) pudo tener acceso a una excelente educación desde una edad temprana y a pesar de las múltiples materias que exploró las matemáticas fueron su mayor pasión, donde prácticamente fue autodidacta. Fue un precursor en la investigación de las ecuaciones diferenciales y su aplicación en el ámbito de la física. Por ello, transformó ese modelo de sistema discreto que propuso Bernoulli en uno continuo, donde la distancia entre las masas se reduce a cero. Aplicó también la Segunda Ley de Newton ($F = m \cdot a$) a la aceleración de estos puntos con movimiento vertical y a la tasa de variación de la inclinación de la cuerda entre dos puntos contiguos. De esta manera, D'Alembert fue el primero en plantear la denominada **Ecuación de Onda unidimensional** (ver Ecuación 1.3), la cual describe todo movimiento ondulatorio. Además, logró proporcionar una solución para dicha ecuación.

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{v^2} \cdot \frac{\partial^2 y}{\partial t^2} \tag{1.3}$$

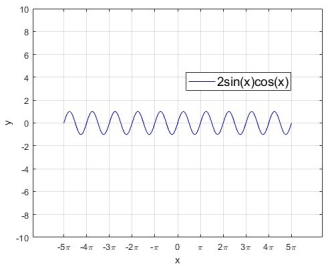
siendo y la magnitud de perturbación con respecto al tiempo y al espacio y v la velocidad con que se propaga la onda ([34], [38], [21], [32]).

Para obtener más información sobre la deducción de la Ecuación de Onda, se puede consultar el libro de Marc Figueras Atienza (ver Referencia [34]).

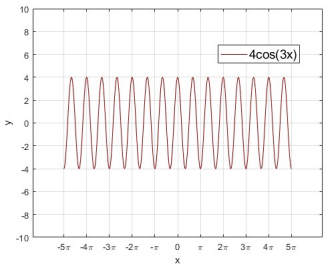
A continuación, en la Figura 1.6, se presenta un ejemplo de la superposición de tres ondas con periodos de 2π, $\frac{2\pi}{2}$ y $\frac{2\pi}{3}$, lo que se traduce en frecuencias de $\frac{1}{2\pi}$, $\frac{2}{2\pi}$ y $\frac{3}{2\pi}$, respectivamente. Esta combinación genera otra onda periódica con una frecuencia de $\frac{1}{2\pi}$ (ver 1.6(d)). Se puede apreciar que las frecuencias de las ondas más simples son múltiplos enteros de la resultante [39].



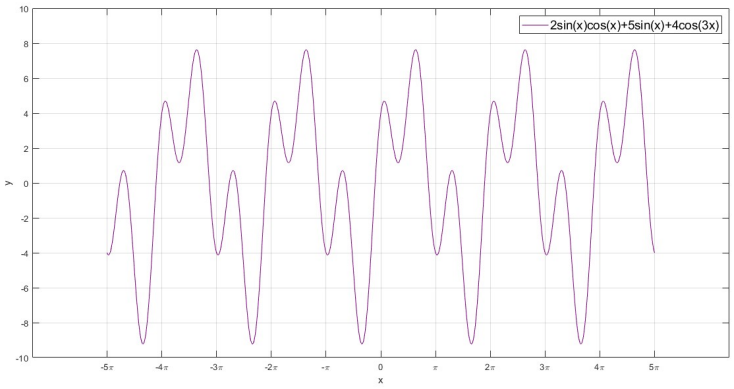
(a) $y = 5 \cdot \sin(x)$



(b) $y = 2 \cdot \sin(x) \cdot \cos(x)$



(c) $y = 4 \cdot \cos(3x)$



(d) $y = 5 \cdot \sin(x) + 2 \cdot \sin(x) \cdot \cos(x) + 4 \cdot \cos(3x)$

Figura 1.6: Superposición de tres ondas puras (Imágenes de creación propia a partir de Matlab)

1.3.3. Ondas estacionarias en una cuerda tensada con los dos extremos fijos

Definición 1.3.5. *Una **onda estacionaria** es aquella que se origina cuando dos ondas armónicas de idéntica amplitud y frecuencia son superpuestas mientras se propagan en una misma dirección pero con sentidos contrarios. Estas ondas viajeras reciben el nombre de incidente y reflejada [40].*

Definición 1.3.6. *Un **nodo** es el punto estático que se forma cuando la onda estacionaria presenta una amplitud nula. Por otro lado, un **antinodo** es el punto situado entre dos nodos donde la onda alcanza su máxima amplitud [3].*

En la Figura 1.7, quedan ilustradas las dos definiciones anteriores. Nótese también que los extremos de la onda son nodos.

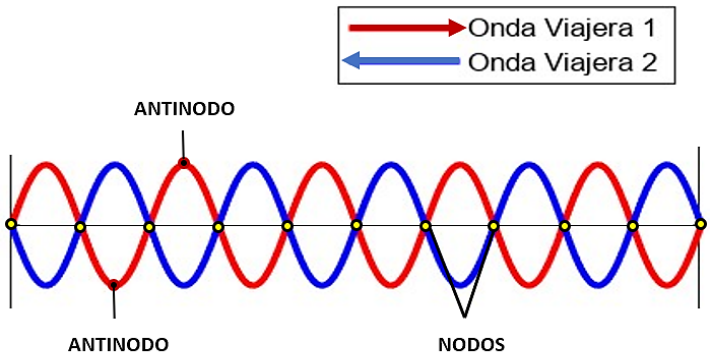


Figura 1.7: Representación de una onda estacionaria con extremos fijos (Imagen de creación propia a partir de Matlab)

Observación 1.3.1. *Es importante no confundir el concepto de nodo con* ***modos normales****, que son los patrones generados por las ondas estacionarias. En un modo normal, todas las partículas del sistema vibran con la misma frecuencia. Por ejemplo, el primer modo o modo fundamental de una cuerda vibrante corresponde a la frecuencia más baja, asociada al tono fundamental [32].*

En este escenario, se imagina una onda que viaja unidimensionalmente dentro de un medio acotado por una frontera que refleja toda la energía sin transmitirla a ningún otro medio adyacente, haciéndola regresar en sentido opuesto. Si se toma como ejemplo una cuerda tensada con sus dos extremos fijos orientada en horizontal, se denota con L su longitud finita.

A partir de 1.2, se extraen las ecuaciones de la onda incidente con desplazamiento hacia la derecha:

$$f_i(x,t) = A\sin(kx - \omega t) \tag{1.4}$$

y de la onda reflejada, con desplazamiento hacia la izquierda:

$$f_r(x,t) = A\sin(kx + \omega t + \phi) \tag{1.5}$$

Cabe señalar que la fase inicial de la onda incidente (ver Ecuación 1.4) es $\phi = 0$, ya que $f(x,t) = 0$ en el instante inicial en el origen de coordenadas. Sin embargo, la fase inicial de la onda reflejada (ver Ecuación 1.5) es desconocida, puesto que su valor dependerá de la longitud de la cuerda.

A continuación, se suman las expresiones 1.4 y 1.5 para hallar la ecuación de la onda estacionaria. En primer lugar, se aplica la fórmula de la suma de senos

de ángulos distintos, obteniéndose como resultado:

$$f_e(x,t) = f_i(x,t) + f_r(x,t) = 2A \cdot \sin(kx + \frac{\phi}{2}) \cdot \cos(-\omega t - \frac{\phi}{2}) \tag{1.6}$$

Por simetría con respecto al eje vertical, se aplica que $\cos(-\alpha) = \cos(\alpha)$ a la Ecuación 1.6:

$$f_e(x,t) = 2A \cdot \sin(kx + \frac{\phi}{2}) \cdot \cos(\omega t + \frac{\phi}{2}) \tag{1.7}$$

Se destaca que la onda estacionaria, a pesar de ser la combinación de dos ondas viajeras, no parece experimentar ningún desplazamiento visible, pero sí vibra de manera armónica [34].

Además, la ecuación 1.7 de la onda estacionaria verifica las siguientes condiciones de contorno tipo Dirichlet en los extremos fijos de la cuerda:

$f_e(0,t) = 2A \cdot \sin(\frac{\phi}{2}) \cdot \cos(\omega t + \frac{\phi}{2}) = 0$ cuando $x = 0$

$f_e(L,t) = 2A \cdot \sin(kL + \frac{\phi}{2}) \cdot \cos(\omega t + \frac{\phi}{2}) = 0$ cuando $x = L$

Como las anteriores condiciones se tienen que cumplir para cualquier instante de tiempo, se deduce que:

$$\sin(\frac{\phi}{2}) = 0 \tag{1.8}$$

de donde $\phi = 0$;

$$\sin(kL) = 0 \tag{1.9}$$

de donde $kL = n\pi \ \forall n \in \mathbb{N}$.

Ahora, por la relación del número de onda definida en la Ecuación 1.2, se tiene que $\frac{2\pi}{\lambda} \cdot L = n\pi$. Por tanto, $\forall n \in \mathbb{N}$:

$$L = n \cdot \frac{\lambda}{2} \tag{1.10}$$

Así se concluye que las ondas estacionarias se forman en estas condiciones sólo si la longitud de la cuerda es un múltiplo entero de la mitad de la longitud de la onda, tal y como queda ilustrado en la Figura 1.8 [3], [40] y [34]:

Por otra parte, se sabe que la longitud de onda λ presenta una relación con la frecuencia f de modo que $\lambda = \frac{v}{f}$. De este modo, sustituyendo en la ecuación 1.10, se obtiene $\forall n \in \mathbb{N}$:

$$f = \frac{nv}{2L} \tag{1.11}$$

La ecuación 1.11 muestra las frecuencias de los armónicos para los distintos modos de vibración n, los cuales son múltiplos exactos de la frecuencia fundamental ($n = 1$). Este resultado matemático corrobora lo mencionado previamente en la Sección 1.2 con las aportaciones de Sauveur [34].

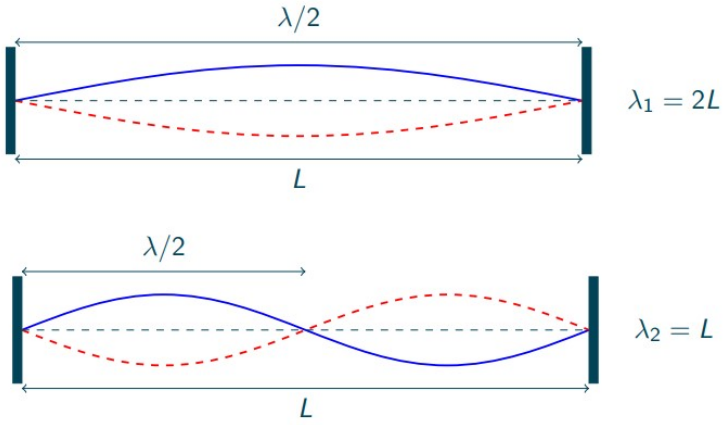


Figura 1.8: Relación entre la longitud de la cuerda y la longitud de la onda de los dos primeros modos (Imagen extraída de [3])

Verificación de la Ecuación de Onda con extremos fijos

A continuación, se va a comprobar si las ecuaciones de las ondas viajeras y de la onda estacionaria, definidas en el apartado anterior, cumplen la Ecuación de Onda 1.3.

En primer lugar, se van a extraer las derivadas parciales de la onda incidente (ver Ecuación 1.4):

$$\frac{\partial f_i}{\partial t} = -Aw\cos(kx - wt) \qquad \frac{\partial f_i}{\partial x} = Ak\cos(kx - wt)$$
$$\frac{\partial^2 f_i}{\partial t^2} = -Aw^2\sin(kx - wt) \qquad \frac{\partial^2 f_i}{\partial x^2} = -Ak^2\sin(kx - wt)$$

Despejando la expresión $-A\sin(kx - wt)$ en las ecuaciones de segundas derivadas parciales e igualando, se obtiene:

$$\frac{\partial^2 f_i}{\partial x^2} \cdot \frac{1}{k^2} = \frac{\partial^2 f_i}{\partial t^2} \cdot \frac{1}{w^2} \tag{1.12}$$

Ahora, por las equivalencias del apartado anterior, se tiene que:

$$\frac{w}{k} = \frac{2\pi}{T} : \frac{2\pi}{\lambda} = \frac{\lambda}{T} = v \tag{1.13}$$

De esta forma, sustituyendo 1.13 en 1.12 da como resultado la Ecuación de Onda 1.3:

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{v^2} \cdot \frac{\partial^2 y}{\partial t^2}$$

De forma análoga, se comprueba fácilmente que la onda reflejada (ver Ecuación 1.5) también satisface la Ecuación de Onda.

Por último, se va a realizar la comprobación para la onda estacionaria (ver Ecuación 1.7), resultante de la suma de las dos ondas viajeras y cuyas derivadas parciales son las siguientes:

$$\frac{\partial f_e}{\partial t} = -2Aw \cdot \sin(kx + \frac{\phi}{2}) \cdot \sin(wt + \frac{\phi}{2}) \qquad \frac{\partial f_e}{\partial x} = 2Ak \cdot \cos(kx + \frac{\phi}{2}) \cdot \cos(wt + \frac{\phi}{2})$$
$$\frac{\partial^2 f_e}{\partial t^2} = -2Aw^2 \cdot \sin(kx + \frac{\phi}{2}) \cdot \cos(wt + \frac{\phi}{2}) \qquad \frac{\partial^2 f_e}{\partial x^2} = -2Ak^2 \cdot \sin(kx + \frac{\phi}{2}) \cdot \cos(wt + \frac{\phi}{2})$$

Del mismo modo, despejando la expresión $-2A \cdot \sin(kx + \frac{\phi}{2}) \cdot \cos(wt + \frac{\phi}{2})$ en las ecuaciones de las segundas derivadas parciales, se consigue la siguiente igualdad:

$$\frac{\partial^2 f_e}{\partial x^2} \cdot \frac{1}{k^2} = \frac{\partial^2 f_e}{\partial t^2} \cdot \frac{1}{w^2} \tag{1.14}$$

que es equivalente a la expresión 1.12 de la onda incidente y culmina nuevamente en la Ecuación de Onda 1.3.

Esto implica que las tres ecuaciones mencionadas son soluciones válidas de la Ecuación de Onda.

1.4. Aplicaciones del Modelo de Fourier

El fenómeno del movimiento vibratorio de una cuerda generó un debate en el que participaron prominentes matemáticos como Johann y Daniel Bernoulli, L. Euler, J. D'Alembert, J.L. Lagrange y L. Dirichlet. No fue hasta mediados del siglo XIX que Joseph Fourier logró obtener una solución concluyente para dicho problema [36].

Jean Baptiste Joseph Fourier (1768-1830) redactó el tratado *Mémoire sur la propagation de la chaleur dans les corps solides*, donde afirma que toda función periódica se puede descomponer en la suma infinita de funciones seno y coseno [21].

En el ámbito de la música, el diapasón es uno de los pocos instrumentos que se aproxima a producir un tono puro y perfecto. Su sonido, con una frecuencia de 440 Hz, corresponde a la nota La_4 de la cuarta octava de un piano, y es el La

estándar empleado como referencia para afinar otros instrumentos. En cambio, en una orquesta sinfónica, la mayoría de los instrumentos musicales que la componen emiten tonos compuestos, cuyas ondas son notablemente más complejas por ser el resultado de la suma o superposición de otras ondas más simples con diferentes frecuencias y amplitudes asociadas al tono fundamental y a sus diferentes armónicos a excepción de la percusión, los placófonos y los membranófonos. Los sonidos generados por estos últimos no son el resultado de la combinación de sonidos puros asociados a los armónicos. Por tanto, al ser discordantes y no ajustarse a la serie armónica, reciben el nombre de *sobretonos inarmónicos*. Cabe señalar que en la práctica, existen factores como el roce o la naturaleza del medio en que se propaga que pueden modificar la regularidad de estas ondas, las cuales raramente son completamente armónicas.

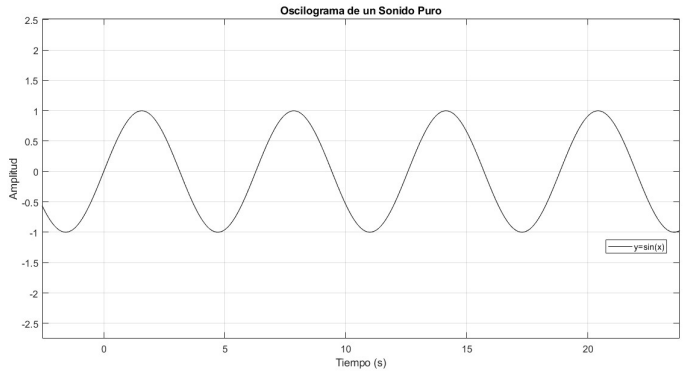
Este concepto puede expresarse a través del Modelo de Fourier (ver Capítulo 3), donde se llevará a cabo un análisis matemático de las Series de Fourier y sus transformadas. Se trata de una técnica muy poderosa y eficiente para analizar el comportamiento de las ondas ([34], [21], [36]).

En este trabajo, se centrará exclusivamente en los sonidos armónicos, dejando a un lado las complejidades de los ruidos cotidianos cuyas frecuencias no son necesariamente múltiplos enteros de una frecuencia específica, y que manifiestan un patrón más desorganizado y sin periodicidad [39]. En la Figura 1.9, se comparan los tres tipos de sonidos mencionados.

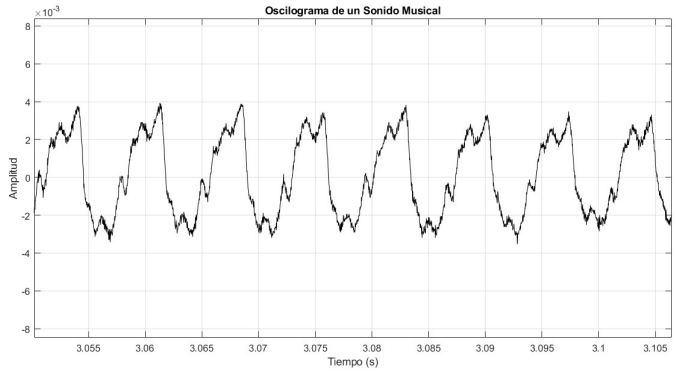
En los campos de la ciencia y la ingeniería, adquiere relevancia la manera en que se manipulan las señales con el objetivo de facilitar procesos eficientes de transmisión, compresión y reconstrucción de la información, gracias a la nueva perspectiva de análisis matemático que introdujo Fourier, alejada de una visión tradicional. En la actualidad, sus conceptos son de gran utilidad en diferentes aplicaciones, como en la modelización de sistemas, la realización de predicciones, la programación lineal y el estudio de ondas electromagnéticas.

La Transformada de Fourier se describe como "el prisma matemático que descompone una función en las frecuencias que la constituyen", análogo a cómo un prisma de cristal separa la luz en sus componentes espectrales [24].

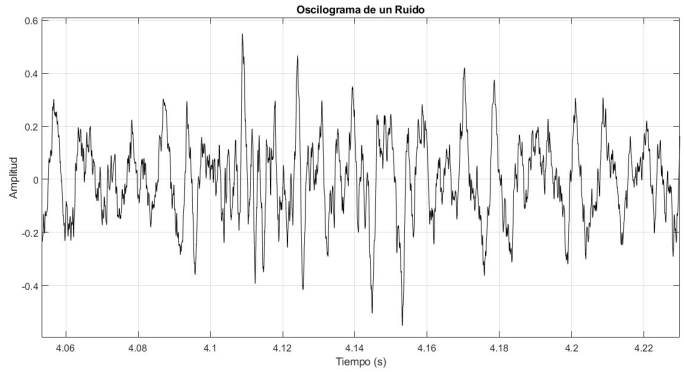
La capacidad auditiva para descomponer un tono compuesto en cada uno de sus elementos, de acuerdo con el Teorema de Fourier, es conocida como la Ley Acústica de Ohm en 1843, formulada por el físico George Simon Ohm. Este increíble don de la naturaleza permite al ser humano discernir notas individuales en combinaciones tocadas simultáneamente, en forma de acordes musicales. En cambio, la visión presenta sus limitaciones, ya que después de mezclar tres colores diferentes, el ojo no es capaz de reconocer con precisión las cantidades de cada uno, percibiendo únicamente un color resultante [21].



(a) Oscilograma de un sonido puro, equivalente a la onda producida por un diapasón en condiciones ideales



(b) Oscilograma de un sonido musical, correspondiente al acorde $C\#_4$ de un banjo



(c) Oscilograma del acorde invertido C_3 de un órgano mezclado con ruido intenso de fondo

Figura 1.9: Representación gráfica de las ondas de diferentes tipos de sonidos (Imágenes de creación propia a partir de Matlab)

2

Objetivos

El propósito principal de este Trabajo de Fin de Grado es encontrar un modelo matemático que sea capaz de reconocer y clasificar de manera efectiva un conjunto de sonidos procedentes de diferentes instrumentos musicales.

Para poder llevar a cabo dicha tarea, el contenido se va a estructurar en dos partes fundamentales, a partir de las cuales se van a establecer unos objetivos más concretos y detallados.

2.1. Creación de la base de datos transformando los sonidos generados en espectrogramas

1. Grabar un número considerable de acordes mediante un sintetizador que simule el timbre de diferentes instrumentos musicales evitando cualquier tipo de ruido externo que pueda distorsionar la señal.

2. Investigar y seleccionar una herramienta adecuada para transformar cada una de las pistas de audio en imágenes denominadas espectrogramas (tanto en 2D como en 3D) en términos de frecuencia, amplitud y tiempo.

3. Conseguir una buena resolución y claridad en los espectrogramas, ajustando los parámetros óptimos como el tamaño y el tipo de ventana seleccionada y el solapamiento entre muestras.

4. Analizar la información extraída de un espectrograma, descubriendo qué notas se encuentran presentes y apreciar sus armónicos correspondientes.

5. Comparar la riqueza armónica o timbre de cada instrumento musical a través de los espectrogramas, observando las diferencias obtenidas tras interpretar un mismo acorde por más de un instrumento diferente.

2.2. Clasificación de espectrogramas por medio de una red neuronal convolucional

Después de obtener los espectrogramas, se almacenarán en diferentes carpetas etiquetadas según el instrumento musical correspondiente. A continuación se indicarán los próximos objetivos de este segundo bloque:

6. Utilizar una arquitectura de red neuronal convolucional como GoogLeNet para entrenar el modelo con las nuevas imágenes originadas.

7. Mejorar el rendimiento del modelo mediante el ajuste inicial de los hiperparámetros.

8. Analizar las métricas del modelo utilizando datos de validación y prueba. Por consiguiente, se podrá verificar su capacidad de clasificar correctamente un conjunto de espectrogramas en las diferentes clases de instrumentos musicales.

9. Investigar posibles características o patrones visuales similares entre espectrogramas de diferentes instrumentos musicales.

10. Proponer soluciones para abordar las confusiones entre espectrogramas mediante el análisis de los errores de clasificación.

Los objetivos del trabajo recién formulados serán evaluados y discutidos en las conclusiones correspondientes al Capítulo 7.

3

Modelos de Fourier

3.1. Las Series de Fourier

Las Series de Fourier permiten descomponer una función periódica en la suma infinita de funciones periódicas simples, como senos y cosenos, cuyas frecuencias son múltiplos de la frecuencia de la función original. De esta forma, se pueden estudiar las propiedades de la función o señal y facilita también la reconstrucción de fenómenos a partir de sus componentes básicos. La información presentada en este capítulo se ha basado en el contenido de las páginas ([5], [41], [42],[43]).

En el Grado de Matemáticas las Series de Fourier aparecen en el III curso en la asignatura de Ecuaciones en Derivadas Parciales donde han sido introducidas y utilizadas para la resolución de EDP lineales en dominios acotados y considerando distintos sistemas de coordenadas.

3.1.1. Serie Trigonométrica de Fourier

Definición 3.1.1. *Sea $f(t)$ una función periódica e integrable de periodo 2π. Se define* ***Serie de Fourier*** *a la serie trigonométrica representada como:*

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(nt) + b_n \sin(nt)] \tag{3.1}$$

suponiendo que la serie converge uniformemente a dicha función en el intervalo $[-\pi, \pi]$ *y siendo* a_0, a_n *y* $b_n \in \mathbb{R}$ *los denominados coeficientes de Fourier, que se hallarán a continuación.*

Antes de obtener los valores de los coeficientes, se va a introducir un concepto teórico de ortogonalidad:

Definición 3.1.2. *Dado el conjunto de funciones* $\{1, \cos(t), \sin(t), \cos(2t), \sin(2t), ..., \cos(n), \sin(n)\}$ *que forman parte de la serie anterior (ver Ecuación 3.1), se verifica la siguiente* ***Propiedad de Ortogonalidad*** *en* $[-\pi, \pi]$*:*

$$\int_{-\pi}^{\pi} \psi(t)\chi(t)\, dt = 0 \tag{3.2}$$

para cualquier par de funciones diferentes $\psi(x)$ *y* $\chi(x)$ *del conjunto definido. Si* $\psi(t) = \chi(t)$*, entonces el resultado de la integral 3.2 es igual a* π*, excepto si ambas funciones son 1, en cuyo caso su valor sería* 2π*, coincidiendo con el periodo de la función.*

En primer lugar, se va a obtener el valor del coeficiente a_0 integrando a ambos lados la Ecuación 3.1 en el intervalo $[-\pi, \pi]$:

$$\int_{-\pi}^{\pi} f(t)\, dt = \int_{-\pi}^{\pi} \left[\frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos(nt) + b_n \sin(nt))\right] dt =$$
$$= \frac{a_0}{2}\int_{-\pi}^{\pi} dt + \sum_{n=1}^{\infty}\left[a_n \int_{-\pi}^{\pi} \cos(nt)\, dt + b_n \int_{-\pi}^{\pi} \sin(nt)\, dt\right] \tag{3.3}$$

Por los criterios de Ortogonalidad vistos en la Definición 3.1.2, las dos últimas integrales trigonométricas son cero. Por tanto, se tiene que:

$$\int_{-\pi}^{\pi} f(t)\, dt = \frac{a_0}{2}\int_{-\pi}^{\pi} dt = a_0\pi \tag{3.4}$$

donde

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\, dt \tag{3.5}$$

A continuación, se va a calcular el valor de a_n, multiplicando 3.1 por $cos(mt)$ a ambos lados (con $m \in \mathbb{N}$) e integrando en $[-\pi, \pi]$:

$$\int_{-\pi}^{\pi} f(t)\cos(mt)\,dt = \int_{-\pi}^{\pi}\left[\frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n\cos(nt) + b_n\sin(nt))\right]\cos(mt)\,dt =$$
$$= \frac{a_0}{2}\int_{-\pi}^{\pi}\cos(mt)\,dt + \sum_{n=1}^{\infty}\left[(a_n\int_{-\pi}^{\pi}\cos(nt)\cos(mt)\,dt + b_n\int_{-\pi}^{\pi}\sin(nt)\cos(mt)\,dt\right] \tag{3.6}$$

Nuevamente, se aplica la Propiedad de Ortogonalidad 3.1.2 a las integrales anteriores, las cuales son todo nulas excepto la segunda integral, en el caso en que m sea igual a n, cuyo valor correspondería a π. Por tanto, la ecuación 3.6 quedaría de la siguiente forma:

$$\int_{-\pi}^{\pi} f(t)\cos(nt)\,dt = a_n\pi \tag{3.7}$$

siendo

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos(nt)\,dt \qquad \forall n \in \mathbb{N}^* \tag{3.8}$$

Análogamente, se extrae b_n, el tercer coeficiente de Fourier, pero esta vez se multiplica a ambos lados la ecuación 3.1 por $\sin(mt)$ con $m \in \mathbb{N}$. Así, siendo $m = n$ el único caso en que la integral no se anula, se obtiene:

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\sin(nt)\,dt \qquad \forall n \in \mathbb{N} \tag{3.9}$$

Es decir, 3.8, 3.9 y 3.5 son los coeficientes de la Serie de Fourier 3.1.

3.1.2. Serie Compleja de Fourier

Las series trigonométricas de Fourier también se pueden expresar de una manera más simplificada y fácil de resolver empleando funciones exponenciales mediante las siguientes fórmulas de Euler, siendo $i = \sqrt{-1}$:

$$e^{ix} = \cos(x) + i\sin(x) \qquad y \qquad e^{-ix} = \cos(x) - i\sin(x) \tag{3.10}$$

Sumando y restando las expresiones de 3.10 con $x = nt$, se obtienen:

$$\cos(nt) = \frac{e^{int} + e^{-int}}{2} \qquad y \qquad \sin(nt) = \frac{e^{int} - e^{-int}}{2i} \tag{3.11}$$

Ahora, se van a sustituir las igualdades obtenidas 3.11 en la Ecuación 3.1:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cdot \frac{1}{2}(e^{int} + e^{-int}) + b_n \cdot \frac{1}{2i}(e^{int} - e^{-int}) \right] \tag{3.12}$$

Racionalizando el número complejo en la expresión anterior se tiene que:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cdot \frac{1}{2}(e^{int} + e^{-int}) - ib_n \cdot \frac{1}{2}(e^{int} - e^{-int}) \right] \tag{3.13}$$

y, sacando factor común:

$$\begin{aligned} f(t) = \ & \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[\frac{1}{2}(a_n - ib_n) \cdot e^{int} + \frac{1}{2}(a_n + ib_n) \cdot e^{-int} \right] = \\ & = A_0 + \sum_{n=1}^{\infty} \left[A_n e^{int} + B_n e^{-int} \right] \end{aligned} \tag{3.14}$$

donde A_0, A_n y B_n se denotan a los nuevos coeficientes complejos tales que:

$$A_0 = \frac{a_0}{2}, \qquad A_n = \frac{a_n - ib_n}{2}, \qquad B_n = \frac{a_n + ib_n}{2} \tag{3.15}$$

Por tanto, aplicando el cambio a función exponencial de los coeficientes hallados en 3.5, 3.8 y 3.9, y a partir de 3.15, se deduce que:

$$A_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t)\, dt, \tag{3.16}$$

$$\begin{aligned} A_n = \ & \frac{1}{2\pi} \left[\int_{-\pi}^{\pi} f(t) \cdot \frac{e^{int} + e^{-int}}{2}\, dt - i \int_{-\pi}^{\pi} f(t) \cdot \frac{e^{int} - e^{-int}}{2i}\, dt \right] = \\ & = \frac{1}{2\pi} \left[\frac{1}{2} \int_{-\pi}^{\pi} f(t) \cdot (e^{int} + e^{-int})\, dt - \frac{1}{2} \int_{-\pi}^{\pi} f(t) \cdot (e^{int} - e^{-int})\, dt \right] = \\ & = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) \cdot e^{-int}\, dt, \end{aligned} \tag{3.17}$$

$$\begin{aligned} B_n = \ & \frac{1}{2\pi} \left[\int_{-\pi}^{\pi} f(t) \cdot \frac{e^{int} + e^{-int}}{2}\, dt + i \int_{-\pi}^{\pi} f(t) \cdot \frac{e^{int} - e^{-int}}{2i}\, dt \right] = \\ & = \frac{1}{2\pi} \left[\frac{1}{2} \int_{-\pi}^{\pi} f(t) \cdot (e^{int} + e^{-int})\, dt + \frac{1}{2} \int_{-\pi}^{\pi} f(t) \cdot (e^{int} - e^{-int})\, dt \right] = \\ & = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) \cdot e^{int}\, dt \end{aligned} \tag{3.18}$$

A continuación, retomando la Ecuación 3.14 e incluyendo A_0 dentro del sumatorio:

$$f(t) = A_0 + \sum_{n=1}^{\infty} A_n e^{int} + \sum_{n=1}^{\infty} B_n e^{-int} = \sum_{n=0}^{\infty} A_n e^{int} + \sum_{n=1}^{\infty} B_n e^{-int} \tag{3.19}$$

El segundo sumatorio, el que contiene al coeficiente B_n, es equivalente a:

$$\sum_{n=1}^{\infty} B_n e^{-int} = \sum_{-\infty}^{n=-1} B_{(-n)} e^{int} \tag{3.20}$$

Introduciendo $B_{(-n)}$ en 3.18 se obtiene que:

$$B_{(-n)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t)\, e^{-int}\, dt = A_n \tag{3.21}$$

Finalmente, la ecuación 3.19 quedaría así:

$$f(t) = \sum_{n=0}^{\infty} A_n e^{int} + \sum_{-\infty}^{n=-1} A_n e^{int} \tag{3.22}$$

Es decir:

$$f(t) = \sum_{-\infty}^{\infty} A_n e^{int} \qquad \forall n \in \mathbb{Z} \tag{3.23}$$

Esta es la forma de la Serie Compleja de Fourier donde A_n es el coeficiente definido como 3.17.

3.1.3. Serie de Fourier para una función de periodo 2L

En este apartado, se considerarán las funciones periódicas $f(t)$ con un periodo genérico $T := 2L > 0$ en el intervalo [-L,L]. Para poder ajustar las series de Fourier a estas condiciones, primero se realizará el siguiente cambio de variable:

$$\frac{t}{x} = \frac{L}{\pi} \tag{3.24}$$

De este modo:

$$f(t) = f\left(\frac{L}{\pi} x\right) := g(x) \tag{3.25}$$

donde g se define como una función periódica dependiente de la variable x y con periodo 2π, dado que:

$$g(x + 2\pi) = f\left(\frac{L}{\pi}(x + 2\pi)\right) = f\left(\frac{L}{\pi} x + 2L\right) = f\left(\frac{L}{\pi} x\right) = g(x) \tag{3.26}$$

Aplicando este cambio a la forma compleja de las Series de Fourier, definida como:

$$g(x) = \sum_{-\infty}^{\infty} A_n e^{inx} \tag{3.27}$$

se obtiene:

$$g(x) = g\left(\frac{\pi}{L}t\right) = f\left(\frac{L}{\pi}\frac{\pi}{L}t\right) = f(t) = \sum_{-\infty}^{\infty} A_n e^{i\frac{n\pi t}{L}} \tag{3.28}$$

Como $w_0 = \frac{2\pi}{T} = \frac{\pi}{L}$ es la frecuencia angular fundamental, entonces:

$$f(t) = \sum_{-\infty}^{\infty} A_n e^{inw_0 t} \tag{3.29}$$

y cuyo coeficiente es equivalente a:

$$A_n = \frac{1}{2L}\int_{-L}^{L} f(t)\, e^{-inw_0 t}\, dt \tag{3.30}$$

La ecuación 3.29 describe la suma infinita de funciones de onda sinusoidal expresadas en términos exponenciales, cada una con una frecuencia distinta. El valor $w_n = nw_0$ simboliza el n-ésimo armónico de dicha función periódica.

Este cambio de variable también es aplicable de manera análoga a las Series de Fourier trigonométricas tratadas en la Subsección 3.1.1.

3.2. La Transformada de Fourier

Las series de Fourier son una técnica eficaz para el análisis de señales periódicas. No obstante, en muchos casos prácticos, las señales no son necesariamente periódicas, por lo que se va a emplear una variante conocida como la Transformada de Fourier para su estudio.

Primeramente, para hallar su expresión se parte de la ecuación exponencial compleja de Fourier hallada en el apartado anterior. Así, al sustituir el valor del coeficiente A_n (ver 3.30) en la Ecuación 3.29 da lugar a:

$$f(t) = \sum_{-\infty}^{\infty} \left[\frac{1}{2L}\int_{-L}^{L} f(t)\, e^{-inw_0 t}\, dt\right] e^{inw_0 t} \tag{3.31}$$

donde $f(t)$ es la función periódica de $T = 2L$, integrable en el intervalo $[-\pi, \pi]$ y definida en todo el conjunto de los números reales.

Teniendo en cuenta que la distancia entre armónicos consecutivos se define como:

$$\Delta w = (n+1)w_0 - nw_0 = w_0 = \frac{\pi}{L} \tag{3.32}$$

Entonces, la anterior expresión se puede escribir como:

$$f(t) = \sum_{-\infty}^{\infty} \left[\frac{\Delta w}{2\pi} \int_{-L}^{L} f(t)\, e^{-inw_0 t}\, dt \right] e^{inw_0 t} \tag{3.33}$$

y, reordenando los términos:

$$f(t) = \frac{1}{2\pi} \sum_{-\infty}^{\infty} \left[\int_{-L}^{L} f(t)\, e^{-inw_0 t}\, dt \right] \Delta w e^{inw_0 t} \tag{3.34}$$

Intervalos no acotados

Ahora, considerando el límite cuando el periodo tiende a infinito, es decir, si $T = 2L \to \infty$, el incremento de separación Δw se vuelve infinitesimalmente pequeño y se transforma en el diferencial dw, provocando que las frecuencias se encuentren más próximas entre sí. Por tanto, el término discreto de frecuencias, nw_0 es sustituido por w, que representa las frecuencias de un modo continuo (en todo $\mathbb{R}$).

En este proceso, la Serie de Fourier se convierte en la Transformada de Fourier de una función no periódica:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(t)\, e^{-iwt}\, dt \right] e^{iwt}\, dw \tag{3.35}$$

De este modo, se definen el siguiente par de funciones:

$$F(w) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t)\, e^{-iwt}\, dt \tag{3.36}$$

y

$$f(t) = \mathcal{F}^{-1}(F(w)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(w)\, e^{iwt}\, dw \tag{3.37}$$

La función $\mathcal{F}$ simboliza la Transformada de Fourier (ver 3.36), la cual se encarga de convertir $f(t)$ del dominio temporal al frecuencial, devolviendo la función compleja denotada como $F(w)$. Por otro lado, la función $\mathcal{F}^{-1}$ representa la Transformada Inversa de Fourier (ver 3.37) cuya misión es intercambiar el dominio frecuencial de $F(w)$ por el dominio temporal, recuperando así la función original.

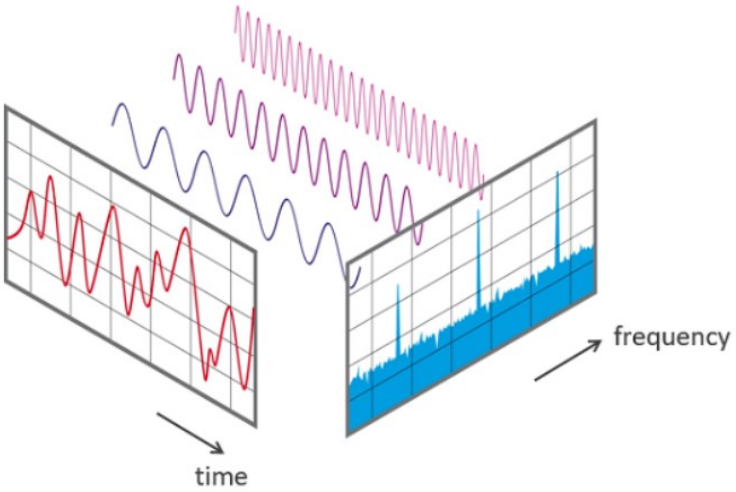


Figura 3.1: (Imagen extraída de [4])

La expresión 3.36 de la Transformada de Fourier también puede presentarse en forma polar:

$$F(w) = |F(w)|\, e^{i\phi(w)} \tag{3.38}$$

donde cada frecuencia w va asociada a un coeficiente complejo $F(w)$, cuyo módulo indica la magnitud o amplitud del espectro y su argumento $\phi(w)$ es la fase, de acuerdo a las siguientes igualdades:

$$|F(w)| = \sqrt{Re^2\left[F(w)\right] + Im^2\left[F(w)\right]} \tag{3.39}$$

y

$$\tan(\phi(w)) = \frac{Im\left[F(w)\right]}{Re\left[F(w)\right]} \tag{3.40}$$

teniendo en cuenta que la parte real e imaginaria representan las componentes del coseno y seno, respectivamente, de una frecuencia w.

En resumidas cuentas, la Transformada de Fourier es una herramienta matemática poderosa que permite separar una señal no periódica en sus armónicos correspondientes, cada uno asociado a una frecuencia determinada, tal y como aparece ilustrado en la Figura 3.1:

3.2.1. La Transformada Discreta de Fourier (DFT)

En muchas aplicaciones prácticas, las señales continuas deben ser convertidas a formato digital para su posterior análisis, por lo que será necesario tomar muestras discretas de la señal original.

Sea N el número finito de puntos o muestras repartidas a intervalos regulares de tiempo, donde la distancia entre ellas se denotará con la letra $\mathcal{T}$. Dada ω_u la frecuencia representada por la DFT, asociada a un índice u. Entonces, se define:

$$\omega_u = \frac{2\pi}{N\mathcal{T}}u, \quad \text{donde } u \in \{0, 1, 2, ..., N-1\} \tag{3.41}$$

Con esta información, se deduce la expresión elemental, para $\mathcal{T} = 1$, de la Transformada Discreta de Fourier, que asocia cada función f con otra función F:

$$F(u) = \mathcal{F}(f(k)) = \sum_{k=0}^{N-1} f(k)\, e^{-i\frac{2\pi uk}{N}}, \quad \text{para } u \in \{0, 1, ..., N-1\} \tag{3.42}$$

donde $F(u)$ recoge el coeficiente complejo de la Transformada de Fourier para una frecuencia u dada.

Su transformada inversa se define como:

$$f(k) = \mathcal{F}^{-1}(F(u)) = \frac{1}{N}\sum_{u=0}^{N-1} f(u)\, e^{i\frac{2\pi uk}{N}}, \quad \text{para } k \in \{0, 1, ..., N-1\} \tag{3.43}$$

Como observación, el índice k recorre los puntos discretos de la señal en el dominio temporal, mientras que el índice u recorre las frecuencias discretas en el dominio frecuencial.

Cabe señalar que la Transformada Discreta de Fourier considera una periodicidad en los datos. Esto significa que, dada una sucesión $\{f(k)\}$ con $k \in \mathbb{N}^*$, periódica y de módulo N se cumple que: $f(k+N) = f(k)$. En otras palabras, el contenido entre $f(0)$ y $f(N-1)$ es idéntico a $f(N)$ y $f(2N-1)$. Y, por tanto, $F(u)$ vuelve a presentar una periodicidad, con periodo N, tal que:

$$F(u+N) = F(u), \qquad \text{con } u \in \mathbb{N}^* \tag{3.44}$$

Además, la DFT también genera una simetría par en el espectro, de modo que $|F(u)| = |F(-u)|$ (ver Figura 5.7).

Otra manera de expresar la Transformada Discreta de Fourier 3.42 es mediante su forma matricial, como se desarrollará a continuación:

Se consideran $W_N = e^{-i\frac{2\pi}{N}}$ y $\widehat{W_N} = W_N^{-1} = e^{i\frac{2\pi}{N}}$ las raíces N-ésimas de la unidad. Ahora, $W_N^N = e^{-2\pi i} = \cos(-2\pi) + i\sin(-2\pi) = 1$ y además se tiene que:

$$W_N^{jN} = 1 \qquad \forall j \in \mathbb{Z} \tag{3.45}$$

En consecuencia:

$$1 + W_N^j + W_N^{2j} + ... + W_N^{(N-1)j} = \begin{cases} N & \text{si } j \text{ es múltiplo de } N, \\ 0 & \text{en otro caso.} \end{cases} \tag{3.46}$$

La comprobación de la suma cuando j no es un múltiplo de N se realiza sencillamente mediante la fórmula de progresión geométrica, teniendo en cuenta que $W_N^j \neq 1$ en este caso. Es decir:

$$S_N = \frac{W_N^{jN} - 1}{W_N^j - 1} = \frac{e^{-2\pi ji} - 1}{W_N^j - 1} = 0 \tag{3.47}$$

A partir de la notación definida, la expresión de la DFT puede también representarse como:

$$F(u) = \mathcal{F}(f(k)) = \sum_{k=0}^{N-1} f(k)\, W_N^{uk}, \quad \text{para } u \in \{0, 1, ..., N-1\} \tag{3.48}$$

Por otro lado, su inversa queda establecida así:

$$f(k) = \mathcal{F}^{-1}(F(u)) = \frac{1}{N} \sum_{u=0}^{N-1} f(u)\, W_N^{-uk}, \quad \text{para } k \in \{0, 1, ..., N-1\} \tag{3.49}$$

Para concluir, la forma matricial de la Transformada Discreta de Fourier quedaría como:

$$F(u) = A(W_N) \cdot f(k) \tag{3.50}$$

siendo $A(W_N)$ la matriz compleja central de dimensión N x N:

$$\begin{pmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ \vdots \\ F(N-1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & W_N^3 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_N^6 & \cdots & W_N^{2(N-1)} \\ 1 & W_N^3 & W_N^6 & W_N^9 & \cdots & W_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & W_N^{3(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ \vdots \\ f(N-1) \end{pmatrix}$$

Cabe destacar que, para simplificar los cálculos, los coeficientes $W_N^{2(N-1)}$, $W_N^{3(N-1)}$ y $W_N^{(N-1)(N-1)}$ se pueden reescribir como:

$$W_N^{2(N-1)} = W_N^{N-2}, \quad W_N^{3(N-1)} = W_N^{N-3)} \quad y \quad W_N^{(N-1)(N-1)} = W_N$$

En cambio, la forma matricial de su Transformada Inversa tendría la siguiente estructura:

$$f(k) = \frac{1}{N} A(W_N^{-1}) \cdot F(u) \tag{3.51}$$

debido a que $A(W_N) \cdot A(W_N^{-1}) = A(W_N^{-1}) \cdot A(W_N) = N \cdot I_N$, siendo W_N^{-1} el conjugado complejo de W_N.
Esto se deduce a partir de la Ecuación 3.46, sabiendo que $W_N \cdot W_N^{-1} = 1$.

3.2.2. La Transformada Rápida de Fourier (FFT)

Después de encontrar las expresiones de la Transformada Discreta de Fourier y de su inversa (véanse 3.42 y 3.43), realizar el cálculo de las ecuaciones lineales resultantes, de orden $O(N^2)$, podría originar un consumo computacional excesivo cuando los valores de N son muy elevados, lo cual se traduciría también en una demora de tiempo indeseada.

Como consecuencia, Cooley y Tukey idearon un algoritmo en el año 1965, conocido como *La Transformada Rápida de Fourier* (FFT) de orden $O(N \log_2(N))$ [44], con el fin de disminuir el número de operaciones realizadas, obteniéndose el mismo resultado final pero de una forma mucho más efectiva. Esto es de especial utilidad en algunas aplicaciones prácticas, donde se requiere de un cálculo a tiempo real de la Transformada de Fourier. Por ejemplo, dada una señal con $N = 4096$ puntos, se necesitaría un total de 16777216 operaciones en el caso de la DFT. En cambio, aplicando el algoritmo FFT, este valor se convierte en 49152.

Para conocer con mayor profundidad el desarrollo de este algoritmo, se puede visitar el Apéndice B.

A partir de la Ecuación 3.48 y con la notación de W_N previamente definida, la Transformada Discreta de Fourier quedaría como:

$$F(u) = \sum_{k=0}^{N/2-1} f(2k)\, W_{N/2}^{uk} + W_N^u \sum_{k=0}^{N/2-1} f(2k+1)\, W_{N/2}^{uk} = E(k) + W_N^u\, O(k) \tag{3.52}$$

y

$$F(u + \frac{N}{2}) = E(k) - W_N^u\, O(k) \tag{3.53}$$

donde $E(k)$ y $O(k)$ representan las DFTs de los índices pares e impares, respectivamente. Cada una de estas DFTs tiene una dimensión que es la mitad de la dimensión original.

3.2.3. La Transformada de Fourier de Tiempo Reducido (STFT)

En contraste con la Transformada de Fourier estándar, que proporciona un resumen de las frecuencias contenidas en una señal durante todo el periodo de tiempo analizado, la Transformada Discreta de Fourier de Tiempo Reducido (STFT) es una herramienta eficaz para examinar con detalle cómo se distribuyen dichas frecuencias a lo largo del tiempo. Esta técnica fue introducida por Denis Gabor en 1946 y utiliza un método conocido como *ventaneado*, que permitirá el análisis de forma individual de los diferentes segmentos en los que se divide la señal, tal y como se explicará a continuación. Para el desarrollo de este apartado se han empleado también como referencia ([45], [46], [47], [48], [5]).

En primer lugar, se aplica sobre la señal $f(k)$ una **ventana de análisis** o **ventana temporal** $w(k)$ de longitud M puntos, cuyo valor se encuentra definido únicamente dentro de un intervalo específico. Esta ventana se va trasladando hacia adelante con saltos de R muestras, por lo que cada nuevo segmento se superpone con el anterior en $L = M - R$ muestras. De esta forma, se calcula la Transformada Discreta de Fourier (DFT) en cada una de las posiciones sucesivas que ocupa la ventana a lo largo de la señal, tal y como se muestra en la Figura 3.2.

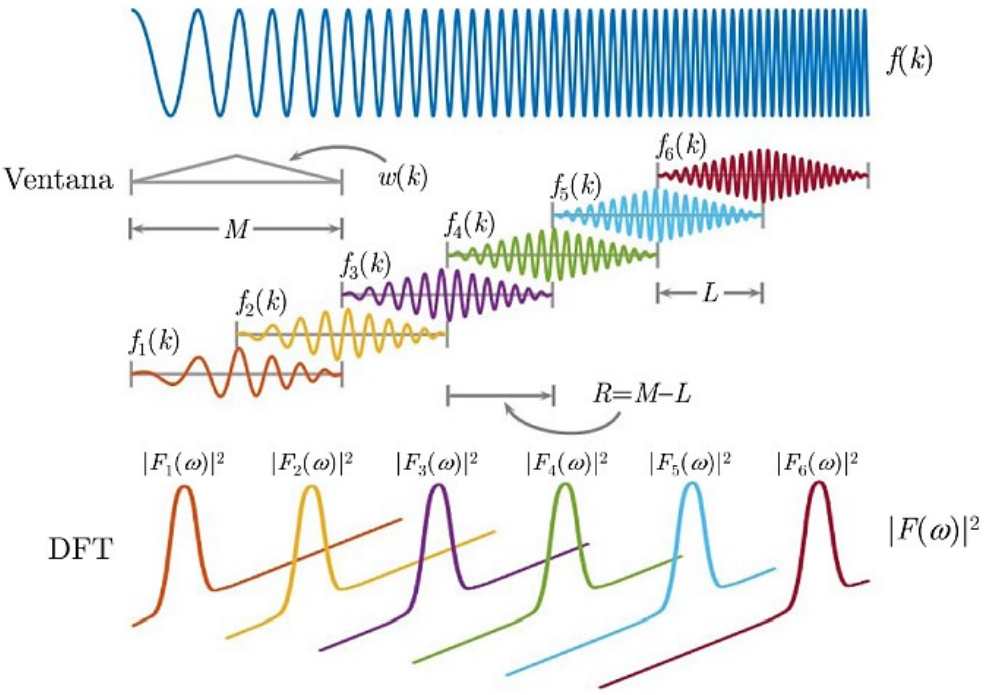


Figura 3.2: Representación gráfica de la STFT (Imagen extraída de [5])

La STFT de define de la siguiente manera, donde el espectro $F_m(u)$ representa la DFT de la porción de señal presente en la ventana de análisis en el instante mR, siendo m la posición que ocupa la ventana de análisis:

$$F_m(u) = \sum_{k=0}^{M-1} f(k)\ w(k - mR)\, e^{-i\frac{2\pi uk}{M}} \quad \text{para } u = 0, 1, ..., M \tag{3.54}$$

Finalmente, los datos obtenidos tras los múltiples análisis realizados sobre una señal se almacenan en una matriz de números complejos cuya representación gráfica se lleva a cabo por medio de un **espectrograma bidimensional** (ver 5.2.3) o **tridimensional** (ver 5.2.4). Los espectrogramas son ilustraciones gráficas que proporcionan información detallada sobre la magnitud de cada punto en el dominio tiempo-frecuencia. En otras palabras, son mapas que reflejan la evolución de los componentes de la señal a lo largo del tiempo, lo que ayuda a comprender mejor su comportamiento y cómo varían en cada momento. Se calculan elevando al cuadrado el valor de la magnitud obtenida tras aplicar la La Transformada de Fourier de Tiempo Reducido:

$$espectrograma\ f(k) \equiv |F_m(u)|^2$$

La elección del tamaño de la ventana de análisis afecta a la resolución tanto en el dominio de frecuencia como en el tiempo. Cuando se utiliza una ventana estrecha, se examinan fragmentos de la señal más diminutos, lo cual genera una buena precisión en la variable temporal pero una capacidad limitada para distinguir entre diferentes frecuencias. En cambio, al ampliar las dimensiones de la ventana, se logra una mejor resolución en frecuencia, pero el factor del tiempo empeora

Esta restricción de la STFT para ofrecer una resolución óptima en los dominios de frecuencia y tiempo simultáneamente se fundamenta en el *Principio de Incertidumbre de Heisenberg*, el cual, extrapolándolo al contexto de la Transformada de Fourier, sugiere que no es posible obtener una representación detallada de la evolución tiempo-frecuencia. Es decir, no se puede discernir con absoluta certeza qué frecuencias están presentes en un momento concreto, pero sí permite identificar las diferentes frecuencias en un determinado intervalo de tiempo.

4

Sistemas de redes neuronales de aprendizaje supervisado

Si se pretende instruir a un niño para que sepa identificar visualmente un piano de entre toda la gran variedad de instrumentos musicales, el paso a seguir es mostrarle diferentes ejemplos y decirle: "Esto es un piano" o "Esto no es un piano". Cuando por fin haya logrado asimilar e interiorizar el concepto de "Piano" y al enfrentarse a nuevos instrumentos, se encontrará preparado para determinar si se trata realmente de un piano o no.

Las redes neuronales artificiales replican este proceso, simulando la estructura y el funcionamiento de las neuronas biológicas que componen el cerebro humano, de modo que su capacidad de aprendizaje se asemeja a la de un niño a la hora de identificar un piano. Este tipo de red, basada en el procesamiento de la información recibida y en la elaboración de una salida específica, se emplea para abordar una amplia variedad de problemas, como el reconocimiento de patrones dentro de un conjunto de imágenes etiquetadas a través del entrenamiento, obteniendo una clasificación eficiente.

Sin embargo, ¿Existen conjuntos de imágenes que una red neuronal pueda reconocer con mayor precisión que el ojo humano? Efectivamente. Un ejemplo sería el conjunto de espectrogramas de acordes musicales creados para este trabajo. Son imágenes que pueden presentar diferencias extremadamente sutiles, complicando así su distinción a simple vista, pero una red neuronal entrenada con una gran cantidad de datos podrá resolver de manera adecuada este problema.

4.1. Analogía de una neurona biológica y una neurona artificial

En primer lugar, para comprender el funcionamiento de las redes neuronales, es necesario recordar brevemente el comportamiento de los sistemas neuronales biológicos en los cuales se basan ([10], [8], [43], [49], [50]).

Las neuronas transportan información a través de impulsos nerviosos que comienzan en las ramificaciones de uno de sus extremos, conocidas como **dendritas**. Dichos impulsos avanzan hasta alcanzar el **núcleo** o **soma**, el cual se encarga de procesar la información y elaborar una respuesta que será enviada al **axón**, la otra parte terminal de la neurona. Finalmente, esta nueva información se transmite a las dendritas de la siguiente neurona mediante un proceso denominado **sinapsis**.

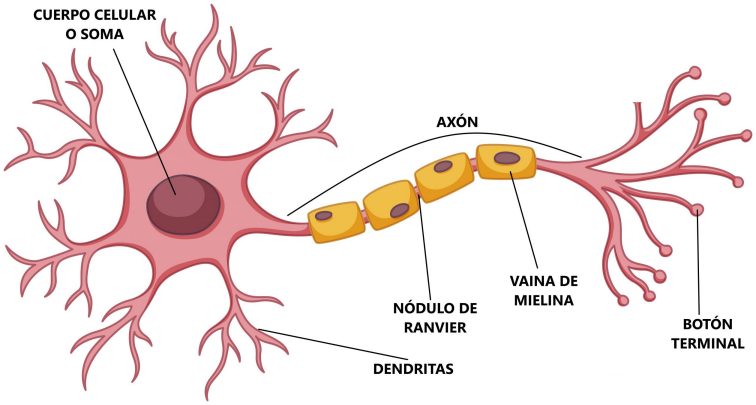


Figura 4.1: Partes de una neurona biológica (Imagen extraída de [6])

A continuación, se describe la construcción de una red neuronal artificial empleando operaciones numéricas reales para intentar imitar el mecanismo real.

La componente principal de la red es la **neurona**, también denominada **perceptrón**, y se encuentra formada por un conjunto de N dendritas, que equivale al número total de **entradas** que introducen información desde el exterior o a partir de una neurona anterior. Se denota como $x_i \ \forall i \in \{1, ..., N\}$ a los valores de entrada de una neurona.

Cada uno de estos valores va asociado a un **peso sináptico** w_i que se establece al principio de manera aleatoria pero se va modificando posteriormente a lo largo del entrenamiento. Por tanto, esta adaptación permite que la neurona genere respuestas con una mayor precisión.

En el núcleo, las entradas y los pesos se combinan y se aplica una **función de activación** al resultado obtenido con el fin de introducir una no linealidad y acotándolo dentro de un intervalo específico, facilitando así su aprendizaje.

De este modo, la señal procesada y se propaga a través del axón sirviendo como entrada para la siguiente neurona o como la salida final de la red neuronal. Cabe destacar que este valor es único para cada neurona específica. Por tanto, se tiene que:

$$y = f\left(\sum_{i=1}^{i=N} x_i w_i + b\right) \tag{4.1}$$

donde b es el umbral o parámetro auxiliar que permite el desplazamiento de la función de activación para encontrar el resultado óptimo del problema.

La Figura 4.2 ilustra, de manera esquematizada, las partes de una neurona artificial recién explicadas:

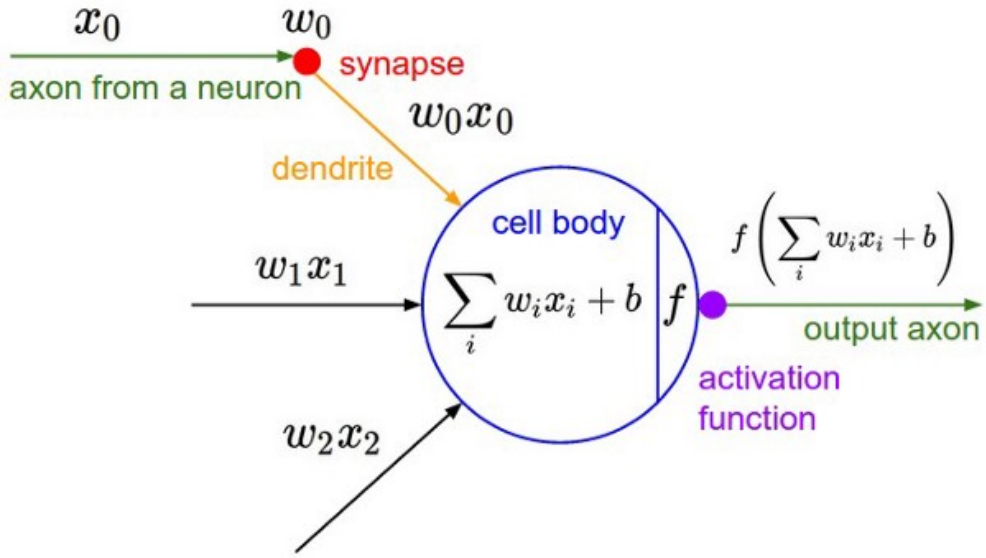


Figura 4.2: Elementos de una neurona artificial (Imagen extraída de [7])

4.2. Estructura de una Red Neuronal Artificial

Después de haber analizado el funcionamiento de un perceptrón, se va a proceder a enlazar múltiples de estos para formar redes neuronales más sofisticadas (ver Figura 4.3). Toda RNA presenta la siguiente estructura ([51], [10], [8], [50]):

- **Capa de Entrada:** recibe del exterior los datos que van a ser procesados por la red neuronal.
- **Capas Ocultas:** permanecen en la zona interna de la red y no interactúan directamente con el ambiente externo. Puede existir un número amplio de capas ocultas con neuronas interconectadas de diferentes formas, haciendo más sofisticada la capacidad de una red a la hora de tomar decisiones.

- **Capa de Salida:** corresponde con la etapa final de la red, donde se proporciona al exterior un resultado final o predicción a partir de los datos de entrada.

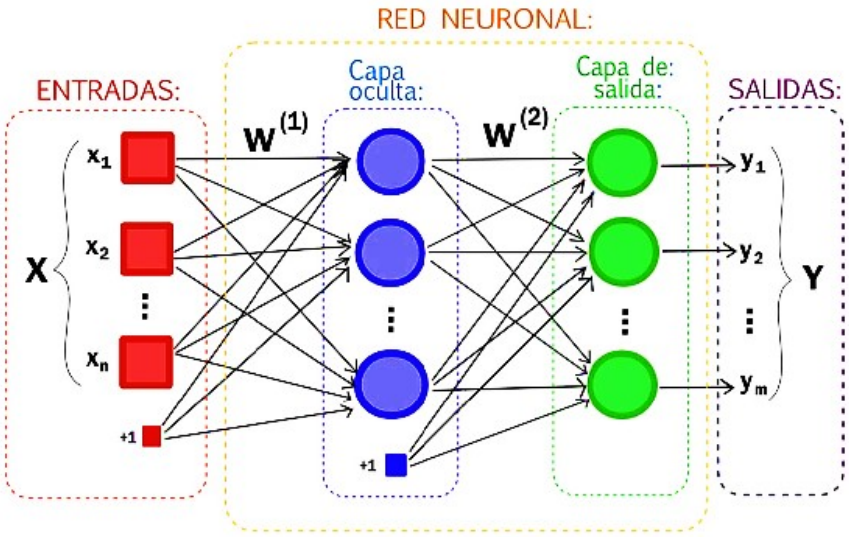


Figura 4.3: Estructura de una red neurona artificial de múltiples capas (Imagen extraída de [8])

Ahora, se va a considerar la siguiente notación formal (inspirada en [52]) para una red neuronal que consta de L capas de profundidad y N_l neuronas en cada capa l, para cualquier valor de l comprendido entre 1 y L:

- **Datos de entrada a la red**: se almacenan en el vector $X = (x_1, x_2, ..., x_{N_1})^t$.
- **Datos de salida reales**: quedan recogidos en el vector $Y = (y_1, y_2, ..., y_{N_L})^t$.
- **Pesos**: se representan como $w_{ij}^{(l)}$ y miden el nivel de conexión entre la neurona i de la capa (l) con la neurona j de la capa $(l-1)$. En forma matricial, presentan la siguiente forma para cada capa l:

$$W_l = \begin{pmatrix} w_{11}^{(l)} & \cdots & w_{1\,N_{l-1}}^{(l)} \\ \vdots & \ddots & \vdots \\ w_{N_l\,1}^{(l)} & \cdots & w_{N_l\,N_{l-1}}^{(l)} \end{pmatrix}$$

- **Sesgo** (o valor umbral): se denota como $b_i^{(l)}$ dentro de la neurona i perteneciente a la capa l. El conjunto de todos los sesgos se puede expresar a través del vector $B_l = (b_1^{(l)}, b_2^{(l)}, ..., b_{N_l}^{(l)})^t$.
- **La Función de Activación** se define como f_l para las neuronas de una capa l y su expresión se determinará en la próxima sección.

- **Las entradas y salidas generales**, denotadas como $a_i^{(l)}$ y $z_i^{(l)}$ respectivamente, son los valores obtenidos antes y después de aplicarse la función de activación en la neurona i de la capa l. De esta forma, se tiene que:

$$z_i^{(l)} = f_l(a_i^{(l)}) = f_l\left(\sum_{j=1}^{N_{l-1}} z_j^{(l-1)} w_{ij}^{(l)} + b_i^{(l)}\right) \qquad \forall i \in \{1, 2, ..., N_l\} \tag{4.2}$$

Además, se puede apreciar, por la notación definida, que:

$$z_j^1 = x_j \qquad y \qquad z_j^L = \hat{y}_j \qquad \forall j \in \{1, 2, ..., N_1\}$$

siendo $\hat{y}_j$ la salida obtenida por la red neuronal.

4.3. Aprendizaje Supervisado durante el entrenamiento de la red

Existen múltiples estrategias para el entrenamiento de una red neuronal. Sin embargo, este estudio se enfocará exclusivamente en el Método de Aprendizaje Supervisado para un conjunto de imágenes etiquetadas ([53], [54], [43], [50]).

Se hace un pequeño paréntesis para informar al lector que el conjunto total de los datos a analizar se va a fragmentar en tres subconjuntos diferentes, donde la mayor parte de ellos se empleará para el entrenamiento de la red, dejando un porcentaje menor para la validación y la prueba, que servirán para evaluar el rendimiento de clasificación tal y como se detallará en el apartado 5.3.1 de Métodos y Materiales.

Volviendo al tema, el Aprendizaje Supervisado se distingue por presentar datos conocidos, tanto de entrada como de salida, de modo que la capacidad de la red para aprender a clasificar de manera efectiva dependerá de la calidad y variedad de los ejemplos proporcionados. Además, este proceso se realizará bajo la monitorización de una entidad externa, a menudo referida como *supervisor* o *maestro*, que evaluará si la respuesta generada por la red se ajusta o no a la salida esperada a partir de cada imagen introducida.

En consecuencia, si la red no está etiquetando de forma correcta las imágenes, el supervisor llevará a cabo una modificación de los valores de los pesos y sesgos de cada una de las conexiones para así lograr obtener un mejor rendimiento. Es decir, el mecanismo de la red es aprender a partir de los errores para mejorar la predicción en cada iteración durante el entrenamiento. Para ello, se van a utilizar algoritmos de optimización que van ajustando estos parámetros con el objetivo de minimizar el error en las predicciones.

Pero antes de todo, es necesario describir las funciones de activación integradas en la arquitectura de GoogLeNet, que es la red neuronal utilizada en este trabajo para la clasificación de imágenes.

4.3.1. Funciones de Activación

GoogLeNet utiliza principalmente estas dos funciones ([55], [8], [49], [52], [49]):

Unidad Lineal Rectificada (ReLU)

Esta función activa la neurona sólamente si la entrada es positiva, en cuyo caso la salida coincide con la entrada. En cambio, si la entrada es negativa, la función devuelve un valor nulo, tal y como se representa a continuación:

$$\text{ReLU}(x) = f(x) = \text{máx}(0, x) \tag{4.3}$$

donde su derivada es:

$$\text{ReLU}'(x) = f'(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases} \tag{4.4}$$

ReLU es la función de activación más empleada debido a su simplicidad y eficiencia, y destaca en las redes neuronales convolucionales para el procesamiento de imágenes.

SoftMax o Función Exponencial Normalizada

A partir de un vector de entrada x_i, la función SoftMax genera un vector de probabilidades como salida comprendido entre 0 y 1, y cuya suma es uno. Esto significa que calcula las probabilidades correspondientes a cada una de las clases K posibles para una imagen específica y, de esta manera, las imágenes serán clasificadas en función de la máxima probabilidad obtenida.

$$\text{SoftMax}(x_i) = f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{4.5}$$

con derivada:

$$\text{SoftMax}'(x_i) = \frac{\partial f(x_i)}{\partial x_j} = \begin{cases} f(x_i)(1 - f(x_j)), & \text{si } i = j \\ -f(x_j)f(x_i), & \text{si } i \neq j \end{cases} \tag{4.6}$$

La función SoftMax es fundamental para sistemas de clasificación multiclase, siendo ampliamente utilizada en las capas finales de la red neuronal.

4.3.2. Función de coste o pérdida

La función de coste se encarga de medir la diferencia entre el valor de la salida predicha por la red neuronal y el valor real deseado. La función que se va a emplear para cuantificar el error se presenta a continuación ([55], [52]):

Entropía Cruzada Categórica

$$C = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{N_L} y_{ij}\log(\hat{y}_{ij}) \tag{4.7}$$

siendo n el número de datos del entrenamiento y además:

- y_{ij} representa el valor original del dato i en la clase j, que será 0 ó 1 dependiendo de si coincide o no con la clase correspondiente.
- $\hat{y}_{ij}$ simboliza el valor predicho por la red neuronal del dato i en la clase j, mediante las probabilidades calculadas gracias a la función SoftMax.

Su derivada sería de la forma:

$$C' = \frac{\partial C}{\partial \hat{y}_{ij}} = -\frac{1}{n}\cdot\frac{y_{ij}}{\hat{y}_{ij}} \tag{4.8}$$

4.3.3. Algoritmo de Propagación hacia adelante

Una vez quedan establecidos por defecto los pesos y sesgos iniciales que se van a ir modificando durante el proceso de entrenamiento, la información se transmite desde la capa inicial hasta la capa final de la red neuronal para así generar una predicción. El valor de salida de las neuronas queda determinado en base a los valores de salida de las neuronas de la capa precedente, tal y como ya se definió en la Fórmula 4.2 ([55], [52], [54]).

Por tanto, gracias a este algoritmo, conocido en inglés como **Forward Propagation**, se puede determinar de manera recursiva la salida de la red empleando los datos de entrada recogidos en el vector X. Es decir:

$$\hat{Y} = f_L(W_L(f_{L-1}(W_{L-1}(...f_1(W_1X + B_1) + ...) + B_{L-1}) + B_L) \tag{4.9}$$

Con la información obtenida en la salida de la red neuronal, el Algoritmo

de Retropropagación y el Método de Descenso del Gradiente Estocástico con Momento tendrán la misión de encontrar la configuración óptima de los pesos y sesgos que reduzca al mínimo la función de coste mencionada, tal y como se explicará en las próximas subsecciones.

4.3.4. Descenso del Gradiente Estocástico con Momento

El propósito del aprendizaje en las redes neuronal es minimizar el error en la función de coste, buscando la mayor precisión posible. Para lograrlo, se utiliza la técnica de optimización del descenso del gradiente, la cual consiste en ajustar repetidamente los parámetros del modelo en el sentido contrario al gradiente de dicha función [56], [55], [52], [9], [57], [51], [58], [10]).

Un claro ejemplo para comprender este método es el siguiente. Se puede imaginar a una persona situada en la cima de una colina, la cual desea descender hasta el río ubicado en el punto más bajo. La gran cantidad de niebla dificulta su orientación, pero decide seguir una regla simple: elegir el camino con pendiente más pronunciada. En este caso, su posición actual en la colina simboliza los parámetros actuales de la red neuronal, mientras que el río es el mínimo de la función de coste. En este punto, el gradiente, asociado a la pendiente, es cercano a cero, indicando una llanura.

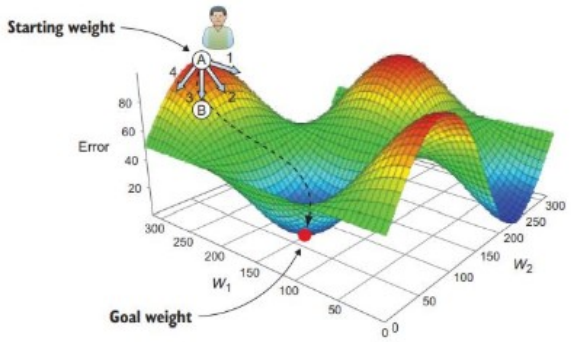


Figura 4.4: Descenso del gradiente (Imagen extraída de [9])

Una variante de este método es el Descenso del Gradiente Estocástico con Momento **(SGDM)**, donde se añade el término de *momento* con el objetivo de reducir las fluctuaciones y mejorar la estabilidad en el proceso de optimización al actualizar los parámetros, considerando las iteraciones anteriores. En otras palabras, consiste en suavizar el camino hasta alcanzar el mínimo. Para ello, se emplea el parámetro γ, comprendido entre 0 y 1, que indica el nivel de información que se conserva de los parámetros del paso previo. Si este valor es nulo, se trataría de un problema del Descenso del Gradiente Estocástico (SGD).

Siendo p el número de iteraciones, los nuevos pesos y sesgos de la siguiente

iteración se modificarían de la siguiente manera para una capa l:

$$[W_l]_{p+1} = [W_l]_p - \alpha \cdot \nabla_{W_l} C_p + \gamma \cdot ([W_l]_p - [W_l]_{p-1}) \tag{4.10}$$

$$[B_l]_{p+1} = [B_l]_p - \alpha \cdot \nabla_{B_l} C_p + \gamma \cdot ([B_l]_p - [B_l]_{p-1}) \tag{4.11}$$

donde ∇C_p es el gradiente de la Función de Coste en función de los pesos o sesgos, y $\alpha > 0$ es la tasa de aprendizaje.

Una tasa de aprendizaje es un hiperparámetro que se asigna antes del entrenamiento y que equivaldría a la longitud del paso de descenso en el ejemplo anterior. No obstante, hay que llevar cuidado a la hora de establecer este valor, ya que si es demasiado grande puede desembocar en un escenario de no convergencia ni estabilidad. Por el contrario, si es muy pequeño, se necesitarían más iteraciones y podría quedarse atascado en un mínimo local donde la solución no fuera óptima.

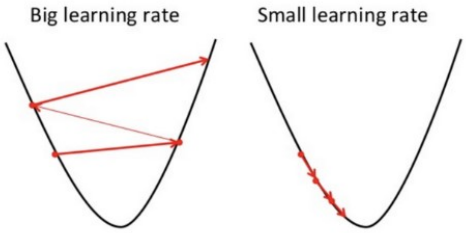


Figura 4.5: Comparación de dos tasas de aprendizaje (Imagen extraída de [10])

Además, para este método se selecciona en cada iteración un subconjunto de datos de entrenamiento al azar, conocido como **minibatch** o **minilotes**, tal y como se especificará en el Capítulo de Métodos y Materiales.

4.3.5. Algoritmo de Retropropagación

El Algoritmo de Retropropagación, también conocido como **Backpropagation**, se encarga de averiguar el gradiente de la función de coste para el ajuste de parámetros entre cada una de las conexiones, comenzando en la ultima capa de la red neuronal y avanzando hacia atrás. Se trata de la continuación del Algoritmo Forward (ver Subsección 4.3.3) ([51], [55], [52], [49]).

En este algoritmo (extraído de [55]), se va a utilizar la *Regla de la Cadena* para hallar las derivadas parciales del gradiente, tal y como se expresa a continuación:

$$\Delta_{W_L} C = \frac{\partial C}{\partial W_L} = \frac{\partial C}{\partial Z_L} \cdot \frac{\partial Z_L}{\partial A_L} \cdot \frac{\partial A_L}{\partial W_L} \tag{4.12}$$

y

$$\Delta_{B_L} C = \frac{\partial C}{\partial B_L} = \frac{\partial C}{\partial Z_L} \cdot \frac{\partial Z_L}{\partial A_L} \cdot \frac{\partial A_L}{\partial B_L} \tag{4.13}$$

para la última capa L, donde A_L y Z_L son dos vectores tales que: $A_l = (a_1^{(l)}, a_2^{(l)}, ..., a_{N_l}^{(l)})$ y $Z_l = (z_1^{(l)}, z_2^{(l)}, ..., z_{N_l}^{(l)})$ por la notación definida anteriormente.

La derivada $\frac{\partial C}{\partial Z_L}$ indica cómo cambia la función de coste en relación a la función de activación de la última capa (ver Derivada 4.8).

La derivada $\cdot \frac{\partial Z_L}{\partial A_L}$ de la función de activación quedó establecida en 4.6 para la última capa de la red y en 4.4 para otra capa diferente.

Por otro lado, $\frac{\partial A_L}{\partial W_L}$ y $\frac{\partial A_L}{\partial B_L}$ son derivadas directas de la Expresión 4.2 antes de aplicar la función de activación.

De este modo, los dos gradientes para la capa L podrían representarse como:

$$\Delta_{W_L} C = \frac{\partial C}{\partial W_L} = \frac{\partial C}{\partial Z_L} \cdot f'_L \cdot Z_{L-1} = \delta_L \cdot Z_{L-1} \tag{4.14}$$

y

$$\Delta_{B_L} C = \frac{\partial C}{\partial B_L} = \frac{\partial C}{\partial Z_L} \cdot f'_L = \delta_L \tag{4.15}$$

siendo δ_l el **error de capa**, que es el cambio que experimenta la función de coste con respecto a la variación de la suma ponderada ($\delta_l = \frac{\partial C}{\partial A_l}$) en una capa l.

Ahora, los gradientes en la penúltima capa serían:

$$\begin{aligned} \Delta_{W_{L-1}} C = \frac{\partial C}{\partial W_{L-1}} = \frac{\partial C}{\partial Z_L} \cdot \frac{\partial Z_L}{\partial A_L} \cdot \frac{\partial A_L}{\partial Z_{L-1}} \cdot \frac{\partial Z_{L-1}}{\partial A_{L-1}} \cdot \frac{\partial A_{L-1}}{\partial W_{L-1}} = \\ = \delta_L \cdot W_L \cdot f'_{L-1} \cdot Z_{L-2} = \delta_{L-1} \cdot Z_{L-2} \end{aligned} \tag{4.16}$$

y

$$\begin{aligned} \Delta_{B_{L-1}} C = \frac{\partial C}{\partial B_{L-1}} = \frac{\partial C}{\partial Z_L} \cdot \frac{\partial Z_L}{\partial A_L} \cdot \frac{\partial A_L}{\partial Z_{L-1}} \cdot \frac{\partial Z_{L-1}}{\partial A_{L-1}} \cdot \frac{\partial A_{L-1}}{\partial B_{L-1}} = \\ = \delta_L \cdot W_L \cdot f'_{L-1} = \delta_{L-1} \end{aligned} \quad , \tag{4.17}$$

teniendo en cuenta que $\frac{\partial A_L}{\partial Z_{L-1}} = W_L$.

Por último, reiterando este proceso hasta alcanzar la capa de entrada de la red, se obtiene:

$$\Delta_{W_1} C = \frac{\partial C}{\partial W_1} = \delta_1 \cdot X \tag{4.18}$$

y

$$\Delta_{B_1} C = \frac{\partial C}{\partial B_1} = \delta_1 \tag{4.19}$$

4.4. Matriz de confusión

Una matriz de confusión es una representación visual que permite evaluar el rendimiento de un modelo de clasificación en aprendizaje supervisado a la hora de predecir un conjunto de datos, proporcionando información acerca del número de aciertos y errores de cada una de las clases. Este mecanismo es útil para averiguar si el modelo está cometiendo errores de clasificación al realizar las predicciones ([59], [60], [43]).

La tabla 4.1 presenta la matriz de confusión utilizada para este estudio en los ocho grupos considerados (banjo, violonchelo, flauta, guitarra, caja de música, órgano, piano y trombón). Por tanto, se trata de una matriz multiclase de dimensión 8 x 8. Las filas van asociadas a las clases reales de los instrumentos, mientras que las columnas son las clases predichas por la red neuronal.

Tabla 4.1: Matriz de confusión para la clasificación de 8 clases

	Banjo Pred.	Cello Pred.	Flauta Pred.	Guitarra Pred.	Music Box Pred.	Órgano Pred.	Piano Pred.	Trombón Pred.
Banjo Real	TP_B	FN_B	FN_B	FN_B	FN_B	FN_B	FN_B	FN_B
Cello Real	FP_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B
Flauta Real	FP_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B
Guitarra Real	FP_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B
Music Box Real	FP_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B
Órgano Real	FP_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B
Piano Real	FP_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B
Trombón Real	FP_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B	TN_B

Los valores que componen la tabla se explican a continuación, considerando el banjo como la clase de interés en este caso:

- **TP** o *True Positive* (Verdadero Positivo): corresponde con la cantidad de clases que han sido identificadas de forma exitosa como pertenecientes a la clase de interés. En este ejemplo, dicho valor (TP_B) se refiere al número de espectrogramas de banjo que han sido predichos correctamente como "Banjo".

- **TN** o *True Negative* (Verdadero Negativo): se refiere al número de clases distintas a la de interés que han sido clasificadas correctamente como clases no pertenecientes a ella. Por ejemplo, si el espectrograma a clasificar se trata de un piano y la red consigue acertar en su predicción o incluso ponerle una etiqueta equivocada siempre y cuando no lo confunda con un espectrograma de banjo, entonces se contabilizaría como verdadero negativo (TN_B).
- **FP** o *False Positive* (Falso Positivo): representa el número de clases que han sido identificadas de manera errónea como la clase de interés. Se incluiría dentro de esta categoría (FP_B) la imagen de un espectrograma de órgano que la red ha etiquetado como "Banjo".
- **FN** o *False Negative* (Falso Negativo): va asociado al número de clases identificadas de manera errónea como clases distintas a la de interés. Se añadiría aquí (FN_B) la imagen de un espectrograma de banjo que ha sido etiquetado como "Trombón".

La Tabla 4.1 muestra únicamente el caso específico para el banjo como clase de interés. De esta forma, se puede extrapolar para el resto de instrumentos musicales. La siguiente tabla, 4.2, ilustra cómo se configuraría la matriz de confusión en su forma general para el número de clases analizadas.

Tabla 4.2: Forma General de la matriz de confusión 8x8

	Banjo Pred. (B)	Cello Pred. (C)	Flauta Pred. (F)	Guitarra Pred. (G)	Music Box Pred. (M)	Órgano Pred. (O)	Piano Pred. (P)	Trombón Pred. (T)
Banjo Real (B)	TP_B	E_{BC}	E_{BF}	E_{BG}	E_{BM}	E_{BO}	E_{BP}	E_{BT}
Cello Real (C)	E_{CB}	TP_C	E_{CF}	E_{CG}	E_{CM}	E_{CO}	E_{CP}	E_{CT}
Flauta Real (F)	E_{FB}	E_{FC}	TP_F	E_{FG}	E_{FM}	E_{FO}	E_{FP}	E_{FT}
Guitarra Real (G)	E_{GB}	E_{GC}	E_{GF}	TP_G	E_{GM}	E_{GO}	E_{GP}	E_{GT}
Music Box Real (M)	E_{MB}	E_{MC}	E_{MF}	E_{MG}	TP_M	E_{MO}	E_{MP}	E_{MT}
Órgano Real (O)	E_{OB}	E_{OC}	E_{OF}	E_{OG}	E_{OM}	TP_O	E_{OP}	E_{OT}
Piano Real (P)	E_{PB}	E_{PC}	E_{PF}	E_{PG}	E_{PM}	E_{PO}	TP_P	E_{PT}
Trombón Real (T)	E_{TB}	E_{TC}	E_{TF}	E_{TG}	E_{TM}	E_{TO}	E_{TP}	TP_T

Los valores distribuidos a lo largo de la diagonal principal corresponden al total de datos clasificados adecuadamente.

Nota 4.4.1. *El término de la forma E_{JI} representa el número de clases reales J que han sido predichas incorrectamente como pertenecientes a la clase I.*

Definición 4.4.1. *Se define el* ***dominio*** *D empleado para el presente estudio como:*

$$D = \{B, C, F, G, M, O, P, T\}$$

donde estas iniciales se utilizan como abreviaturas para representar los instrumentos de banjo (B), violonchelo (C), flauta (F), guitarra (G), caja de música (M), órgano (O), piano (P) y trombón (T).

4.4.1. Métricas de Rendimiento para la clasificación en ocho clases

En primer lugar, es fundamental tener constancia de varios conceptos estándar que se van a definir a continuación para evaluar el rendimiento de clasificación del modelo:

- **Exactitud** (*Accuracy*): es la proporción de muestras clasificadas correctamente respecto al número total de muestras evaluadas, denotada como N. Este valor indica en términos generales cuánto se aproxima el modelo con sus predicciones. Se calcula mediante la siguiente fórmula:

$$\mathbf{ACC} = \frac{\sum_{\forall I \in D} TP_I}{N} \tag{4.20}$$

  siendo I la letra inicial de cada uno de los instrumentos del dominio (ver Definición 4.4.1). El numerador está compuesto por la suma de los elementos de la diagonal principal de la matriz. Es decir:

$$TP_I = TP_B + TP_C + TP_F + TP_G + TP_M + TP_O + TP_P + TP_T$$

- **Precisión** (o *Valor Predictivo Positivo*): es la proporción de elementos verdaderos positivos de una determinada clase de interés que han sido reconocidos de forma correcta con respecto al número total de predicciones positivas. Su expresión es:

$$\mathbf{VPP} = \frac{TP_I}{TP_I + FP_I} \tag{4.21}$$

  donde FP_I es el número de falsos positivos del instrumento I mencionado (ver Tabla 4.1) Por tanto, se obtiene la siguiente relación para un I fijo:

$$FP_I = \sum_{\substack{\forall J \in D \\ I \neq J}} E_{JI}$$

con E_{JI} ya definido previamente (ver Nota 4.4.1) y siendo J la inicial de otro instrumento diferente al de interés.

- **Sensibilidad** (*Recall* o *Tasa de Verdaderos Positivos*): es la proporción del número de elementos positivos correctamente clasificados por el modelo con respecto al total de casos positivos reales (pertenecientes a esa categoría).

$$\textbf{TVP} = \frac{TP_I}{TP_I + FN_I} \tag{4.22}$$

donde FN_I es el número de falsos negativos del instrumento I mencionado (ver Tabla 4.1). Asimismo, se cumple que:

$$FN_I = \sum_{\substack{\forall J \in D \\ I \neq J}} E_{IJ}$$

siendo nuevamente I el instrumento de interés fijo.

- **Especificidad** (o *Tasa de Verdaderos Negativos*): es la proporción de elementos verdaderos negativos TN_I (ver Tabla 4.1) reconocidos correctamente como no pertenecientes a la clase de interés I en relación con el número total de casos negativos (que no forman parte de dicha clase). Permite evaluar si el modelo ha logrado diferenciar de manera efectiva las clases restantes. Su fórmula es:

$$\textbf{TVN} = \frac{TN_I}{TN_I + FP_I} \tag{4.23}$$

Del mismo modo, se logra la siguiente igualdad:

$$TN_I = \sum_{\substack{\forall J \in D \\ J \neq I}} TP_J + \sum_{\substack{\forall J,K \in D \\ J \neq K \neq I}} E_{JK}$$

siendo J y K dos clases diferentes del dominio definido y que no corresponden con la de interés.

- **F1 Score**: se trata de la combinación de la precisión y la sensibilidad en una única métrica:

$$\textbf{F1 Score} = 2 \cdot \frac{VPP \cdot TVP}{VPP + TVP} \tag{4.24}$$

4.5. Red Neuronal Convolucional

En términos generales, las Redes Neuronales Convolucionales (también conocidas como CNN) son una forma de Red Neuronal Artificial comúnmente utilizadas en el ámbito del aprendizaje supervisado para la clasificación de imágenes. Su diseño presenta múltiples capas ocultas ordenadas jerárquicamente; las capas iniciales se encargan de detectar características o formas simples y a medida que la información pasa a través de capas más profundas, estas se vuelven capaces de reconocer estructuras más complejas así como figuras o rostros.

Este tipo de redes establecen relaciones entre las variables de entrada, que corresponden a los píxeles de una imagen, siendo relevante el lugar que ocupan en la imagen. Además, toda CNN se compone principalmente de tres tipos de capas diferentes: la capa convolucional, la capa de reducción o *pooling* y la capa totalmente conectada o *fully connected*, que se explicarán a continuación ([55], [57], [49], [52], [8], [61], [62], [63], [64], [43]):

- **La Capa Convolucional:** La función principal de esta capa es identificar patrones en la imagen de entrada mediante la aplicación de una serie de filtros que consisten en operaciones matemáticas sencillas, y están representados por una matriz cuadrada conocida como *kernel*. Por tanto, se generarán así nuevas imágenes llamadas *mapas de características*, que contienen información sobre las esquinas, los bordes de la imagen o los cambios de contraste, entre otros. Posteriormente, estos mapas se pasan a otras capas con el fin de ir aprendiendo detalles más específicos de la imagen.

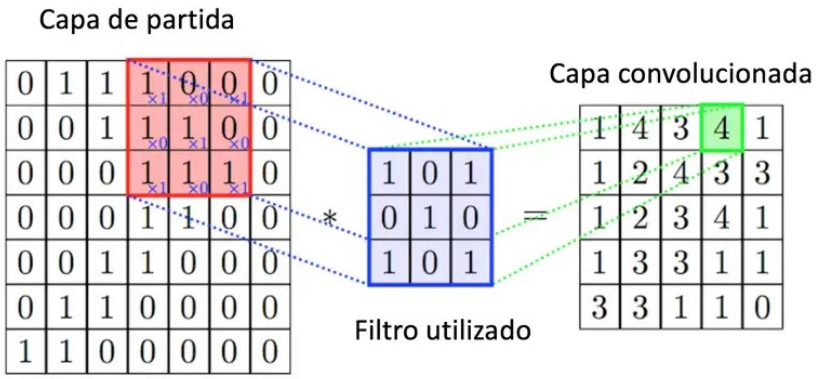


Figura 4.6: Operación de Convolución (Imagen extraída de [11])

El filtro utilizado comienza en la esquina superior izquierda de la imagen y va recorriendo los píxeles, avanzando de izquierda a derecha y de arriba a abajo. En cada cambio de posición se realiza una operación de convolución entre los elementos de la matriz y los valores de los píxeles de esa región, dando como resultado un único píxel. La agrupación de estos nuevos

valores obtenidos se recogen en una nueva matriz de tamaño más reducido, denominada *matriz de activación*, y muestra el mapa de características completo. En la Figura 4.6 se puede apreciar un ejemplo de convolución para comprender mejor este mecanismo.

- **La Capa de Reducción**: Esta capa disminuye la dimensionalidad de los mapas de características, rebajando la cantidad de parámetros de la red para así optimizar el tiempo de cómputo y reducir las posibilidades de sobreajuste. El proceso tiene lugar después de una operación de convolución y detecta los rasgos predominantes en cada mapa generado, evitando que la red relacione una ubicación específica con un determinado atributo.

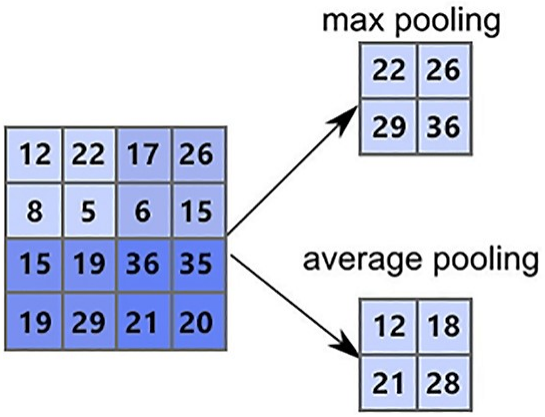


Figura 4.7: Operación de Reducción (Imagen extraída de [12])

Los dos principales filtros de reducción son **Max Pooling**, que selecciona el valor más alto de cada subdivisión de la matriz, y **Average Pooling**, encargado de calcular el promedio de los elementos presentes en dichas regiones. Los resultados se almacenan en una nueva matriz más simplificada. El procedimiento de desplazamiento del filtro es análogo al descrito en la Capa de Convolución.

- **La Capa Totalmente Conectada:** Esta capa establece las conexiones neuronales antes de la Capa de Salida de la red. Después del periodo de aprendizaje, se genera un conjunto de mapas de características que contienen detalles relevantes de la imagen de entrada y se utilizarán para la etapa final de clasificación. Para llevar a cabo este proceso, será necesario convertir las matrices multidimensionales de los mapas generados en un vector columna, que será introducido en la Capa Totalmente Conectada. Al final de esta capa, se devolverá un vector donde cada elemento representará la probabilidad de que la imagen de entrada pertenezca a una de las diferentes categorías existentes. Además, cada categoría se asocia a una neurona diferente.

En la Figura 4.8 se presenta un diagrama de las diversas capas implicadas en el proceso de clasificación de imágenes a partir de una CNN:

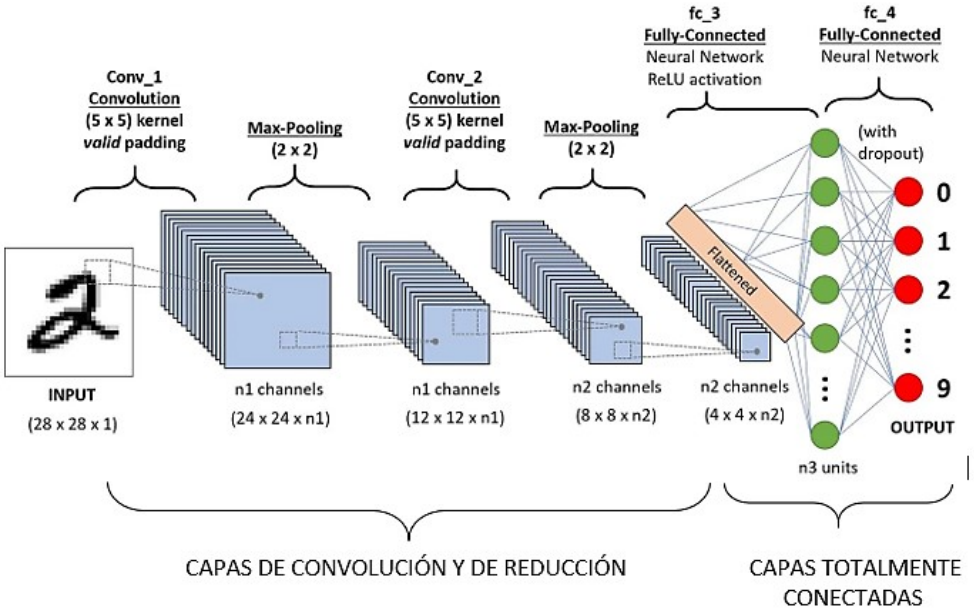


Figura 4.8: Esquema de las capas de una red neuronal convolucional (Imagen extraída de [13])

4.5.1. GoogLeNet

GoogLeNet es un tipo de Red Neuronal Convolucional que fue diseñada por Christian Szegedy y su equipo en 2014. Este modelo presenta una profundidad de 22 capas, de las cuales 9 son de una clase especial conocida como **Inception Modules**. El concepto de *Inception* se basa en la capacidad de expansión de la red sin aumentar su complejidad computacional de manera significativa. Por ello, cada módulo se encuentra formado por varias subcapas donde se utilizan distintos filtros de convolución de dimensiones como 1x1, 3x3 y 5x5, que operarán en paralelo y se concatenarán para formar una única salida (ver Figura 4.9). Así, considerando cada una de las capas de la red de forma individual, existen un total de 144 capas, tal como se visualiza en la Figura 5.14 extraída de Matlab.

Grosso modo, la red se puede dividir en tres partes distintas. La primera de ellas consiste en dos pares de capas de convolución seguidas de una capa de reducción *Max Pooling*. A continuación, en la segunda parte, se encuentra una secuencia de tres grupos de *Inception Layers*, que contienen dos, cinco y dos módulos respectivamente. Al final de cada uno de estos grupos, aparece nuevamente una capa de *Max Pooling*. En las capas de convolución e *Inception*, las neuronas utilizan la función de activación ReLU (ver Ecuación 4.3). Finalmente, en la última parte de la red, aparecen una serie de capas que realizan diferentes funciones. Primero,

hay una capa de reducción *Average Pooling* que reduce la dimensionalidad de los datos. Luego, hay una capa de *Dropout* que desactiva el 40 % de las neuronas de manera aleatoria para evitar el sobreajuste. Y después, existe una capa con una función de activación *SoftMax* (ver Ecuación 4.5) encargada de realizar la clasificación. Además, para la función de pérdida se emplea la Entropía Cruzada Categórica (ver Ecuación 4.7), que es común en este tipo de tareas ([65], [66], [63], [52], [8], [49], [67])

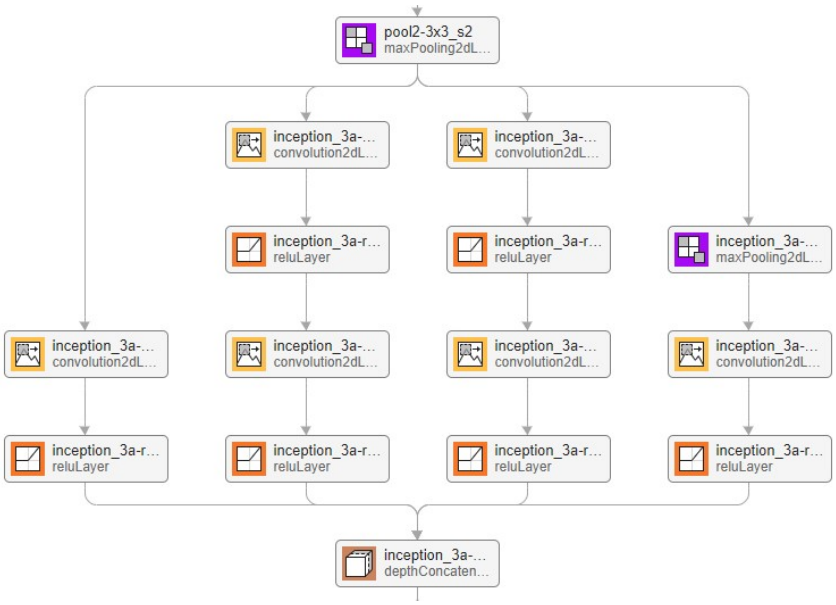


Figura 4.9: Estructura de un Módulo Inception de GoogLeNet (Imagen de creación propia a partir de Matlab)

La dimensión de las imágenes de entrada a la red GoogLeNet, que en este proyecto se han considerado espectrogramas de acordes musicales, es de 224 x 224 píxeles y tres canales de color (RGB). Como cada pixel es recogido por una neurona, habría por tanto un total de 150528 neuronas en la capa de entrada. Por otro lado, la capa de salida constará de ocho neuronas en total, correspondiendo cada una de ellas a un instrumento musical seleccionado ([43], [39]).

En el próximo capítulo de Métodos y Materiales se explicará con más detalle el proceso de modificación de las últimas capas de la red neuronal GoogLeNet para poder clasificar el conjunto específico de imágenes de la base de datos, que será redimensionado previamente.

5

Métodos y materiales

5.1. Dataset

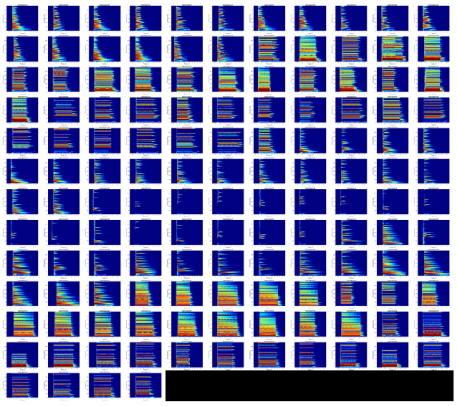
El conjunto de datos que ha sido empleado para la realización de este trabajo es de elaboración propia y se compone de 136 imágenes. Cada imagen representa un único espectrograma generado a partir de un archivo de audio específico. Estos archivos de audio contienen acordes musicales de ocho instrumentos diferentes, los cuales han sido grabados también expresamente para el presente estudio.

Las herramientas que se han utilizado para la grabación de sonidos son las siguientes:

- Teclado electrónico Yamaha YPT-230.
- Interfaz de audio: Behringer U-Phoria UMC204HD
- Estación de trabajo de audio digital: Reaper v.7.07 (software)

Gracias a las funciones del teclado electrónico o sintetizador, se ha permitido reproducir sonidos de acordes simulando los timbres del banjo, violonchelo, flauta, guitarra, music box (o caja de música), órgano, piano y trombón.

Es importante señalar que se ha tenido en cuenta el registro característico de cada instrumento musical, además de que no todos ellos son capaces formar un acorde. Por ejemplo, los instrumentos de viento, como el trombón a la flauta, no pueden tocar más de una nota al mismo tiempo debido a su naturaleza, así que vamos a suponer que existen al menos tres instrumentos idénticos sonando simultáneamente.



Conjunto total de espectrogramas 2D

Conjunto total de espectrogramas 3D

Figura 5.1: Imágenes de la base de datos (creación propia a partir de Matlab)

La Figura 5.1 proporciona la visualización del conjunto total de imágenes recogidas en la base de datos, las cuales van a ser posteriormente analizadas, primero en dos dimensiones y luego en tres dimensiones. Estas imágenes han sido organizadas de manera equitativa en ocho subcarpetas, lo que equivale a 17 espectrogramas por cada instrumento musical seleccionado, como se muestra a continuación:

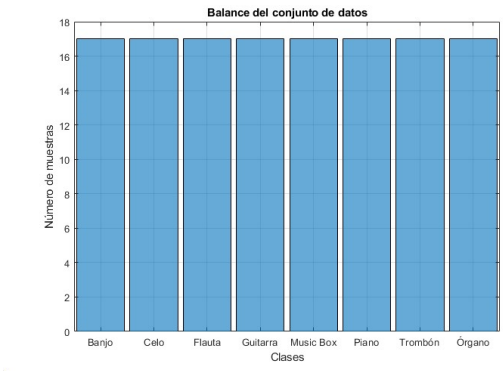


Figura 5.2: Balance del Dataset (Imagen de creación propia a partir de Matlab)

La resolución del problema se divide en dos fases:

- La transformación de archivos de audio en imágenes: espectrogramas y oscilogramas (Sección 5.2)
- El entrenamiento y validación de la red neuronal GoogleNet para desarrollar un modelo de reconocimiento y clasificación de imágenes según sus instrumentos musicales correspondientes (Sección 5.3)

El código se lleva a cabo a partir del lenguaje de programación Matlab.

5.2. Generación de espectrogramas y oscilogramas a partir de archivos de audio

Para la elaboración de esta primera parte del código se han tomado como referencia [15], [16], [68], [69], [17], [70] y el Proyecto de Frecuencia de Audio del Curso Online de Matlab [71].

5.2.1. Preparación de la señal de audio y elaboración del Oscilograma

Las señales de audio grabadas específicamente para este trabajo tienen una duración fija de 6 segundos y un tamaño de 24 bits por muestra. Además, se encuentran en formato *.wav* y son de tipo mono, tal y como se muestra en la Figura 5.3. Esto significa que se reproducen a partir de un único canal de audio. Si hubiera dos canales, uno para el oído izquierdo y otro para el derecho, se trataría de un sonido estéreo. De esta forma, se obtendría una matriz formada por dos columnas y para transformarla a una señal mono se realizaría el promedio entre ambas columnas

```
info = struct with fields:
             Filename: 'C:\Users\irene\Documents\
    CompressionMethod: 'Uncompressed'
          NumChannels: 1
           SampleRate: 44100
         TotalSamples: 264600
             Duration: 6
                Title: []
              Comment: []
               Artist: []
        BitsPerSample: 24
```

Figura 5.3: Información de un archivo de audio (Imagen de creación propia a partir de Matlab)

```
1     [x, fs] = audioread(archivo_audio)
```

En primer lugar, la función *audioread* va a leer los datos a partir de una ruta introducida y devuelve las siguientes dos variables:

- x: es un vector que contiene los datos muestreados de la señal de audio y están representados como valores de tipo double normalizados en un rango comprendido entre -1 y 1.

- *fs*: es un escalar positivo que representa la tasa de muestreo del conjunto de datos de audio. En otras palabras, este valor indica la frecuencia con la que se han tomado las muestras de la señal analógica para transformarla en una señal digital discreta. Su unidad se mide en Hercios (Hz) y especifica el número de muestras tomadas por segundo. En este caso concreto, la tasa de muestreo de cada archivo es de 44100 Hz.

Por tanto, multiplicando la tasa de muestreo por la duración de la señal de audio en segundos, se obtiene el número total de muestras, que en este estudio son 264600. Esta cantidad representa la longitud del vector x y es constante para todos los archivos grabados; se guarda en la variable n.

```
n = length(x)
```

A continuación, se define el vector t, que indica los instantes de tiempo regulares, medidos en segundos, en los que se registran cada una de las muestras de la señal.

```
t = (0:(n-1)) / fs
```

Representación gráfica de la señal de audio

Con los datos previamente definidos, el siguiente paso es elaborar un oscilograma, que es una representación gráfica de la amplitud de las muestras discretas de la señal en función de la variable temporal.

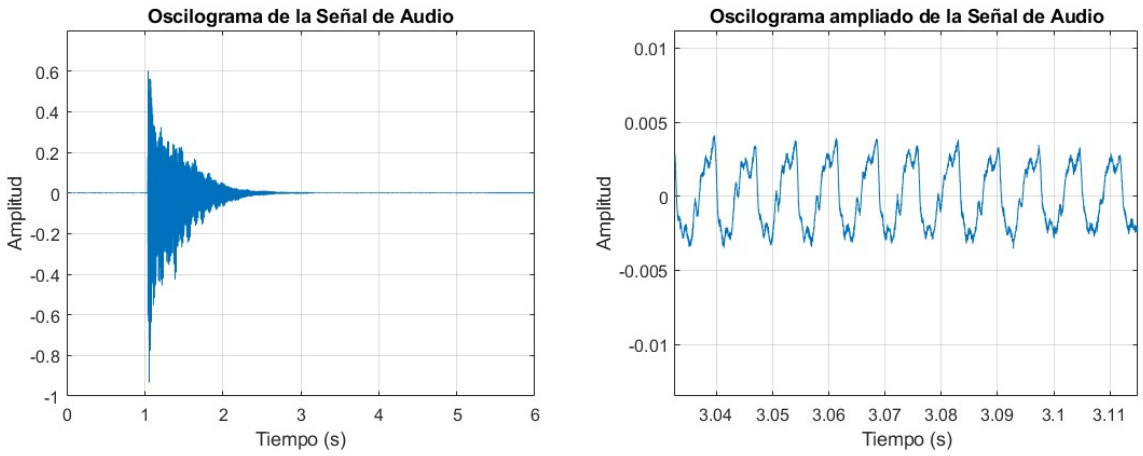


Figura 5.4: Oscilograma del Acorde $C\#_4$ de Banjo (Imágenes de creación propia a partir de Matlab)

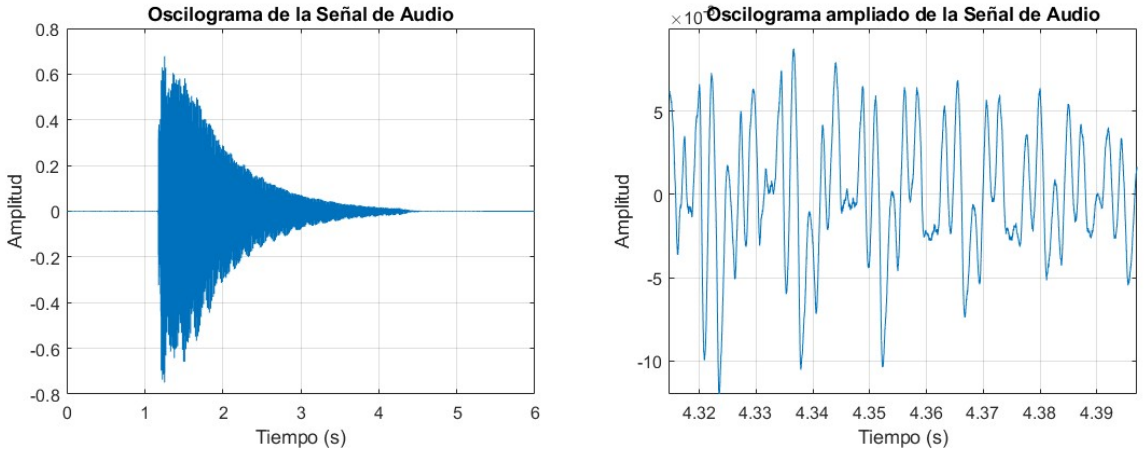


Figura 5.5: Oscilograma del Acorde $C\#_4$ de Piano (Imágenes de creación propia a partir de Matlab)

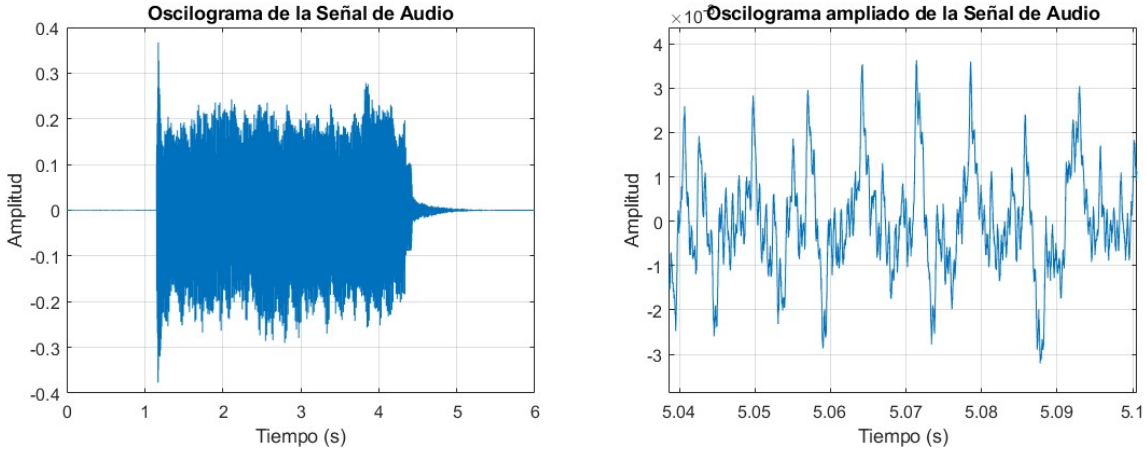


Figura 5.6: Oscilograma del Acorde $C\#_4$ de Órgano (Imágenes de creación propia a partir de Matlab)

En las imágenes de la derecha se visualiza una ampliación de los oscilogramas de un mismo acorde interpretado por tres instrumentos musicales: banjo, piano y órgano. Se puede distinguir una cierta periodicidad en la forma de las ondas, las cuales, influenciadas por las características tímbricas únicas de cada instrumento, presentan diferencias en su forma y amplitud, tal y como ya se desarrolló en la Sección 1.2.

Además, ninguno de ellos sigue el patrón de una única onda sinusoidal simple, sino que están compuestos por múltiples sinusoides con determinadas frecuencias correspondientes a las notas fundamentales que conforman el acorde central de $C\#_4$ y al resto de sus armónicos sucesivos.

Esto ha sido posible a partir de la siguiente función de Matlab:

```
1    plot(t, x);
```

5.2.2. Magnitud de la FFT (Transformada Rápida de Fourier)

A continuación, se va a aplicar la Transformada de Fourier a la señal para obtener información detallada sobre las frecuencias que componen el acorde. De esta forma, se podrán identificar cada una de las notas musicales presentes en él.

Como dicha señal se encuentra asociada al vector x formado por valores discretos, se utilizará la **Transformada Discreta de Fourier (DFT)** y el algoritmo empleado para su cálculo es conocido como **Transformada Rápida de Fourier (FFT)**. Por tanto, se procederá a calcular el valor absoluto de la salida de la función *fft* de Matlab, la cual devuelve números complejos, tal y como queda reflejado en el siguiente fragmento de código:

```
1     xfft=abs(fft(x))
```

El vector *xfft* generado contiene información sobre la magnitud de las diferentes frecuencias presentes en el archivo de audio. En otras palabras, una línea vertical más alta en la gráfica del espectro de frecuencia indica una mayor intensidad o fuerza de dicha frecuencia en la señal analizada.

El siguiente paso es construir el vector de frecuecias *f*, el cual estará compuesto por un total de n elementos, coincidiendo con el número de muestras de la señal almacenadas en la variable x. Estos valores estarán distribuidos uniformemente entre 0 y la frecuencia de muestreo *fs*. Además, la distancia entre dos frecuencias consecutivas del dominio corresponderá a la división de la frecuencia de muestreo entre el número total de elementos, es decir, $\frac{fs}{n}$. Así se consigue que las unidades de f queden expresadas en Hercios (Hz).

```
1     f=((0:(n-1))*fs)/n
```

Con toda esta información, ya se puede graficar la magnitud de la Transformada de Fourier en función de las frecuencias de la señal:

```
1     plot(f,xfft)
```

La Figura 5.7 representa el espectro de frecuencias del acorde Do central del piano (C_4). Al aplicar la Transformada de Fourier, se obtiene una simetría en torno a la mitad de la frecuencia de muestreo. Este valor coincide con el máximo valor posible de frecuencia que se puede alcanzar para recoger la señal correctamente, y se denomina *frecuencia de Nyquist* (ver Definición 5.2.1).

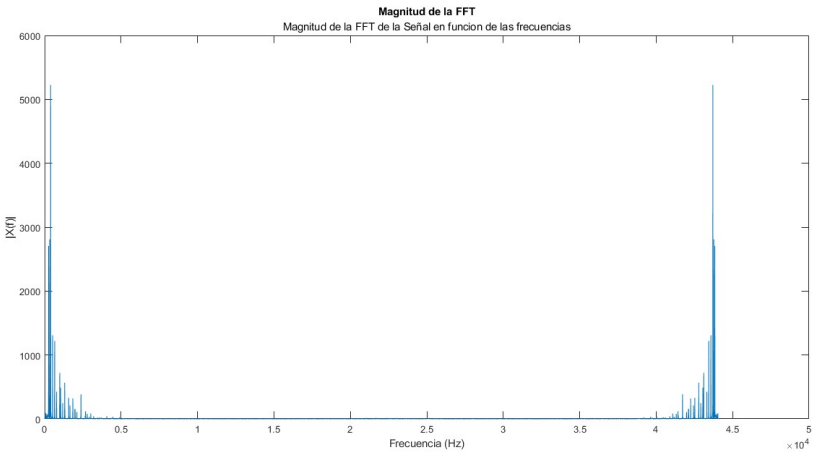


Figura 5.7: Magnitud de la FFT (Imagen de creación propia a partir de Matlab)

Definición 5.2.1. *El **Teorema de Muestreo**, también conocido como Teorema de Muestreo de Nyquist-Shannon, establece que para reconstruir correctamente una señal analógica y continua a partir de sus muestras discretas, la frecuencia de muestreo debe ser al menos el doble de la máxima frecuencia presente en la señal, la cual se denomina **frecuencia de Nyquist**. Es decir,*

$$f_s \geq 2 \cdot f_{máxima}$$

Por tanto, si no se verifica la anterior desigualdad, la digitalización de la señal podría perder información importante ([72], [35]).

Así, al seleccionar sólamente la primera mitad del espectro, se obtiene la información completa sobre el conjunto de frecuencias presentes en la señal.

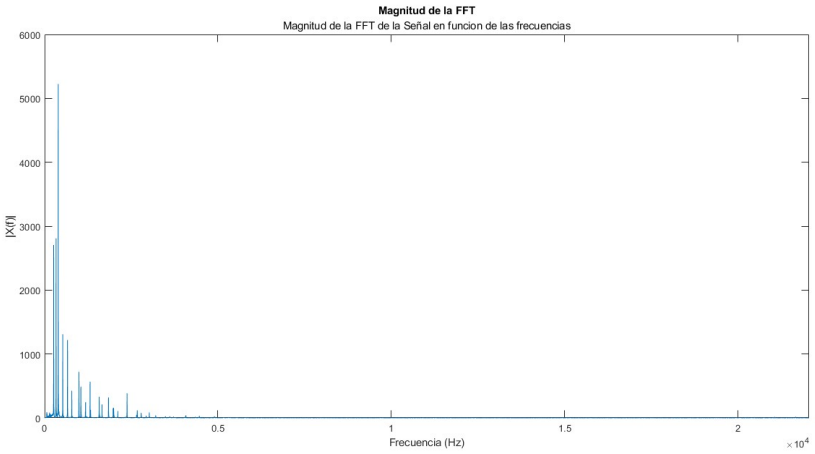


Figura 5.8: Espectro de frecuencias completo (Imagen de creación propia a partir de Matlab)

```
plot(f(1:n/2),xfft(1:n/2))
```

Para una mayor compresión de las frecuencias presentes en la gráfica, se abordará con más detalle en el próximo capítulo de resultados (ver 6.1).

Otra alternativa para hallar la Magnitud de la FFT

El algoritmo de la Transformada Rápida de Fourier (FFT) es más eficaz si su tamaño es una potencia de 2. Por lo tanto, se busca la potencia de 2 más cercana al número de muestras tomadas de la señal de audio. Este valor se redondea siempre hacia arriba, tal y como se muestra a continuación:

```
Nfft = 2^nextpow2(n)
```

Como el tamaño de la variable *Nfft* es superior a la longitud del vector x, se rellenarán con valores nulos las muestras faltantes en el vector x hasta que coincida con la dimensión especificada en *Nfft*. Las muestras adicionales no añaden información de frecuencias al archivo inicial, pero permiten un cálculo óptimo de la FFT. Luego, se aplica nuevamente la transformada de Fourier y se obtienen sus valores absolutos, que se almacenan en el vector X.

```
X = abs(fft(x,Nfft))
```

A continuación, se redefine el vector de frecuencias f comprendido entre 0 y fs, cuya longitud es ahora *Nfft*.

```
f = linspace(0,fs,Nfft)
```

Con estos datos, ya es posible representar la magnitud de la FFT en función de las frecuencias de la señal hasta alcanzar la Frecuencia de Nyquist, como se comentó anteriormente.

```
plot(f(1:Nfft/2), X(1:Nfft/2))
```

El resultado gráfico obtenido es el mismo que en la Figura 5.8 para un mismo archivo de audio.

5.2.3. Creación de un espectrograma 2D

En este apartado se pretende construir una representación visual de la señal de audio en dos dimensiones, que recoja información acerca de la variación de la amplitud y las frecuencias de los componentes sonoros a lo largo del tiempo. A este tipo de gráficas se les denomina **espectrogramas**, como ya se explicó en 3.2.3.

El código empleado para su elaboración es el siguiente:

Código 5.1: Código de creación de un espectrograma 2D ([15], [16])

```
[EspectX, Fespec, Tespec] = spectrogram(detrend(x), hann
    (2048), 1024, [], fs)
imagesc(Tespec, Fespec, log(abs(EspectX.^2)))
axis xy
set(gca, 'clim', [-1 1]*5, 'ylim', Fespec([1 dsearchn(
    Fespec, 3000)]), 'xlim', Tespec([1 end]))
xlabel('Tiempo (s)'), ylabel('Frecuencia (Hz)')
title('Espectrograma 2D');
colormap jet
```

Se emplea la función *spectrogram* de Matlab para calcular el espectrograma de una señal de audio en el dominio tiempo-frecuencia mediante la Transformada de Fourier de Tiempo Reducido (STFT). A continuación, se muestra una explicación más detallada de los distintos argumentos de esta función, en orden, que aparecen en el Código 5.1:

- *detrend(x)*: esta función elimina cualquier tendencia lineal presente en la señal x, evitando así el análisis de datos no sesgados.
- *hann(2048)*: Se selecciona una ventana de tipo Hann con un tamaño de 2048 muestras. Presenta una forma específica diseñada para atenuar los extremos de cada segmento de la señal, reduciendo así la presencia de lóbulos y fugas espectrales. Esto facilita una transición suave y continua entre segmentos adyacentes y mejora la precisión de la imagen [73].
- *Noverlap(1024)*: Los segmentos consecutivos de la señal tienen un solapamiento del 50 %, lo que significa que comparten la mitad de las muestras del tamaño de la ventana.
- *Nfft([])*: Recoge la cantidad de puntos que se van a emplear para el cálculo de la FFT en cada segmento. Al no especificarse, se proporciona un valor predeterminado, que equivale al máximo entre 256 y la siguiente potencia de 2 mayor que la longitud de cada segmento de la señal.
- *fs*: Corresponde a la Frecuencia de Muestreo, ya explicada con anterioridad.

La función devuelve dos vectores: uno de frecuencias, llamado *Fespec*, y otro de tiempo, denominado *Tespec*, además de una matriz, *EspectX*, compuesta por números complejos que proporcionan información sobre la magnitud. Además,

la longitud de *Fespec* y *Tespec* coinciden con el número de filas y columnas, respectivamente, de *EspectX*:

```
EspectX = 1025×257 complex
10^2 ×
   -0.0000 + 0.0000i  -0.0000 + 0.0000i ...
   -0.0000 + 0.0000i  -0.0000 + 0.0000i
    0.0001 - 0.0001i   0.0001 - 0.0000i
   -0.0001 + 0.0000i  -0.0000 + 0.0000i
   -0.0000 + 0.0000i  -0.0000 - 0.0000i
    0.0000 - 0.0000i  -0.0000 + 0.0000i
   -0.0000 + 0.0000i   0.0000 + 0.0000i
    0.0000 + 0.0000i  -0.0000 - 0.0000i
    0.0000 - 0.0000i   0.0000 + 0.0000i
   -0.0000 + 0.0000i  -0.0000 + 0.0000i
       ⋮
```

Figura 5.9: Matriz EspectX de la magnitud de la STFT (Imagen de creación propia a partir de Matlab)

La visualización del espectrograma se lleva a cabo mediante la función *imagesc* de Matlab, que representa los datos anteriores en una imagen con escala de colores. En el eje Y se distribuyen las frecuencias y en el eje X, el tiempo. La magnitud se calcula en una escala logarítmica, medida en decibelios, aplicando el valor absoluto del cuadrado de los elementos de la matriz *EspectX* (ver Código 5.1), asegurando así que solo se obtengan números reales. De esta forma, se logra una mejor eficacia de su representación en un rango tan amplio, donde los colores reflejan la intensidad o energía de las frecuencias a lo largo del tiempo. En este contexto, los colores más cálidos indican una mayor intensidad de la frecuencia.

La siguiente imagen muestra un ejemplo de espectrograma en dos dimensiones de un acorde de órgano:

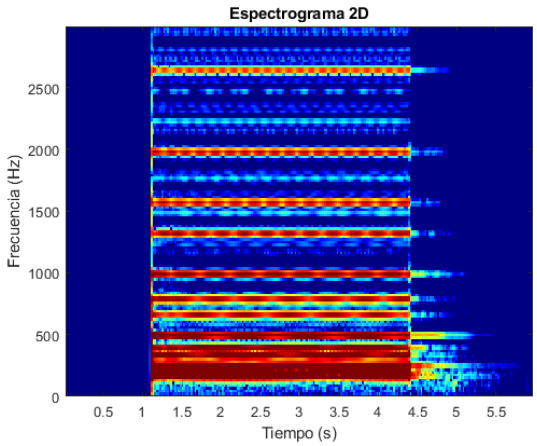


Figura 5.10: Espectrograma bidimensional del acorde E_3 de órgano (Imagen de creación propia a partir de Matlab)

En la Sección 6.2 se detalla una interpretación más exhaustiva de las frecuencias presentes en un espectrograma, y se compararán los diferentes espectrogramas para un mismo acorde de cada instrumento musical de la base de datos.

5.2.4. Creación de un espectrograma 3D

A partir de la función *mesh* de Matlab, se va a poder representar la información anterior en un espectrograma tridimensional, donde el eje X corresponde a la variable temporal, el eje Y se asocia a las frecuencias y el eje Z, a la magnitud en escala logarítmica.

Código 5.2: Código de creación de un espectrograma 3D ([17])

```
1    mesh(Tespec, Fespec, log(abs(EspectX.^2)))
2    xlabel('Tiempo (s)')
3    ylabel('Frecuencia (Hz)')
4    zlabel('Log Magnitud')
5    title('Espectrograma en 3D')
6    set(gca, 'clim', [-1 1]*5, 'xlim', Tespec([1 end]), 'ylim',
         Fespec([1 dsearchn(Fespec, 3000)]), 'zlim', [-1 1]*5, '
        FontName', 'Time New Roman', 'FontSize', 12);
7    colormap jet;
```

La figura 5.11 es un ejemplo de un espectrograma 3D para el mismo acorde de órgano que 5.10. Del mismo modo, se proporcionará una mayor información en la Sección 6.2 del próximo capítulo.

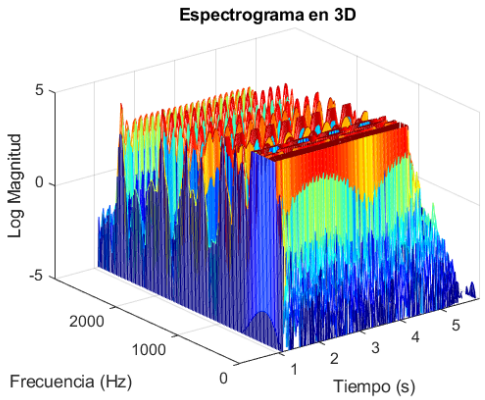


Figura 5.11: Espectrograma tridimensional del acorde E_3 de órgano (Imagen de creación propia a partir de Matlab)

Una vez se han creado todos los espectrogramas de cada archivo de audio, estos se almacenan en carpetas separadas por cada instrumento musical. Luego, se procederá a entrenar la red neuronal para su posterior clasificación, tanto de imágenes en 2D como en 3D, tal y como se explicará en la siguiente sección.

5.3. Transferencia de aprendizaje de una red neuronal convolucional preentrenada

Se partirá de una red neuronal preentrenada, como GoogLeNet, para realizar la clasificación del conjunto de imágenes mediante ciertas modificaciones que se detallarán a continuación ([18], [19], [74], [75], [58] y los cursos online de Matlab [76], [71], [77]).

5.3.1. Preparación de datos

Carga de datos y etiquetado

Es imprescindible realizar un buen etiquetado de datos, así que se va a asignar una etiqueta a cada espectrograma con el nombre de la subcarpeta correspondiente donde se encuentra almacenado. Por ejemplo, si se trata de una imagen que contiene el acorde de Fa mayor con el timbre de una flauta, la etiqueta que va a recibir será "Flauta".

Gracias a la fución *imageDatastore*, se genera un almacén de datos que contiene a las imágenes etiquetadas de manera automática. Esto se muestra en el siguiente fragmento de código, donde *pathToImages* es la ruta del directorio que contiene los espectrogramas de cada instrumento musical:

Código 5.3: Creación de un almacén de datos con imágenes etiquetadas ([18], [19])

```
imds = imageDatastore(pathToImages, 'IncludeSubfolders', true,
    'LabelSource', 'foldernames')
```

División de datos en conjuntos de entrenamiento, prueba y validación

- **Conjunto de entrenamiento ("Training")**: contiene el 80 % de los espectrogramas destinados a entrenar la red neuronal. Van modificando y optimizando los parámetros internos de la red, que son los pesos y los sesgos de cada conexión neuronal, permitiendo así un aprendizaje más eficiente.
- **Conjunto de validación ("Validation")**: abarca el 10 % de los espectrogramas que no han sido utilizados en el entrenamiento y permiten validar el rendimiento de la red neuronal, mejorando su eficacia mediante el ajuste de hiperparámetros, como la tasa de aprendizaje.
- **Conjunto de prueba ("Testing")**: se encuentra formado por el 10 % restante de los espectrogramas, que no han sido previamente analizados por la red neuronal, y servirán para comprobar la precisión del modelo una vez terminado el proceso de entrenamiento.

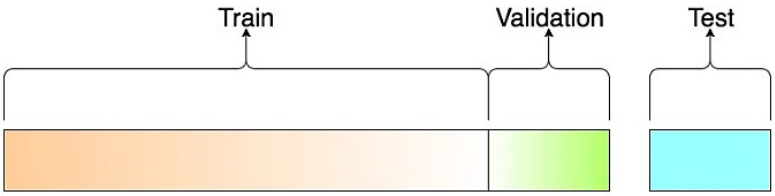


Figura 5.12: División de los tres conjuntos (Imagen extraída de [14])

Para llevar a cabo esta división, se empleará la función *randperm* para generar índices aleatorios y crear un vector de números enteros con las permutaciones de 136, que corresponde al número total de imágenes en el conjunto de datos.

A continuación, se procede a la creación de otros tres vectores adicionales a partir de este. El primero contiene 109 elementos y corresponde a los índices del conjunto de entrenamiento. El segundo está formado por 14 elementos, que representan los índices del conjunto de validación. Finalmente, el tercer vector está compuesto por 13 elementos, que son los índices que definirán el conjunto de prueba. Estas cantidades se ajustan a las proporciones mencionadas anteriormente (80-10-10 %).

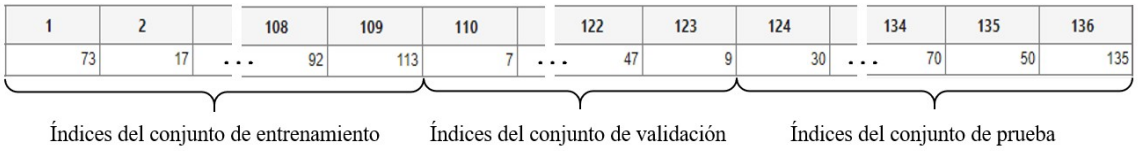


Figura 5.13: Representación gráfica de la obtención de los índices (Imagen de creación propia a partir de Matlab)

Los índices de estos vectores indicarán las posiciones exactas de las imágenes dentro del conjunto de datos original, denotado como *imds* en el siguiente código. Por tanto, nos permitirán seleccionar las imágenes necesarias para formar, los ya explicados, conjuntos de entrenamiento, prueba y validación, denotados como *trainds, testds* y *valds*, respectivamente.

Código 5.4: División de datos en tres conjuntos a partir de índices aleatorios ([20])

```
numImages = numel(imds.Files);
randIdx = randperm(numImages);
numTrain = round(0.8 * numImages);
numTest = round(0.1 * numImages);
trainIdx = randIdx(1:numTrain);
testIdx = randIdx(numTrain+1:numTrain+numTest);
valIdx = randIdx(numTrain+numTest+1:end);
trainds = subset(imds, trainIdx);
testds = subset(imds, testIdx);
valds = subset(imds, valIdx);
```

Ausencia de solapamiento

Para comprobar que no hay solapamiento entre los índices de cada conjunto se emplea la siguiente línea de código:

Código 5.5: Comprobación de la ausencia de superposición entre vectores ([20])

```
isequal(sort([trainIdx, testIdx, valIdx]), 1:numImages)
```

La función *isequal* compara la concatenación ordenada de los índices aleatorios de entrenamiento, prueba y validación con un vector comprendido entre el 1 y el número total de imágenes de la base de datos original. El resultado obtenido es un valor lógico de 1 ó 0, equivalente a *Verdadero* o *Falso*, respectivamente.

```
ans = logical
   1
```

Por lo tanto, se comprueba que la división de los datos se ha realizado correctamente, de manera que la combinación de los tres conjuntos obtenidos da lugar al conjunto de imágenes inicial y además, no se superponen entre ellos.

La aleatorización de los datos no solapados permite que la distribución de las imágenes sea diferente en cada iteración. Esta técnica contribuye a mitigar el riesgo de sobreajuste, evitando que el modelo memorice los datos de entrenamiento o aprenda patrones específicos, sesgados e irrelevantes que podrían conducir a una pérdida en la capacidad de generalización para nuevos datos. De esta forma, ayuda a mejorar el rendimiento y la eficacia del modelo, produciendo resultados más precisos.

5.3.2. Visualización de la arquitectura de GoogLeNet y ajuste de las imágenes de entrada

Figura 5.14: Visualización de la arquitectura de GoogLeNet (Imagen de creación propia a partir de Matlab)

```
net = googlenet;
analyzeNetwork(net)
```

La arquitectura de la red neuronal GoogLeNet está compuesta por 144 capas, como se muestra en la Figura 5.14. En su primera capa, conocida como la *Capa de Entrada* o *Input Layer*, se especifica que las imágenes deben tener un tamaño de 224 x 224 píxeles y tres canales de color (RGB) para poder ser procesadas. Esto queda reflejado en la Figura 5.15.

ANALYSIS RESULT

	Name	Type	Activations	Learnable Proper...
1	data 224×224×3 images with 'zerocenter' nor...	Image Input	224(S) × 224(S) × 3(C) × 1(B)	-
⋮	⋮	⋮	⋮	⋮
142	loss3-classifier 1000 fully connected layer	Fully Connected	1(S) × 1(S) × 1000(C) × 1(B)	Weigh... 1000 × 10... Bias 1000 × 1
143	prob softmax	Softmax	1(S) × 1(S) × 1000(C) × 1(B)	-
144	output crossentropyex with 'tench' and 999 othe...	Classification Output	1(S) × 1(S) × 1000(C) × 1(B)	-

Figura 5.15: Detalles de la primera y tres últimas capas de la red GoogLeNet (Imagen de creación propia a partir de Matlab)

El conjunto de datos empleado presenta imágenes con dimensiones rectangulares de 420 x 560 píxeles. Por lo tanto, se requiere normalizarlas al tamaño especificado en la Figura 5.15 antes de entrenar la red neuronal. Este proceso de redimensionamiento permite que las imágenes sean compatibles con la arquitectura de la red y se realiza de manera automática a partir del Código 5.7.

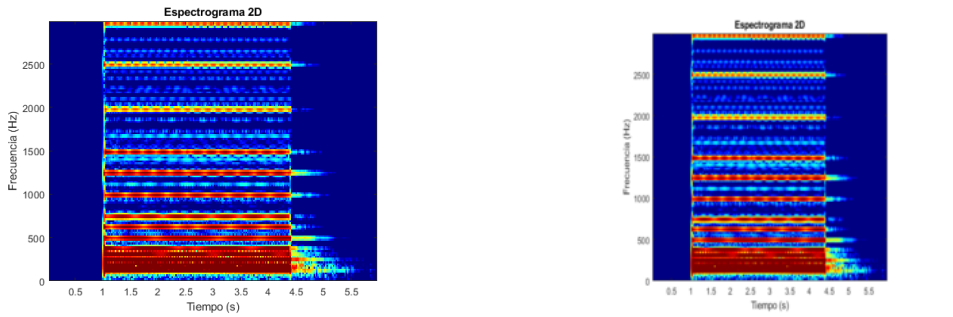


Imagen original **Imagen redimensionada**

Figura 5.16: Ejemplo del redimensionamiento de una imagen (Imágenes de creación propia a partir de Matlab)

En la Figura 5.16 se presenta la comparación visual de una imagen antes y después de redimensionarla.

5.3.3. Modificación de las últimas capas de GoogLeNet

142	Extractor de Características de Instrumentos 8 fully connected layer	Fully Connected	1(S) × 1(S) × 8(C) × 1(B)	Weights 8 × 1024 Bias 8 × 1
143	prob softmax	Softmax	1(S) × 1(S) × 8(C) × 1(B)	-
144	Clasificador de Instrumentos crossentropyex with 'Banjo' and 7 other classes	Classification Output	1(S) × 1(S) × 8(C) × 1(B)	-

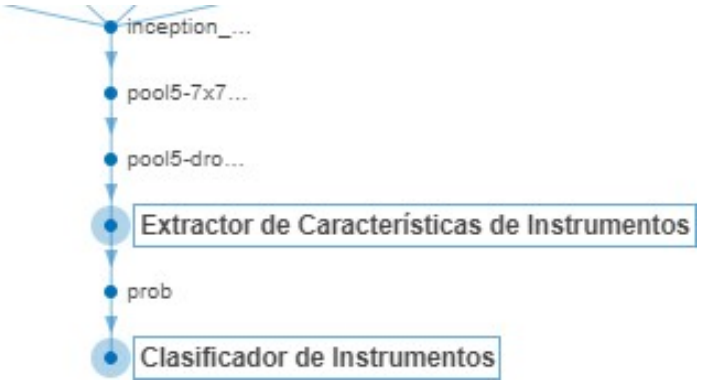


Figura 5.17: Detalles de las últimas capas de 'InstruNet' (Imágenes de creación propia a partir de Matlab).

La arquitectura de la red neuronal GoogLeNet está diseñada originalmente para clasificar 1000 tipos diferentes de objetos, como se detalla en la tabla 5.15. Sin embargo, dado que el conjunto de imágenes que se va a analizar se centra en la clasificación de instrumentos musicales en 8 clases distintas, es necesario reemplazar la capa número 144 con esta especificación. Se trata de la última capa de GoogLeNet, conocida como output de clasificación, y será sustituida por otra capa que se llamará **Clasificador de Instrumentos**, como se muestra en 5.17.

Por otro lado, la capa número 142, responsable de la extracción de características, también debe ser modificada para que pueda aprender los patrones relevantes presentes en cada espectrograma del conjunto de datos. Corresponde con la antepenúltima capa de GoogLeNet, denominada *loss3-classifier*, y se considera como una capa totalmente conectada o *Fully Connected*. Posteriormente, será renombrada como **Extractor de Características de Instrumentos**. En esta capa específica, se establecen los parámetros *WeightLearnRateFactor* y *BiasLearnRateFactor* para la transferencia de aprendizaje, asignándoles el valor de 10. Esto posibilita un aumento significativo en la velocidad de aprendizaje de los pesos y sesgos, siendo diez veces superior a la tasa de aprendizaje global utilizada en las demás capas.

A continuación, aparece el código que permite la sustitución de estas dos capas de la red.

Código 5.6: Modificación de las capas 142 y 144 de GoogLeNet ([18])

```
FeatureLearner = net.Layers(142)
OutputClassifier = net.Layers(144)
NumberOfClasses = numel(categories(trainds.Labels))
NewFeatureLearner = fullyConnectedLayer(NumberOfClasses, '
   Name', 'Extractor de Caracteristicas de Instrumentos', '
   WeightLearnRateFactor', 10, 'BiasLearnRateFactor', 10)
NewClassifierLayer = classificationLayer('Name', '
   Clasificador de Instrumentos')
LayerGraph = layerGraph(net)
NewLayerGraph = replaceLayer(LayerGraph, FeatureLearner.Name
   , NewFeatureLearner)
NewLayerGraph = replaceLayer(NewLayerGraph, OutputClassifier
   .Name, NewClassifierLayer)
```

5.3.4. Entrenamiento de la red GoogLeNet

Data Augmentation o aumento de datos

Una estrategia efectiva para mejorar la capacidad de generalización y prevenir el sobreajuste en modelos de aprendizaje automático es aumentar la cantidad de datos disponibles para el entrenamiento de la red neuronal.

Por lo tanto, en este estudio, se llevará a cabo un proceso de aumento de datos que implica generar nuevas imágenes mediante transformaciones aleatorias aplicadas al conjunto original de imágenes del dataset. Estas transformaciones van a incluir reflexiones en el eje X y desplazamientos horizontales y verticales dentro de un rango determinado de píxeles, cuya magnitud se definirá a continuación:

Código 5.7: Redimensionamiento de imágenes y aumento de datos ([19])

```
RangoPixel = [-30 30];
AumentoDatos = imageDataAugmenter( ...
'RandXReflection',true, ...
'RandXTranslation',RangoPixel, ...
'RandYTranslation',RangoPixel);
ResizedTrainingImage = augmentedImageDatastore([224 224 3],
    trainds, 'DataAugmentation',AumentoDatos)
ResizedValidationImage = augmentedImageDatastore([224 224
   3], valds, 'DataAugmentation',AumentoDatos)
ResizedTestingImage = augmentedImageDatastore([224 224 3],
   testds, 'DataAugmentation',AumentoDatos)
```

La función *augmentedImageDatastore* da lugar a un almacén de datos de imágenes aumentadas que, además de incrementar la diversidad y la cantidad de

muestras para el conjunto de entrenamiento, permite redimensionar las imágenes de entrada para cada conjunto, tal y como se describió en el apartado 5.3.2.

Esta herramienta suele resultar especialmente útil para conjuntos de datos más reducidos, ya que enriquece la variedad de muestras disponibles y mejora la capacidad de clasificación. Asimismo, también tiene un impacto positivo en conjuntos más grandes, dado que las imágenes modificadas no se guardan en la memoria.

Opciones de entrenamiento y ajuste de parámetros

Después de haber realizado los ajustes de modificación de la red neuronal, el siguiente paso es establecer las opciones para su entrenamiento mediante la función *trainingOptions*, donde se van a especificar los parámetros globales:

- El algoritmo matemático **'sgdm'** *(gradiente descendiente estoclástico con momento)* corresponde con el primer argumento de la función y es empleado para optimizar el modelo y reducir la función de pérdida durante el entrenamiento.
- **MiniBatchSize**: hace referencia al tamaño de los lotes o subconjuntos de datos utilizados para actualizar los parámetros en cada iteración durante el entrenamiento. Se establece en 5 el número de muestras por minilote, y en cada iteración se procesa uno de ellos. Así, la agrupación de todos los minilotes conforma una época.
- **MaxEpochs** o número de épocas: indica la cantidad de veces que se utiliza todo el conjunto de datos para entrenar a la red neuronal. En este caso, se han definido un total de 7 épocas.
- **InitialLearnRate**: determina el valor inicial de la tasa de aprendizaje global para el entrenamiento de la red, que se establece en 0,0001.
- **ValidationData**: se escoge el conjunto redimensionado de datos de validación para el entrenamiento de la red, según se definió en el Código 5.7.
- **ValidationFrequency**: corresponde al número de iteraciones de cada época, el cual coincide con el número de minilotes completos de tamaño 5. En este estudio, dado que el número de datos en el conjunto de entrenamiento es 109, al realizar la división, el valor resultante es de 21 iteraciones por época. Sin embargo, como la división no es exacta, algunos datos quedan excluidos en esta fase, pero no supone ningún problema gracias al ajuste de aleatorización de los datos que se explica a continuación.
- **Shuffle en 'every-epoch'** significa que los datos se organizan de manera aleatoria en cada época antes de ser divididos en minilotes. Esta configuración es útil para garantizar que todos los datos sean utilizados en el entrenamiento, incluso si el tamaño del conjunto de datos no es un múltiplo entero del tamaño del minilote.
- **Verbose** establecido en **false** evita la aparición de información detallada

sobre el progreso del entrenamiento mientras se ejecuta el código.

- **Plots** de **training-progress** realiza la representación gráfica de la evolución del entrenamiento. En la Figura 5.18 se muestra en la parte de arriba la función de precisión del modelo y, abajo, la función de perdida a lo largo de las diferentes épocas.

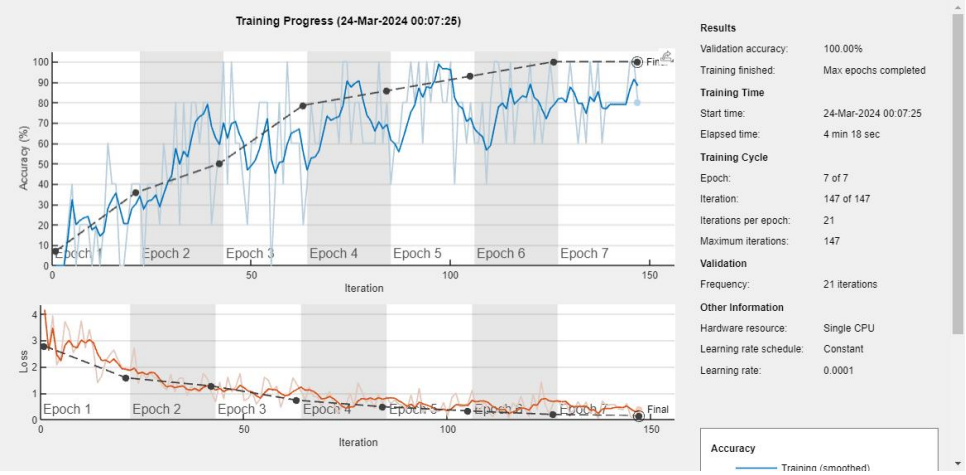


Figura 5.18: Progreso del entrenamiento (Imagen de creación propia a partir de Matlab)

El código empleado para ajustar las opciones de entrenamiento es el siguiente:

Código 5.8: Opciones de entrenamiento ([19])

```
opts = trainingOptions('sgdm', ...
  'MiniBatchSize', 5, 'MaxEpochs', 7, ...
  'InitialLearnRate', 1e-4, ...
  'Shuffle', 'every-epoch', ...
  'ValidationData', ResizedValidationImage, ...
  'ValidationFrequency', floor(numel(ResizedTrainingImage.
     Files)/5), ...
  'Verbose', false, ...
  'Plots', 'training-progress')
```

La función *trainNetwork* es la responsable de entrenar la red neuronal utilizando el conjunto redimensionado de imágenes de entrenamiento, la nueva arquitectura modificada y las opciones definidas anteriormente. Esta modificación de la red GoogLeNet recibe el nombre de InstruNet, que ha sido asignado de manera específica por la autora de este trabajo. Ahora ya se encuentra preparada para realizar el renocomiento y clasificación de espectrogramas a partir de los instrumentos musicales seleccionados. Este proceso se detallará en el próximo capítulo.

```
instrunet = trainNetwork(ResizedTrainingImage, NewLayerGraph
   , opts)
```

6

Resultados

6.1. Magnitud de la FFT

6.1.1. Interpretación del espectro de frecuencias

La Figura 6.1 representa la magnitud de la Transformada de Fourier (eje Y) en función de las frecuencias (eje X) para el acorde Do central (C_4) de un piano.

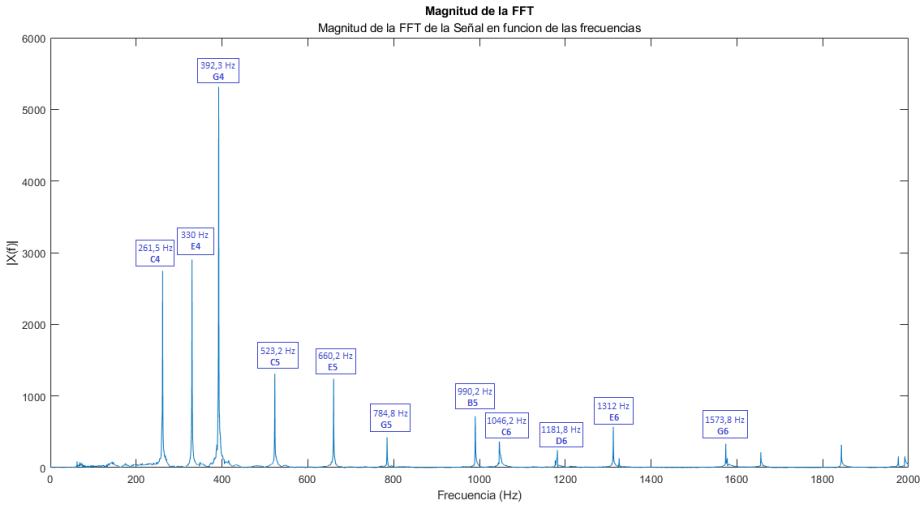


Figura 6.1: Espectro de frecuencias del acorde C_4 de piano (Imagen de creación propia a partir de Matlab)

Al ampliar la imagen, se aprecia que las tres primeras frecuencias presentan una mayor intensidad y se asocian con las notas fundamentales que conforman el acorde de Do Central de un piano ($C_4 - E_4 - G_4$). En la Tabla 6.1 aparecen representadas en color amarillo junto a sus respectivas frecuencias.

Tabla 6.1: Notas y frecuencias de los primeros armónicos de Do_4 (PARTE I)

NOTA	C_4	E_4	G_4	C_5	E_5	G_5
FRECUENCIA	261.6 Hz	329.6 Hz	392.0 Hz	523.3 Hz	659.3 Hz	784.0 Hz

Tabla 6.2: Notas y frecuencias de los primeros armónicos de Do_4 (PARTE II)

NOTA	B_5	C_6	D_6	E_6	G_6
FRECUENCIA	987.8 Hz	1046.5 Hz	1174.7 Hz	1318.5 Hz	1568.0 Hz

Además, se pueden distinguir también varios armónicos más agudos con una intensidad menor, tal y como aparecen reflejados en las Tablas 6.1 y 6.2:

- Acorde $C_5 - E_5 - G_5$: este conjunto de notas representa el primer trío de armónicos, ubicado una **octava justa** por encima del acorde fundamental.
- Acorde $G_5 - B_5 - D_6$: estos tonos forman el segundo trío de armónicos, dispuestos en un intervalo de **quinta justa** con respecto al acorde anterior. Es importante destacar que la nota G_5 se repite, siendo la octava de G_4 y la quinta de C_5.
- Acorde $C_6 - E_6 - G_6$: estas tres notas forman el tercer trío de armónicos, ubicado una cuarta más arriba que el acorde anterior, lo que equivale a un intervalo de **octava justa** con respecto al acorde fundamental.

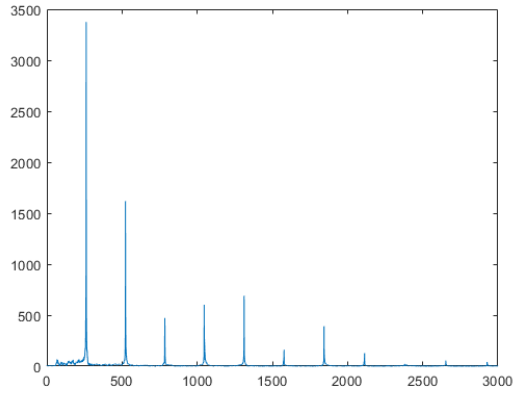
Cabe señalar que las frecuencias obtenidas en la Figura 6.1 son aproximaciones de los valores mostrados en las tablas, los cuales se han extraído de [78]. Además, la secuencia de tríos armónicos se distribuye con el mismo patrón de separación entre intervalos, tal y como se mostró en la Tabla 1.1 para una única nota.

6.1.2. Comparación del espectro de frecuencias en instrumentos diferentes

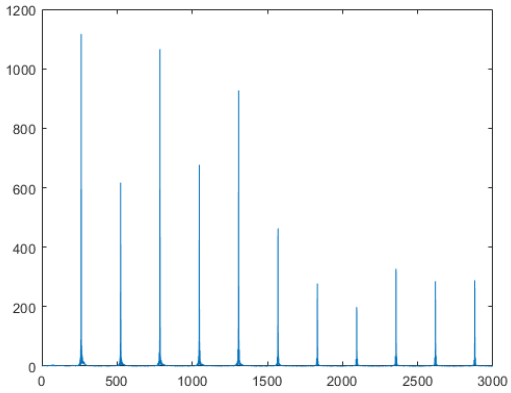
En este apartado, se realiza una comparación gráfica de las primeras frecuencias que componen una misma nota interpretada por dos instrumentos de diferente timbre: el piano y el trombón. La nota fundamental representada corresponde con el Do central de 261,6 Hz.

Al analizar la composición armónica de ambos sonidos, se observa que comparten las mismas frecuencias para sus armónicos, pero sus intensidades se distribuyen de manera diferente. Esta particularidad contribuye a la creación de perfiles

únicos en los espectros acústicos para cada instrumento musical, que podrían considerarse como una "huella dactilar armónica"que los caracteriza [21] y [36].



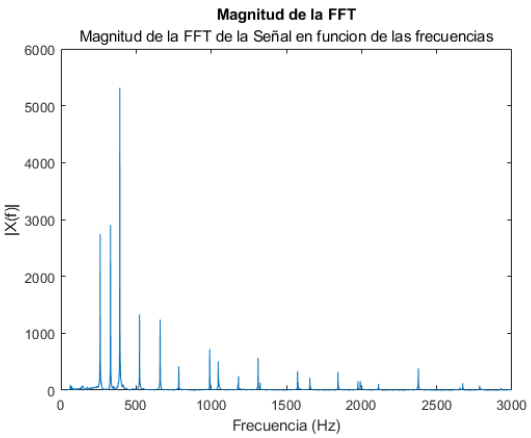
Nota C_4 de un piano



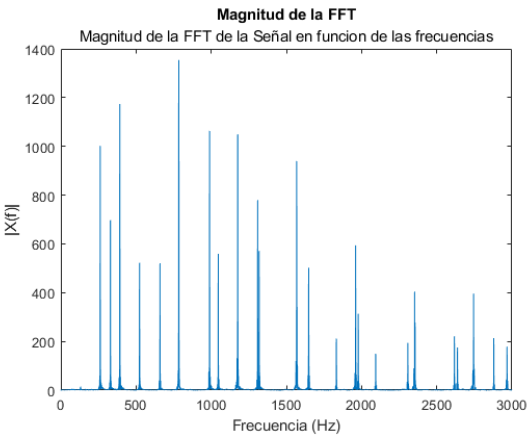
Nota C_4 de un trombón

Figura 6.2: Comparación entre la intensidad de los armónicos de la nota C_4 de dos instrumentos musicales (Imágenes de creación propia a partir de Matlab)

De manera análoga, continuando con el ejemplo del piano y del trombón, también se pueden apreciar las diferencias mencionadas anteriormente a partir del acorde C_4.



Acorde C_4 de un piano



Acorde C_4 de un trombón

Figura 6.3: Comparación entre la intensidad de los armónicos del acorde C_4 de dos instrumentos musicales (Imágenes de creación propia a partir de Matlab)

Se recuerda que los sonidos han sido grabados utilizando un sintetizador, por lo que la reproducción del acorde de trombón solo sería posible en un escenario real si se dispusiera de tres trombones, cada uno tocando una nota distinta del acorde de Do central (Do - Mi - Sol).

6.2. Análisis de espectrogramas 2D y 3D

En esta sección, se va a realizar un estudio de identificación de frecuencias en un espectrograma bidimensional y tridimensional. Posteriormente, se compararán gráficamente las imágenes generadas por cada instrumento musical para un mismo acorde y se harán algunas observaciones de los resultados obtenidos.

6.2.1. Interpretación de un espectrograma bidimensional

Primero de todo, en el apartado 5.2.3 de la Sección 5 se comentó el proceso de creación de un espectrograma y los distintos elementos que lo componen. Ahora, en la siguiente imagen se muestra un ejemplo de espectrograma 2D de piano para el mismo acorde de C_4 cuyas frecuencias fueron analizadas anteriormente (ver Figura 6.1).

Al seleccionar cualquier punto del espectrograma, se obtiene información sobre la frecuencia (eje Y), el instante de tiempo (eje X) y la intensidad medida a través de una escala de colores RGB, cuyo valor es numérico. A continuación, se han añadido etiquetas que contienen las notas de los primeros armónicos reconocidos y sus correspondientes frecuencias, tal y como se muestra en la Figura 6.4.

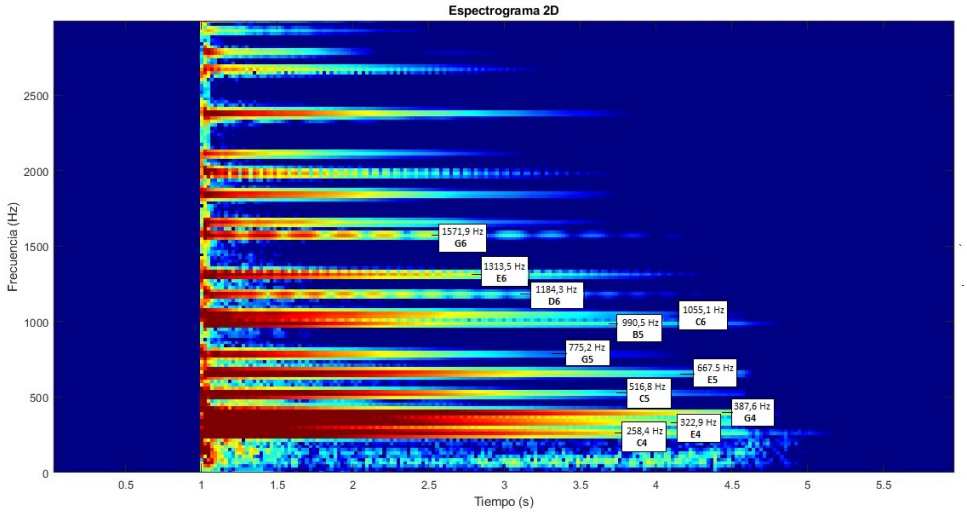


Figura 6.4: Interpretación del espectrograma bidimensional del acorde C_4 de piano (Imagen de creación propia a partir de Matlab)

Por tanto, es fácil darse cuenta de que las frecuencias de los primeros armónicos que componen el acorde analizado coinciden, de manera aproximada, con los valores establecidos en las tablas 6.1 y 6.2.

6.2.2. Interpretación de un espectrograma tridimensional

Siguiendo la misma línea de análisis, las anteriores frecuencias se pueden localizar también en un espectrograma tridimensional, añadiendo la componente Z con la información de la magnitud en escala logarítmica. Acto seguido, se presentan las imágenes correspondientes desde dos perspectivas diferentes, señalando los primeros armónicos de C_4:

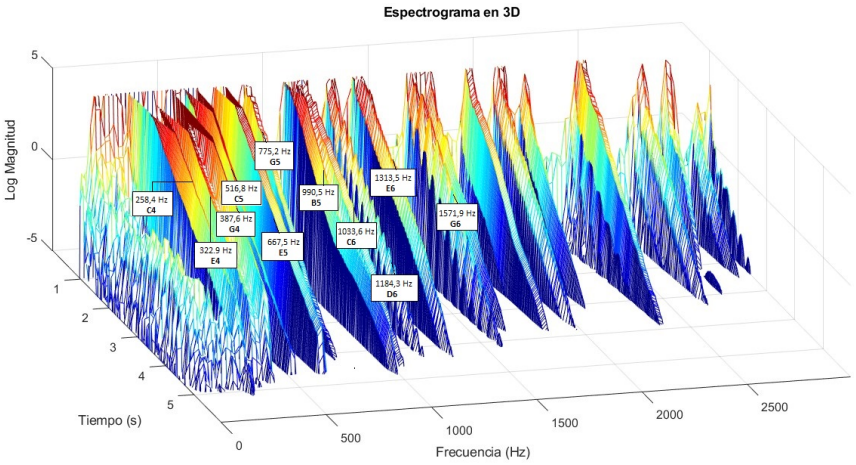


Figura 6.5: Interpretación del espectrograma tridimensional del acorde C_4 de piano I (Imagen de creación propia a partir de Matlab)

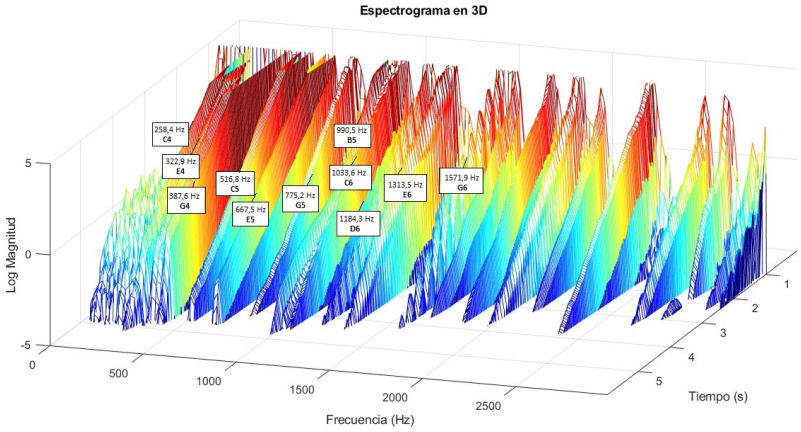


Figura 6.6: Interpretación del espectrograma tridimensional del acorde C_4 de piano II (Imagen de creación propia a partir de Matlab)

En los dos siguientes apartados, se representará el mismo acorde (C_4) reproducido por cada uno de los instrumentos presentes en la base de datos, utilizando espectrogramas tanto bidimensionales como tridimensionales. A partir de las diferencias existentes en las intensidades de sus frecuencias, se puede comprender de forma más clara el concepto de timbre, como ya se comentó en la Sección 1.2.

6.2.3. Comparación de espectrogramas bidimensionales

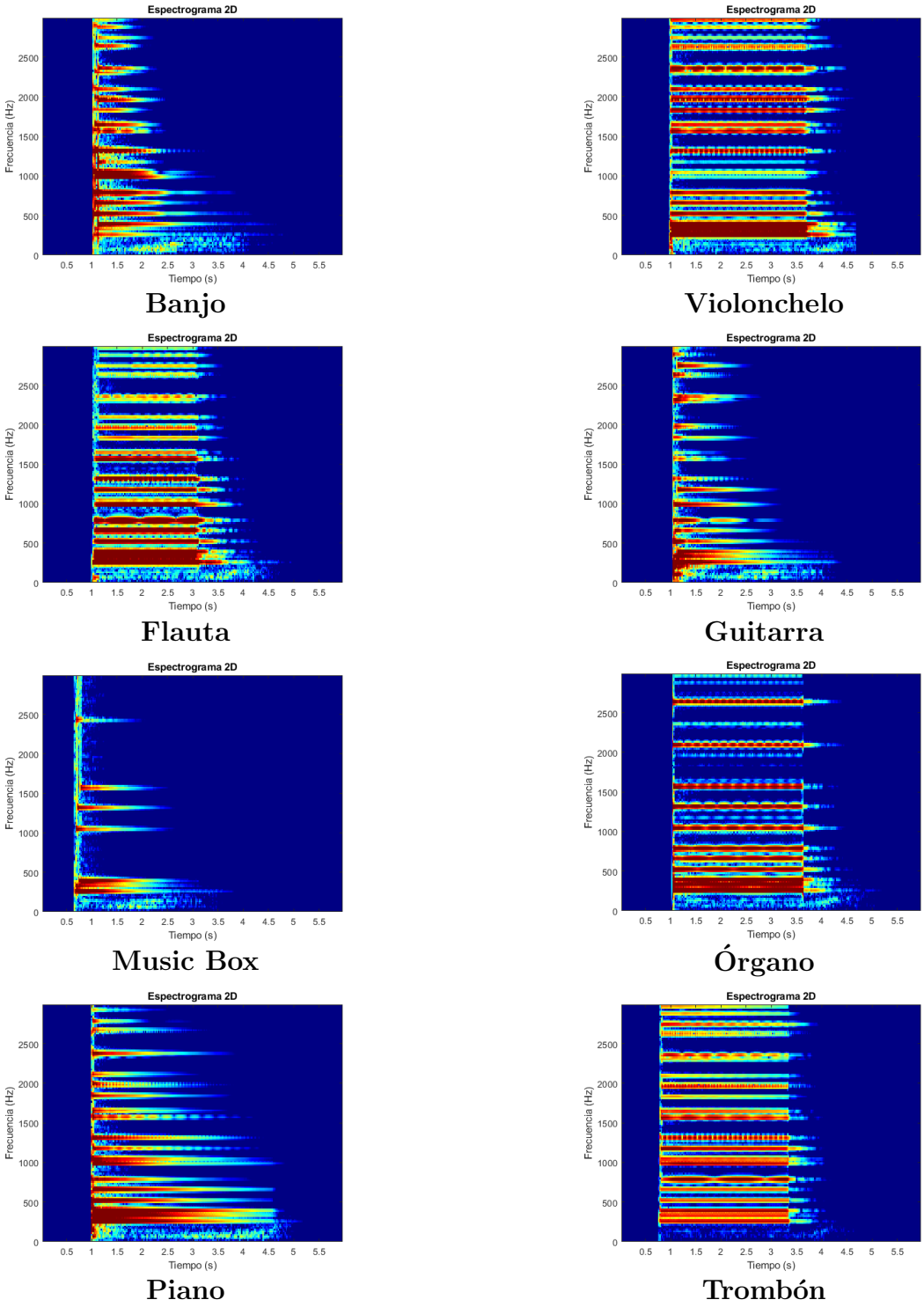


Figura 6.7: Comparación de espectrogramas 2D para el acorde C_4 central (Imágenes de creación propia a partir de Matlab)

6.2.4. Comparación de espectrogramas tridimensionales

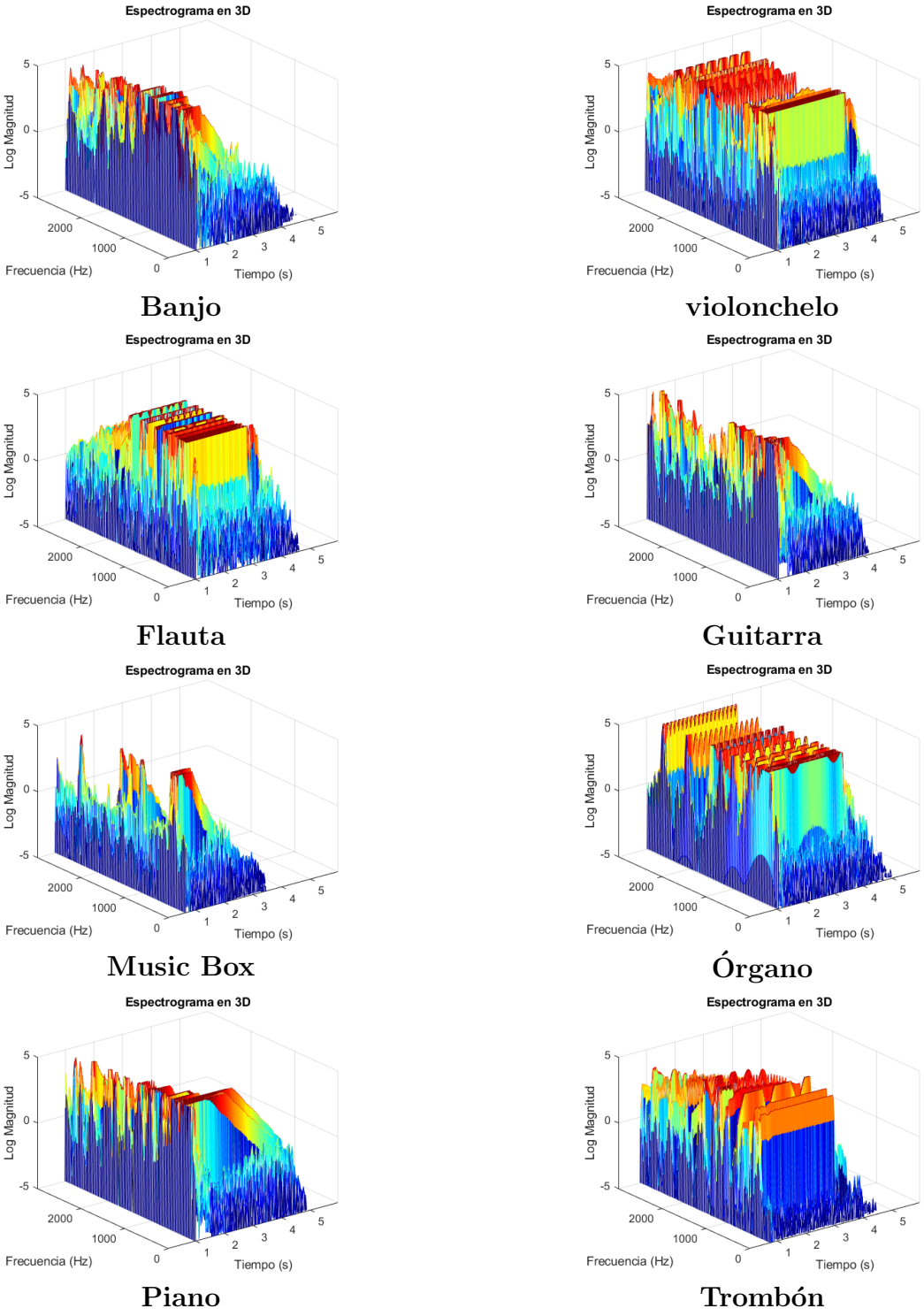


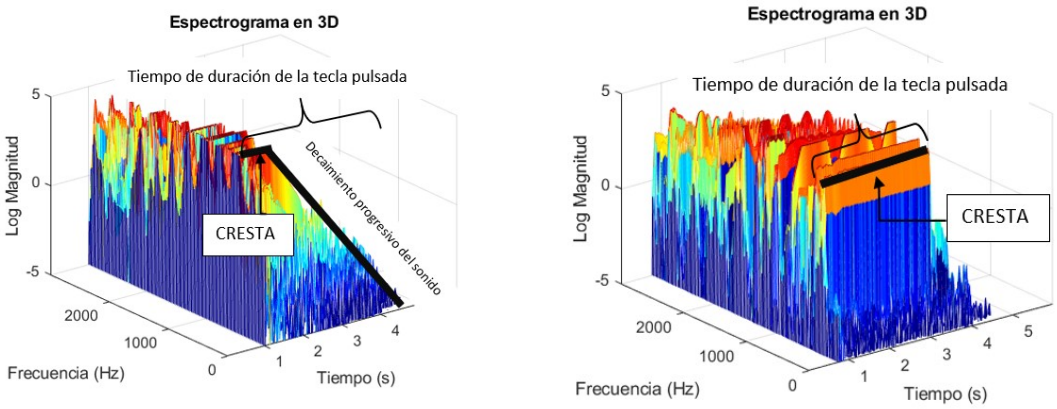
Figura 6.8: Comparación de espectrogramas 3D para el acorde C_4 central (Imágenes de creación propia a partir de Matlab)

6.2.5. Observaciones obtenidas tras la representación de espectrogramas

Los espectrogramas son una herramienta valiosa para comprender y analizar la estructura armónica y las características de un sonido de manera más detallada. Tras comparar las anteriores imágenes, se puede comprobar que la variación en la intensidad de las diferentes frecuencias que componen el acorde de Do Mayor es única en cada instrumento musical seleccionado, y su evolución en el tiempo difiere también entre ellos.

Por ejemplo, instrumentos como el órgano, el trombón o el violonchelo, emiten sonidos que se mantienen en el tiempo mientras las teclas del sintetizador permanecen pulsadas. En los espectrogramas 2D y 3D se aprecia durante varios segundos una franja horizontal o 'cresta' prolongada, respectivamente, por cada frecuencia presente en la señal y de un color más cálido e intenso en las notas fundamentales que componen el acorde.

Por otro lado, instrumentos como el piano, la guitarra o el banjo, generan un ataque inicial del sonido que se refleja como un pico máximo en las gráficas, seguido de una disminución gradual en su intensidad. Esto significa que el sonido se desvanece antes de que se levanten las teclas del sintetizador, y por lo tanto, la franja o cresta en el espectrograma es más reducida.



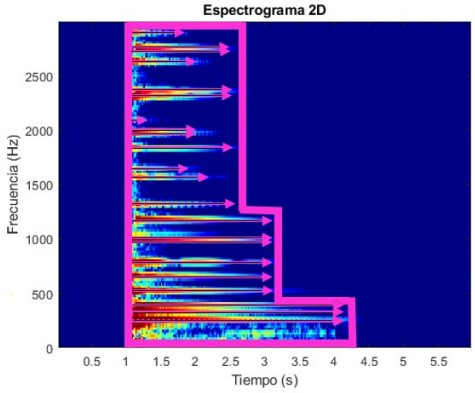
Acorde C_4 de un banjo

Acorde C_4 de un trombón

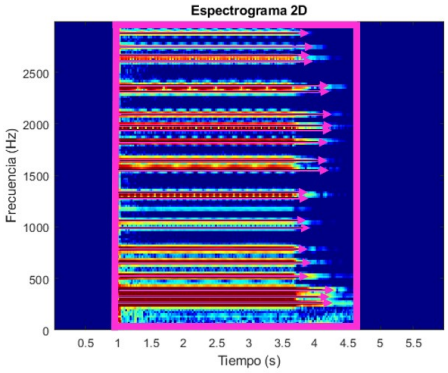
Figura 6.9: Comparación de la evolución del sonido a lo largo del tiempo de dos instrumentos musicales en un espectrograma 3D (Imágenes de creación propia a partir de Matlab)

Con respecto a la riqueza de armónicos en el conjunto de instrumentos estudiado, es evidente que el trombón y el violonchelo contribuyen con una amplia variedad de frecuencias que muestran niveles de volumen notables. Por el contrario, en otros instrumentos como la guitarra o la caja de música, la mayoría de sus armónicos poseen intensidades más bajas y su duración en el tiempo es más

corta, tal y como se puede visualizar en las siguientes imágenes:



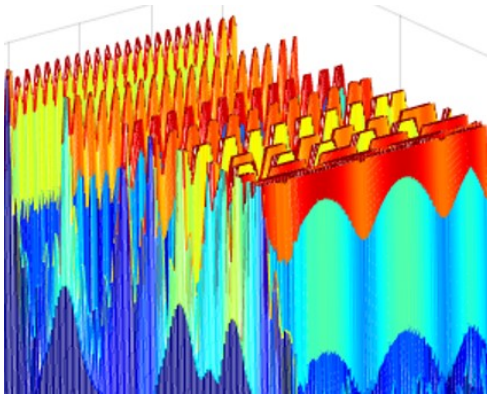
Acorde C_4 de una guitarra



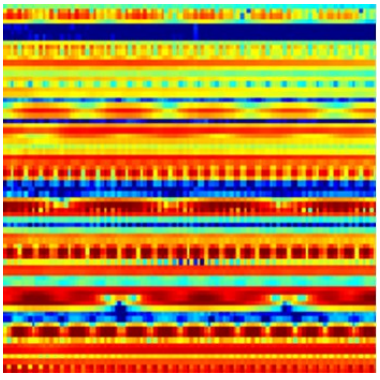
Acorde C_4 de un violonchelo

Figura 6.10: Comparación de la riqueza armónica de dos instrumentos musicales en un espectrograma 2D (Imágenes de creación propia a partir de Matlab)

El sintetizador tiene la particularidad de producir alteraciones periódicas en el volumen del sonido percibido para emular de manera naturalista algunos instrumentos musicales. Pretende recrear la variabilidad natural presente en la ejecución de instrumentos en vivo, añadiendo un elemento de realismo y expresividad a la reproducción sintetizada del sonido. Estas alteraciones se reflejan en el espectrograma como fluctuaciones regulares en la magnitud o intensidad. En los espectrogramas 3D se observan subidas y bajadas ondulatorias de la cresta en cada una de las frecuencias, mientras que en los espectrogramas 2D se pueden apreciar manchas de diferentes colores que se van alternando en el periodo de tiempo que dura el sonido. Se han tomado como ejemplos los acordes de Do mayor de órgano y trombón en octavas diferentes para ilustrar el fenómeno:



Acorde C_4 invertido de órgano (3D)



Acorde C_3 de trombón (2D)

Figura 6.11: Alteraciones periódicas de la magnitud visibles en el espectrograma (Imágenes de creación propia a partir de Matlab)

Sin embargo, esta oscilación automatizada puede afectar a la calidad percibida del sonido, ya que las oscilaciones son demasiado regulares para imitar fielmente la variabilidad natural de un instrumento tocado en vivo.

6.3. Clasificación de espectrogramas a partir de GoogLeNet

Una vez generados los espectrogramas bidimensionales y tridimensionales, se almacenan en carpetas separados por cada instrumento musical y se procede a entrenar la red neuronal. El propósito de este análisis es lograr la mayor precisión posible a la hora de asociar un conjunto de imágenes con su instrumento correspondiente.

Así pues, se han llevado a cabo diversos experimentos y se han registrado los resultados obtenidos de cada uno utilizando las métricas de rendimiento a partir de la matriz de confusión, cuya explicación se encuentra en la Sección 4.4.

6.3.1. Evaluación de espectrogramas con aumento de datos

Para comenzar, se recuerda que la base de datos cuenta con un total de 136 imágenes, de las cuales 14 forman parte del Conjunto de Validación y 13 del Conjunto de Prueba (o Test), representados en las tablas siguientes con las letras **V** y **T**, respectivamente. En este primer experimento, las imágenes del entrenamiento han sido sometidas a un procedimiento de *Aumento de Datos*, aplicando reflexiones y desplazamientos en los ejes X e Y (ver Código 5.7).

	PRECISIÓN		SENSIBILIDAD		ESPECIFICIDAD		F1 SCORE	
	V	T	V	T	V	T	V	T
Banjo	**100** %	**100** %	93,75 %	96,88 %	**100** %	**100** %	96,77 %	98,41 %
Cello	92,86 %	80,00 %	83,33 %	83,33 %	98,96 %	97,92 %	87,84 %	81,63 %
Flauta	85,00 %	57,14 %	93,75 %	**100** %	96,75 %	94,86 %	89,16 %	72,73 %
Guitarra	87,50 %	87,50 %	**100** %	**100** %	98,08 %	96,78 %	93,33 %	93,33 %
Music Box	**100** %	89,33 %	**100** %	93,33 %	**100** %	97,47 %	**100** %	91,29 %
Órgano	**100** %	**100** %	77,78 %	80,95 %	**100** %	**100** %	87,50 %	89,47 %
Piano	71,43 %	83,33 %	83,33 %	71,43 %	99,11 %	99,04 %	76,92 %	76,92 %
Trombón	64,29 %	92,86 %	71,43 %	78,57 %	96,07 %	98,96 %	67,67 %	85,12 %

Tabla 6.3: Métricas de clasificación de espectrogramas 2D con aumento de datos

La Tabla 6.3 muestra los porcentajes de precisión, sensibilidad, especificidad y F1 Score tras la clasificación de espectrogramas bidimensionales para los ocho instrumentos de la base de datos. Por tanto, se ha ejecutado el modelo ocho veces

consecutivas, primero para el conjunto bidimensional y luego para el tridimensional, y se ha calculado la media aritmética de cada valor utilizando Excel. Además, la exactitud global (accuracy) del modelo alcanza un **90,18 %** para el conjunto de Validación y un **86,54 %** para el conjunto de Prueba.

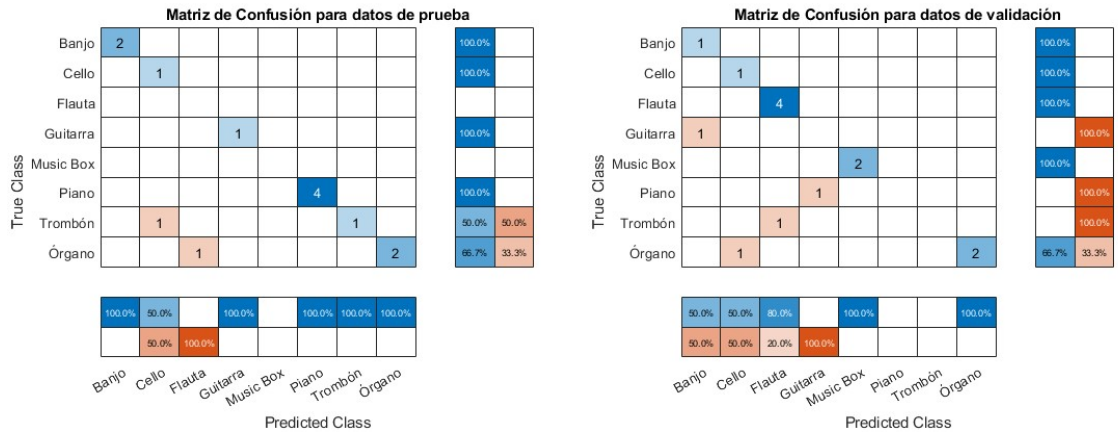
De forma similar, se han extraído las métricas de evaluación para espectrogramas tridimensionales con los mismos ajustes, como se refleja en la Tabla 6.4:

	PRECISIÓN		SENSIBILIDAD		ESPECIFICIDAD		F1 SCORE	
	V	T	V	T	V	T	V	T
Banjo	90,63 %	91,67 %	83,75 %	**100 %**	97,90 %	98,96 %	87,05 %	95,65 %
Cello	64,58 %	43,33 %	94,44 %	60,00 %	96,21 %	92,77 %	76,71 %	50,32 %
Flauta	97,14 %	81,25 %	87,50 %	**100 %**	98,75 %	98,00 %	92,07 %	89,66 %
Guitarra	58,33 %	78,57 %	81,25 %	68,75 %	93,03 %	96,69 %	67,91 %	73,33 %
Music Box	**100 %**	60,00 %	91,67 %	**100 %**	**100 %**	98,08 %	95,65 %	75,00 %
Órgano	**100 %**	**100 %**	93,33 %	66,67 %	**100 %**	**100 %**	96,55 %	80,00 %
Piano	88,89 %	89,29 %	78,57 %	72,92 %	97,92 %	95,97 %	83,41 %	80,28 %
Trombón	87,50 %	87,50 %	60,00 %	85,71 %	99,04 %	99,04 %	71,19 %	86,60 %

Tabla 6.4: Métricas de clasificación de espectrogramas 3D con aumento de datos

La exactitud es ligeramente inferior, con un **84,82 %** y **81,73 %** de clases correctamente clasificadas en los conjuntos de validación y prueba, respectivamente.

La Figura 6.12 muestra un ejemplo de matrices 8x8 generadas en una determinada iteración para los datos de prueba y validación.



2ª Iteración del Conjunto Test (2D) 5ª Iteración del Conjunto Validación (3D)

Figura 6.12: Experimento 1: Matrices de confusión 8x8 para espectrogramas con aumento de datos (Imágenes de creación propia a partir de Matlab)

Debido al número tan limitado de datos en la base, es posible que en una única iteración no se incluyan imágenes de alguna clase, como es el caso del Music Box en la matriz de la izquierda. Por tanto, para este estudio va a ser crucial

realizar múltiples iteraciones del modelo para obtener resultados más acertados. Los niveles de precisión en el reconocimiento del cello y de la guitarra en los espectrogramas 3D no son muy elevados, al igual que ocurre con el trombón o el piano en 2D. Por tanto, se van a llevar a cabo nuevos experimentos con el fin de mejorar y refinar este modelo.

6.3.2. Evaluación de espectrogramas sin aumento de datos

El segundo experimento consiste en entrenar la red neuronal empleando los mismos espectrogramas pero sin aplicar el aumento de datos. En este caso, se han llevado a cabo nueve iteraciones o ejecuciones del modelo y a continuación se presentan los resultados obtenidos:

	PRECISIÓN		SENSIBILIDAD		ESPECIFICIDAD		F1 SCORE	
	V	T	V	T	V	T	V	T
Banjo	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**
Cello	74,07 %	78,57 %	95,83 %	**100 %**	96,57 %	96,92 %	83,56 %	88,00 %
Flauta	95,83 %	91,67 %	90,63 %	89,58 %	99,07 %	97,88 %	93,16 %	90,61 %
Guitarra	87,50 %	**100 %**	**100 %**	**100 %**	99,21 %	**100 %**	93,33 %	**100 %**
Music Box	94,44 %	95,24 %	**100 %**	95,24 %	99,07 %	98,99 %	97,14 %	95,24 %
Órgano	**100 %**	96,43 %	84,38 %	**100 %**	**100 %**	98,89 %	91,53 %	98,18 %
Piano	**100 %**	96,88 %	93,52 %	88,89 %	**100 %**	98,89 %	96,65 %	92,71 %
Trombón	79,17 %	**100 %**	85,71 %	68,52 %	97,35 %	**100 %**	82,31 %	81,32 %

Tabla 6.5: Métricas de clasificación de espectrogramas 2D sin aumento de datos

La exactitud del conjunto de validación de espectrogramas 2D es **92,06 %** y del conjunto de prueba, **93,16 %**.

	PRECISIÓN		SENSIBILIDAD		ESPECIFICIDAD		F1 SCORE	
	V	T	V	T	V	T	V	T
Banjo	**100 %**	96,30 %	**100 %**	**100 %**	**100 %**	98,99 %	**100 %**	98,11 %
Cello	80,95 %	95,83 %	72,86 %	89,58 %	97,49 %	98,99 %	76,69 %	92,60 %
Flauta	71,43 %	85,71 %	85,71 %	**100 %**	96,70 %	98,15 %	77,92 %	92,31 %
Guitarra	**100 %**	**100 %**	89,58 %	81,25 %	**100 %**	**100 %**	94,51 %	89,66 %
Music Box	85,71 %	**100 %**	**100 %**	95,83 %	99,21 %	**100 %**	92,31 %	97,87 %
Órgano	95,83 %	**100 %**	**100 %**	**100 %**	99,07 %	**100 %**	97,87 %	**100 %**
Piano	89,29 %	92,71 %	95,24 %	**100 %**	98,14 %	97,88 %	92,17 %	96,22 %
Trombón	94,44 %	**100 %**	86,30 %	**100 %**	99,15 %	**100 %**	90,19 %	**100 %**

Tabla 6.6: Métricas de clasificación de espectrogramas 3D sin aumento de datos

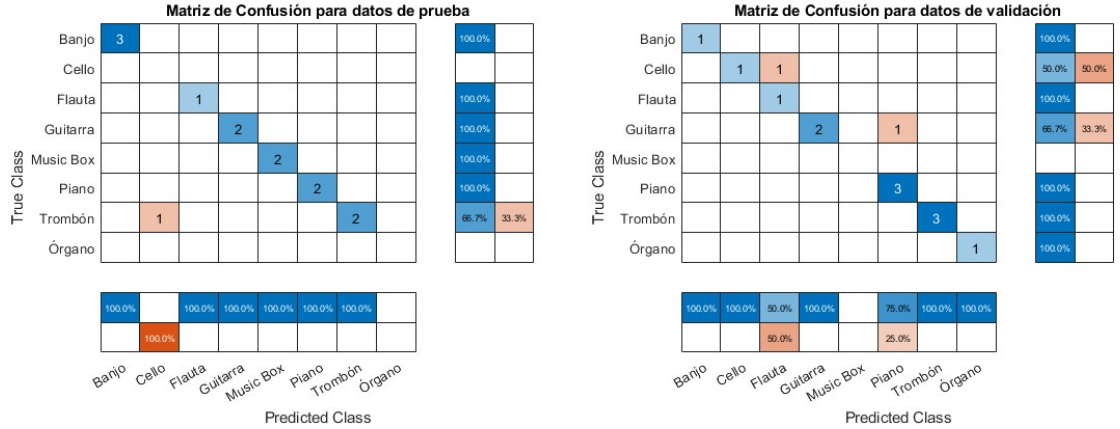
El conjunto de validación de espectrogramas 3D presenta una exactitud del **90,48 %**, mientras que el conjunto de prueba alcanza un **94,87 %**.

El análisis indica una mejora notable en este segundo experimento. Los espectrogramas, al ser representaciones visuales de datos de audio, presentan una

estructura muy concreta y específica, determinada por cada uno de los ejes X e Y, y también Z en el caso tridimensional. Por consiguiente, cualquier transformación aplicada, ya sea un giro o un desplazamiento, puede provocar una distorsión significativa de la información. Esta situación afecta negativamente a la capacidad del modelo para capturar las características relevantes, dificultando además la identificación precisa del instrumento.

Además, cabe destacar que, en condiciones reales, los espectrogramas de sonidos musicales no se encuentran generalmente invertidos ni volteados, sino que respetan su posición estándar previamente explicada. Por todos estos motivos, se concluye que el aumento de datos no es una herramienta beneficiosa para analizar un conjunto de espectrogramas.

Sin embargo, si se tomase como ejemplo una red neuronal entrenada para clasificar imágenes de instrumentos musicales reales, el enfoque del aumento de datos en este caso podría resultar útil debido a que estas imágenes pueden presentarse desde diferentes ángulos y perspectivas.



9ª Iteración del Conjunto Test (2D) 2ª Iteración del Conjunto Validación (3D)

Figura 6.13: Experimento 2: Matrices de confusión 8x8 para espectrogramas sin aumento de datos (Imágenes de creación propia a partir de Matlab)

La Figura 6.13 muestra dos ejemplos de iteraciones para la clasificación de imágenes sin la aplicación del aumento de datos. En la primera matriz, se puede apreciar que el modelo ha confundido un espectrograma real de trombón con uno de cello; y en la segunda, ha clasificado incorrectamente un espectrograma de cello como flauta y uno de guitarra como piano.

En general, tras analizar en profundidad todas las iteraciones generadas, se han observado errores frecuentes de clasificación entre el trombón, el cello, la flauta y el órgano, por un lado, y entre la guitarra, el banjo, el piano y el music box, por otro. Se distinguen así dos grupos con características acústicas similares en sus espectrogramas, como se explicará a continuación.

6.3.3. Evaluación de espectrogramas divididos en dos grupos diferentes

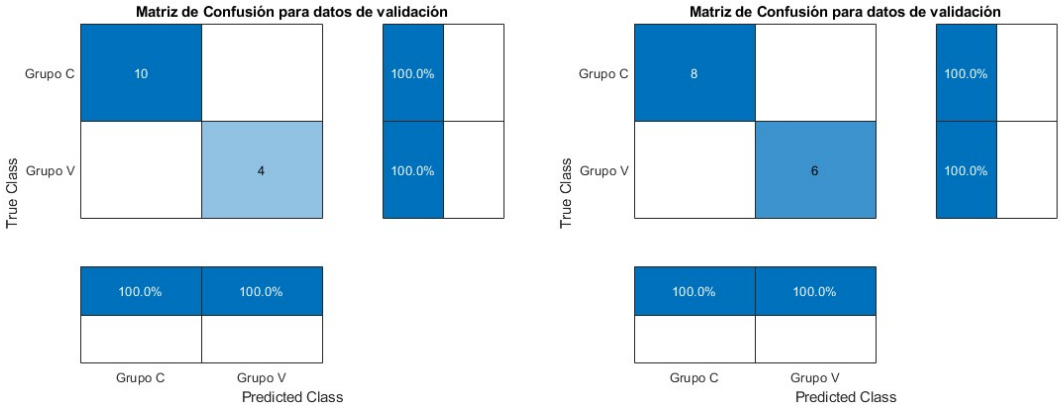
En el tercer experimento, se pretende clasificar el conjunto de espectrogramas en dos categorías con propiedades comunes:

- **Grupo C**: Este grupo agrupa tres instrumentos de cuerda (banjo, piano y guitarra), además de la caja de música (music box). Los espectrogramas de estos instrumentos muestran un sonido más intenso al principio, que se va atenuando con el tiempo.
- **Grupo V**: Este grupo incluye tres instrumentos de viento (trombón, flauta y órgano), junto al violonchelo. Sus espectrogramas presentan un sonido más prolongado cuya intensidad no disminuye gradualmente.

	PRECISIÓN		SENSIBILIDAD		ESPECIFICIDAD		F1 SCORE	
	V	T	V	T	V	T	V	T
Grupo C	**100** %	**100** %	**100** %	**100** %	**100** %	**100** %	**100** %	**100** %
Grupo V	**100** %	**100** %	**100** %	**100** %	**100** %	**100** %	**100** %	**100** %

Tabla 6.7: Métricas de clasificación de espectrogramas 2D y 3D divididos en dos categorías

Se han tomado cinco iteraciones para cada caso y se ha logrado una exactitud del **100 %** para los conjuntos de prueba y validación, tanto en espectrogramas 2D como 3D.

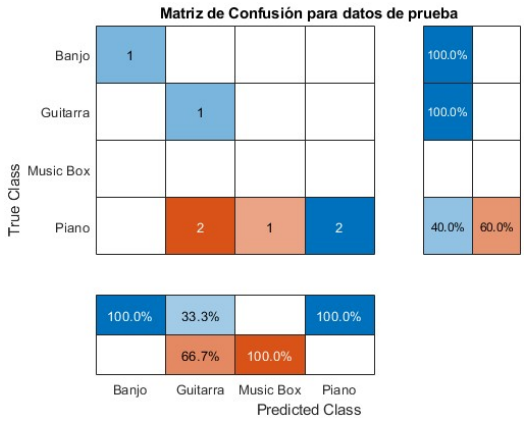


1ª Iteración. Conjunto Validación (2D) 4ª Iteración. Conjunto Validación (3D)

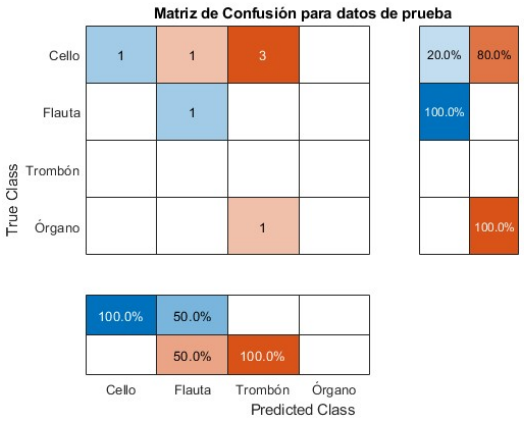
Figura 6.14: Experimento 3: Matrices de confusión binarias de la división de imágenes en dos grupos (Imágenes de creación propia a partir de Matlab)

La agrupación de diferentes clases con características parecidas es de gran utilidad para simplificar el problema de clasificación, pero pierde capacidad para distinguir con exactitud entre esas clases. Por ejemplo, si en el conjunto de prueba

hay un espectrograma de cello, el modelo es capaz de garantizar que pertenece al Grupo V, pero presenta más dificultades para identificar cuál de los cuatro instrumentos que lo componen es en realidad, tal y como se ilustra a continuación:



2ª Iteración. Grupo C (Test 3D)
Exactitud: 57,14 %



6ª Iteración. Grupo V (Test 2D)
Exactitud: 28,57 %

Figura 6.15: Matrices de confusión 4x4 de la clasificación de espectrogramas con características acústicas afines (Imágenes de creación propia a partir de Matlab)

6.3.4. Eliminación de clases problemáticas

El cuarto y último experimento consiste en eliminar ciertas clases que presentan mayores dificultades a la hora de su clasificación. En el conjunto de datos 2D, se han excluido las clases de trombón, cello y music box. En cambio, para el conjunto 3D, se han descartado las clases de guitarra, cello y flauta.

De este modo, se obtendrán matrices de dimensión 5x5, formadas por un total de 85 imágenes; el conjunto de validación se compone de 9 elementos y el conjunto de prueba de 8. Tras ejecutar diez veces consecutivas el modelo, se han logrado los siguientes resultados métricos:

	PRECISIÓN		SENSIBILIDAD		ESPECIFICIDAD		F1 SCORE	
	V	T	V	T	V	T	V	T
Banjo	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**
Flauta	**100 %**	**100 %**	**100 %**	88,89 %	**100 %**	**100 %**	**100 %**	94,12 %
Guitarra	96,30 %	**100 %**	**100 %**	**100 %**	98,57 %	**100 %**	98,11 %	**100 %**
Órgano	**100 %**	95,24 %	**100 %**	**100 %**	**100 %**	98,33 %	**100 %**	97,56 %
Piano	**100 %**	**100 %**	94,44 %	**100 %**	**100 %**	**100 %**	97,14 %	**100 %**

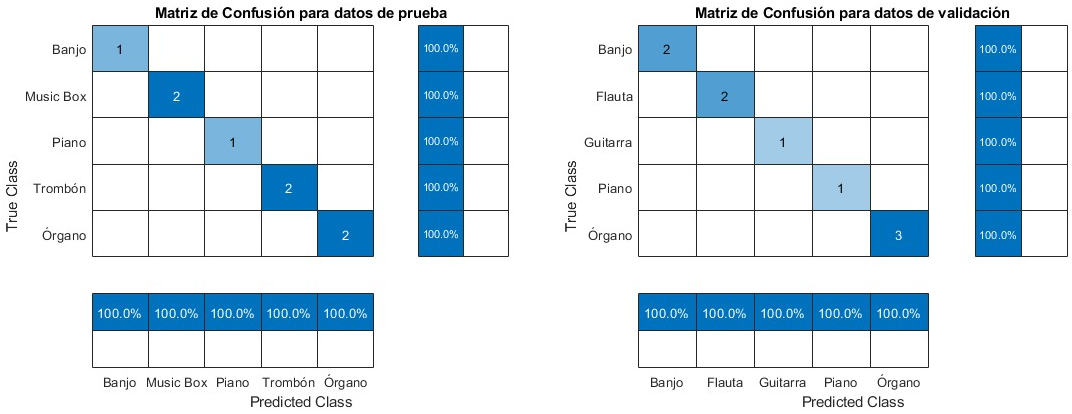
Tabla 6.8: Métricas de clasificación de espectrogramas 2D para cinco clases distintas sin aumento de datos

El modelo 2D presenta una exactitud del **98,89 %** para los datos de validación y un **98,75 %** para los datos de prueba.

	PRECISIÓN		SENSIBILIDAD		ESPECIFICIDAD		F1 SCORE	
	V	T	V	T	V	T	V	T
Banjo	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**	**100 %**
Music Box	**100 %**	**100 %**	94,44 %	**100 %**	**100 %**	**100 %**	97,14 %	**100 %**
Órgano	**100 %**	**100 %**	90,00 %	95,83 %	**100 %**	**100 %**	94,74 %	97,87 %
Piano	93,75 %	**100 %**	**100 %**	**100 %**	98,75 %	**100 %**	96,77 %	**100 %**
Trombón	94,44 %	93,75 %	**100 %**	**100 %**	98,75 %	98,57 %	97,14 %	96,77 %

Tabla 6.9: Métricas de clasificación de espectrogramas 3D para cinco clases distintas sin aumento de datos

La exactitud del modelo 3D es de **97,78 %** y **98,75 %** para los datos de validación y prueba, respectivamente.



9ª Iteración del Conjunto Prueba (3D) 3ª Iteración del Conjunto Validación (2D)

Figura 6.16: Experimento 4: Matrices de confusión 5x5 tras la eliminación de clases problemáticas (Imágenes de creación propia a partir de Matlab)

De los cuatro experimentos realizados, se puede visualizar que en este último los resultados son más satisfactorios.

6.3.5. Gráficas para la comparación de métricas del Conjunto de Validación

Para observar con mayor claridad las diferencias de clasificación entre los cuatro experimentos realizados, se presentarán gráficas de barras para el conjunto de **Validación**, tanto en dos como en tres dimensiones. Estas gráficas resumirán los resultados de las métricas obtenidas (precisión, sensibilidad, F1 score y especificidad) para cada uno de los instrumentos seleccionados. Además, van a ir acompañadas de tablas que incluirán el cálculo de la media, la desviación típica y el rango a partir de los datos anteriores.

Las gráficas de barras con los resultados métricos del Conjunto de Prueba aparecen representadas en el Apéndice A.

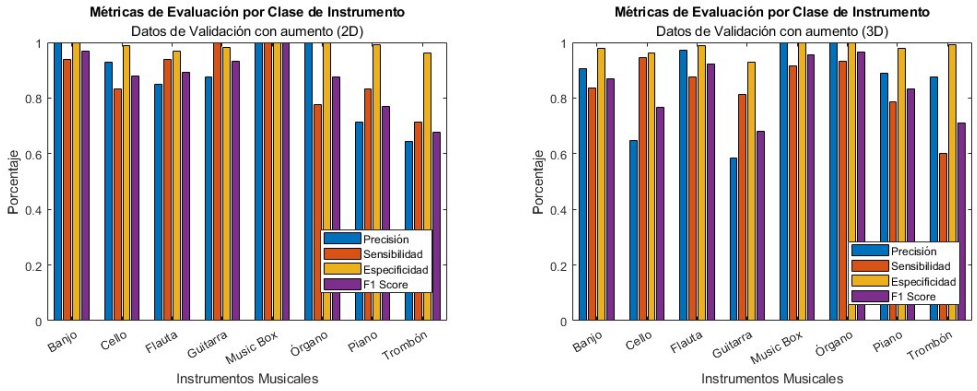
Experimento 1: Clasificación de imágenes con aumento

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	87,63 %	87,92 %	98,62 %	87,40 %
Desv. Típica	12,75 %	9,85 %	1,43 %	9,90 %
Rango	[64,29-100]	[71,43-100]	[96,75-100]	[67,67-100]

Tabla 6.10: Experimento 1: Resumen estadístico de la clasificación 2D

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	85,88 %	83,81 %	97,86 %	83,82 %
Desv. Típica	14,88 %	10,49 %	2,16 %	10,27 %
Rango	[58,33-100]	[60-94,44]	[93,03-100]	[67,91-96,55]

Tabla 6.11: Experimento 1: Resumen estadístico de la clasificación 3D



Conjunto de Validación 2D Conjunto de Validación 3D

Figura 6.17: **Experimento 1:** Gráficas de Métricas para imágenes de Validación 2D y 3D con aumento de datos (creación propia a partir de Matlab)

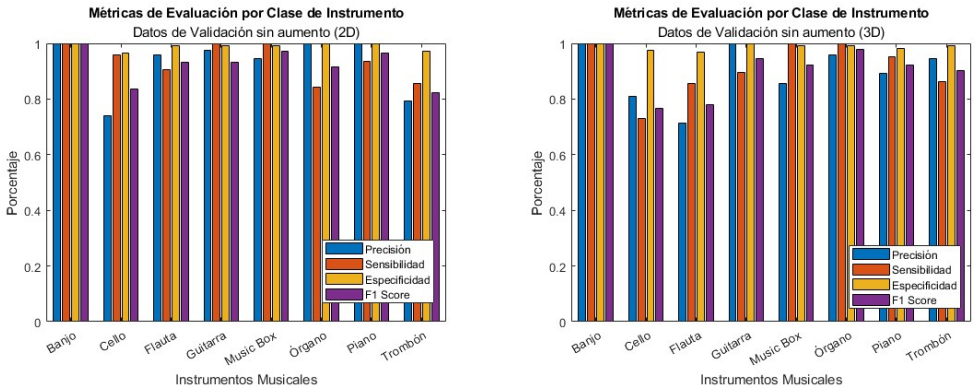
Experimento 2: Clasificación de imágenes sin aumento

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	91,38 %	93,76 %	98,91 %	92,21 %
Desv. Típica	9,46 %	5,96 %	1,20 %	5,92 %
Rango	[74,07-100]	[84,38-100]	[96,57-100]	[82,31-100]

Tabla 6.12: Experimento 2: Resumen estadístico de la clasificación 2D

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	89,71 %	91,21 %	98,72 %	90,21 %
Desv. Típica	9,35 %	8,96 %	1,10 %	8,02 %
Rango	[71,43-100]	[72,86-100]	[96,70-100]	[76,69-100]

Tabla 6.13: Experimento 2: Resumen estadístico de la clasificación 3D



Conjunto de Validación 2D Conjunto de Validación 3D

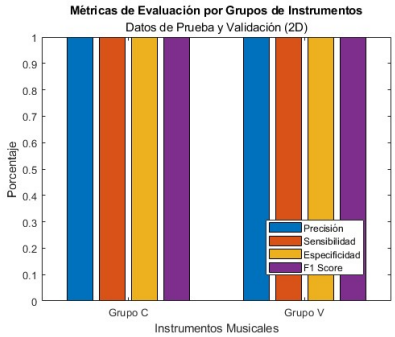
Figura 6.18: **Experimento 2:** Gráficas de Métricas para imágenes de Validación 2D y 3D sin aumento de datos (creación propia a partir de Matlab)

Experimento 3: Clasificación de imágenes en dos grupos

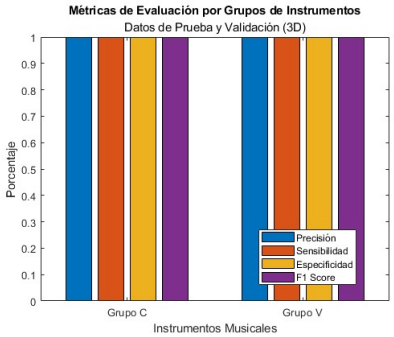
Dado que el resumen estadístico en este experimento es idéntico para ambos conjuntos (bidimensionales y tridimensionales), se omite una de las tablas para evitar redundancias.

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	**100** %	**100** %	**100** %	**100** %
Desv. Típica	0 %	0 %	0 %	0 %
Rango	[100-100]	[100-100]	[100-100]	[100-100]

Tabla 6.14: Experimento 3: Resumen estadístico de la clasificación 2D y 3D



Conjunto de Validación 2D



Conjunto de Validación 3D

Figura 6.19: **Experimento 3:** Gráficas de Métricas para imágenes de Validación 2D y 3D divididas en dos grupos (creación propia a partir de Matlab)

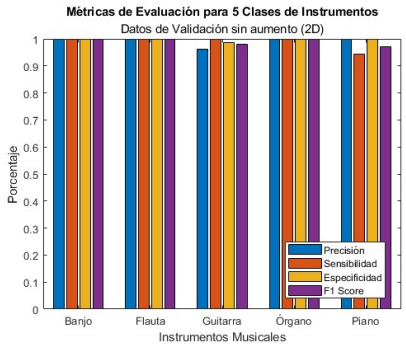
Experimento 4: Clasificación de imágenes en cinco clases

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	99,26 %	98,89 %	99,71 %	99,05 %
Desv. Típica	1,48 %	2,22 %	0,57 %	1,20 %
Rango	[96,30-100]	[94,44-100]	[98,57-100]	[97,14-100]

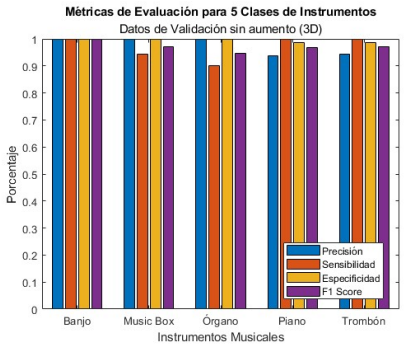
Tabla 6.15: Experimento 4: Resumen estadístico de la clasificación 2D

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	97,64 %	96,89 %	99,50 %	97,16 %
Desv. Típica	2,90 %	4,06 %	0,61 %	1,68 %
Rango	[93,75-100]	[90-100]	[98,75-100]	[94,74-100]

Tabla 6.16: Experimento 4: Resumen estadístico de la clasificación 3D



Conjunto de Validación 2D



Conjunto de Validación 3D

Figura 6.20: **Experimento 4:** Gráficas de Métricas para imágenes de Validación 2D y 3D divididas en cinco clases (creación propia a partir de Matlab)

7

Conclusiones y trabajos futuros

A lo largo de este capítulo, se exponen las conclusiones alcanzadas y se exploran las posibles direcciones para futuras investigaciones derivadas de este estudio.

7.1. Conclusiones

- Se han grabado un total de 136 acordes diferentes utilizando un sintetizador que reproduce el sonido de ocho instrumentos musicales, distribuyendo de forma equitativa 17 acordes por cada instrumento. Se ha prestado especial atención a la eliminación de ruido para no alterar la calidad de los espectrogramas que formarán la base de datos (**Objetivo 1**).

- Se ha logrado encontrar una poderosísima herramienta matemática para descubrir las sucesivas frecuencias de armónicos que componen una señal de audio: *las Transformadas de Fourier*. Estas se consideran una extensión de las Series de Fourier para funciones no periódicas y son capaces de transformar el dominio temporal en frecuencial. Gracias al algoritmo de la Transformada de Fourier de Tiempo Reducido (STFT) se han generado imágenes de *espectrogramas* que contienen información de cada una de las tres variables: tiempo, frecuencia y amplitud (**Objetivo 2**). La información teórica se encuentra en el Capítulo 3 y su procedimiento paso a paso se puede visualizar en el Capítulo 5 donde se ha utilizado el programa Matlab.

- Los espectrogramas 2D y 3D presentan una alta nitidez (**Objetivo 3**) y se ha conseguido mediante el ajuste de hiperparámetros (ver 5.2.3 y 5.2.4).

- La interpretación de los componentes que forman un espectrograma y la identificación de los armónicos que constituyen un acorde se han puesto en práctica en las Secciones 6.1 y 6.2, donde se ha tomado como referencia el Acorde de Do Mayor de la octava central de un piano con frecuencia de 261,6 Hz. En el Capítulo 1 se ha definido el concepto de *armónicos*, cuyas intensidades se distribuyen de manera única dando lugar al *timbre* característico de cada instrumento musical; serán de vital importancia para comprender el desarrollo del trabajo. Su representación gráfica se puede apreciar en las figuras 6.2 y 6.3, así como en los espectrogramas 6.7 y 6.8 (**Objetivos 4 y 5**).

- Se ha optado por utilizar la arquitectura de GoogLeNet, que es un modelo de red neuronal convolucional de aprendizaje supervisado, para el reconocimiento y clasificación de los espectrogramas. Esta red ya ha sido preentrenada con otras imágenes, y en este proyecto se ha sometido a un proceso de entrenamiento con el 80 % de los espectrogramas de la base de datos para el aprendizaje de características o patrones relevantes (**Objetivo 6**). La explicación teórica de las redes neuronales se ha llevado a cabo en el Capítulo 4 y su procedimiento se encuentra a partir de la Sección 5.3.

- Tras ajustar los hiperparámetros para el entrenamiento de la red neuronal (ver Apartado 5.3.4) se ha obtenido un resultado de clasificación bastante satisfactorio. Se han realizado pruebas externas variando los valores de la tasa de aprendizaje, el tamaño de los minilotes por cada iteración y el número total de épocas. Se ha llegado a la conclusión de que un número muy alto de épocas requiere un tiempo computacional elevado y, aunque es capaz de predecir con gran precisión los datos del conjunto de validación, presenta grandes dificultades con las imágenes del conjunto de prueba, que no ha visto con anterioridad. Es decir, puede desembocar en un sobreajuste al memorizar demasiado bien los datos del entrenamiento. Por el contrario, un número muy pequeño de épocas puede conducir a que el modelo no aprenda lo suficiente las características importantes. En cuanto a la tasa de aprendizaje, se ha establecido también un valor que se adapta correctamente a la base de datos, logrando así una buena convergencia (**Objetivo 7**).

- A partir de la Sección 4.4 se han analizado las métricas de clasificación de los espectrogramas (2D y 3D) para datos de validación y prueba a lo largo de cuatro experimentos diferentes (ver Sección 6.3). Se puede observar que los resultados de clasificación de espectrogramas 2D han superado ligeramente a los de espectrogramas 3D. Esto podría deberse a que su menor complejidad facilita a la red neuronal la detección y aprendizaje de patrones de manera más eficiente, especialmente en un conjunto de datos no muy amplio, como es el caso. Además, la eliminación del Método de Aumento de Datos incrementó aproximadamente un 10 % la exactitud del modelo (ver Experimentos 6.3.1 y 6.3.2). A partir de los errores obtenidos, se ha podido

descubrir qué instrumentos comparten características similares y por ello se han dividido en dos grupos: de cuerda y de viento, mayoritariamente (ver Experimento 6.3.3), alcanzando un porcentaje del 100 % en cada una de las métricas. Este experimento es útil para conocer con absoluta certeza a cuál de los dos grupos establecidos pertenece un determinado espectrograma. Sin embargo, tiene sus limitaciones, ya que no predice el instrumento exacto. Dado que cada grupo contiene cuatro instrumentos, una vez identificado el grupo, hay un 25 % de posibilidades de que el espectrograma corresponda al instrumento correcto. Por tanto, se añadió un cuarto experimento (ver Experimento 6.3.4) eliminando tres instrumentos problemáticos a la hora de su clasificación. De este modo, se han conseguido unos resultados muy próximos al 100 %. Se han generado unas gráficas de barras que ayudan a visualizar estas comparaciones (ver A y 6.3.5) (**Objetivos 8, 9 y 10**).

7.2. Trabajos futuros

- En primer lugar, se sugiere la creación de una base de datos con un mayor número de imágenes de espectrogramas. Esto podría favorecer significativamente el rendimiento del modelo de clasificación y aumentar su robustez.
- Se recomienda explorar la clasificación de la base de datos utilizando diferentes arquitecturas de redes neuronales convolucionales, tales como AlexNet, VGGNet, o ResNet. También se sugiere probar otros algoritmos de optimización como RMSprop o Adam. Realizar comparativas de clasificación entre estos modelos podría ser de gran interés para determinar cuál es el más apropiado para este caso en particular.
- Como futura línea de investigación, se propone también la grabación de sonidos de los instrumentos presentes en la base de datos utilizando instrumentos reales y en condiciones ambientales óptimas. A continuación, se podrían comparar estos sonidos auténticos con los simulados por un sintetizador para determinar, mediante una red neuronal, si existen diferencias significativas entre ellos y así evaluar la calidad del sintetizador empleado.
- Por último, se plantea la implementación de herramientas avanzadas de aprendizaje automático para el reconocimiento de instrumentos musicales que suenen simultáneamente en una pista de audio. En particular, se puede explorar el uso de Support Vector Machines (SVM) y Random Forest, como se desarrolla en el proyecto [73]. Además, se podría considerar la opción de integrar Mel-Frequency Cepstral Coefficients (MFCC) o Learning Vector Quantization (LVQ) para el reconocimiento de patrones más complejos.

Bibliografía

[1] J. Boyd-Brent, "Harmony and Proportion," *About Scotland*, 2024. [Online]. Available: http://www.aboutscotland.co.uk/harmony/prop.html

[2] J. Arbonés y P. Milrud, *La armonía es numérica: música y matemáticas*. RBA Contenidos Editoriales y Audiovisuales, S.A., 2012.

[3] C. A. Jiménez Carballo, "Ondas estacionarias," Master's thesis, Escuela de Física del Instituto Tecnológico de Costa Rica, 2018.

[4] Transformación rápida de Fourier FFT. Conceptos básicos. NTi Audio. [Online]. Available: https://www.nti-audio.com/es/servicio/conocimientos/transformacion-rapida-de-fourier-fft

[5] W. Gómez Flores, *Introducción al Análisis de Fourier*, Cinvestat. Unidad Tamaulipas. Fundamentos de Ingeniería Computacional.

[6] Diferencia entre neurona y neuroglía. Psycolab: Laboratorio de Psicología, Ciencia y Emoción de Benalmádena. [Online]. Available: https://www.psycolab.com/diferencia-entre-neurona-y-neuroglia/

[7] (2019) Redes Neuronales Perceptrón Estructura, Tipos [Weka]. Solo para entendidos. [Online]. Available: https://www.soloentendidos.com/redes-neuronales-estructura-tipos-weka-2119

[8] C. Bonilla Carrión, "Redes Convolucionales," Master's thesis, Universidad de Sevilla, 2020.

[9] J. A. Ascencio Laguna, C. D. Martner Peyrelongue y A. Bustos Rosales, "Implementación de un modelo de predicción de tráfico con aprendizaje profundo," Master's thesis, Instituto Mexicano del Transporte, 2022.

[10] M. Esparza Andrés, "Definición y aplicación de un entorno de programación para Deep Learning aplicado al procesado de imágenes," Master's thesis, Universidad de Zaragoza, 2018.

[11] D. Calvo. (2017) Red Neuronal Convolucional CNN. [Online]. Available: https://www.diegocalvo.es/red-neuronal-convolucional/

[12] A. Saad Almryad y H. Kutucu, "Automatic identification for field butterflies by convolutional neural networks," *Engineering Science and Technology, an International Journal*, 2019-2020.

[13] (2024) Introducción a las redes neuronales convolucionales (CNN). DataCamp. [Online]. Available: https://www.datacamp.com/es/tutorial/introduction-toconvolutionalneural-networks-cnns?dc_referrer=https%3A%2F%2Fwww.google.com%2F

[14] T. Shah, "About Train, Validation and Test Sets in Machine Learning," *Towards Data Science*, 2017. [Online]. Available: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

[15] J. Buck. (2022) Matlab spectrogram tutorial. University of Massachusetts Dartmouth. [Online]. Available: https://www.youtube.com/watch?v=KU53LnEgn4w&t=8s

[16] (2020) Curso de MATLAB 81 - Espectrograma de Áudio. 2001 Engenharia. [Online]. Available: https://www.youtube.com/watch?v=O9-3Dtwe9Ug&t=120s

[17] spectrogram. Espectrograma utilizando la transformada de Fourier de tiempo corto. MathWorks. [Online]. Available: https://es.mathworks.com/help/signal/ref/spectrogram.html

[18] N. Faruqui. (2022) Fruit Classification using GoogleNet Convolutional Neural Network (CNN). [Online]. Available: https://www.youtube.com/watch?v=58-1KmsIEcQ

[19] Transferencia del aprendizaje mediante una red preentrenada. MathWorks. [Online]. Available: https://es.mathworks.com/help/deeplearning/ug/transfer-learning-using-pretrained-network.html

[20] A. Danz. Split data to train, test and validation. MathWorks. [Online]. Available: https://es.mathworks.com/matlabcentral/answers/648643-split-data-to-train-test-and-validation

[21] E. Maor, *La música y los números: De Pitágoras a Schoenberg.* Turner Noema, 2018.

[22] M. Lado Bascoy, "Ármonicos e series de fourier," Master's thesis, Universidad de Santiago de Compostela. Facultad de Matemáticas, 2020-2021.

[23] B. le Bovier de Fontenelle, 1766, páginas 424-438.

[24] J. A. Cortés, F. A. Medina y J. A. Chaves, "Del Análisis de Fourier a las Wavelets Análisis de Fourier," *Revista Scientia et Technica*, vol. XIII, no. 34, Mayo 2007, Universidad Tecnológica de Pereira.

[25] M. Lucas Rodríguez, "Detección automática de géneros musicales," Master's thesis, Universidad Politécnica de Madrid, 2021.

[26] B. Ruz, "I Unidad: Fundamentos del Sonido," Master's thesis, Universidad Metropolitana de Ciencias de la Educación de Chile. Facultad de Artes y Educación Física. Departamento de Música, 2018.

[27] H. Grabner, *Teoría general de la música.* Akal S.A., 2001.

[28] C. Schmidt-Jones. (2007) Understanding basic music theory. Capítulo 3: The Physical Bases. Páginas 95-115. [Online]. Available: https://www.academia.edu/41595678/Understanding_Basic_Music_Theory

[29] *Física avanzada 1 (AP Physics 1). Unidad 9: Ondas y sonido.* Khan Academy. [Online]. Available: https://es.khanacademy.org/science/ap-physics-1/ap-mechanical-waves-and-sound/standing-waves-ap/a/standing-waves-review-ap

[30] "El fenómeno físico-armónico y la serie armónica," *Acústica Musical*, 2021. [Online]. Available: https://acusticamusical.info/sonido/fisico-armonico-y-serie-armonica/

[31] (2022) Índice acústico científico. [Online]. Available: https://es.wikipedia.org/wiki/%C3%8Dndice_ac%C3%BAstico_cient%C3%ADfico

[32] H. Barco Rios y E. Rojas, *Física General para estudiantes de ingeniería.* Universidad Nacional de Colombia. Sede Manizales., 1996.

[33] A. Herrera Escudero, "Movimiento ondulatorio," Master's thesis, Universidad Veracruzana, 2014.

[34] M. Figueras Atienza, *Ondas: Introducción a los fenómenos ondulatorios.* Universitat Oberta de Catalunya, 2013.

[35] D. Jaramillo Chamba y L. Chuquimarca Jiménez, "Estudio comprensivo de la Transformada de Fourier Discreta para el análisis de señales digitales," *Universidad Estatal Península de Santa Elena, UPSE*, 2022. [Online]. Available: http://scielo.senescyt.gob.ec/pdf/rctu/v9n1/1390-7697-rctu-9-01-00075.pdf

[36] J. González-Dávila, "Matemáticas y música. sobre la contribución de las matemáticas a la teoría del sonido," *UPV/EHU - Campus de Álava*.

[37] J.J. O'Connor y E.F. Robertson, "Daniel Bernoulli," *MacTutor History of Mathematics, University of St Andrews, Scotland*, 1998. [Online]. Available: https://mathshistory.standrews.ac.uk/Biographies/Bernoulli_Daniel/

[38] J. J. O'Connor y E. F. Robertson, "Jean Le Rond d'Alembert," *MacTutor History of Mathematics, University of St Andrews, Scotland*, 1998. [Online]. Available: https://mathshistory.st-andrews.ac.uk/Biographies/DAlembert/

[39] A. Nava Cesar, "Proyecto de señales y sistemas. Análisis de fourier en la música," Master's thesis, Ingeniería en Cibernética y Sistemas Computacionales. Universidad la Salle.

[40] "Ondas estacionarias. Cuerda vibrante." Master's thesis, Laboratorio de Física, CC Físicas, UCM, 2013-2014.

[41] J. Duoandikoetxea, *Lecciones sobre las series y transformadas de Fourier*, 2003, Universidad Nacional Autónoma de Nicaragua, Managua.

[42] D. Alcaraz Candela, *Series y transformadas de Fourier. Capítulo 2*, Departamento de Matemática Aplicada y Estadística. Universidad Politécnica de Cartagena.

[43] M. Gómez Martín, *Clasificación de voces a través de Series de Fourier y Redes Neuronales*, 2023, Escuela Técnica Superior de Ingeniería Informática.

[44] J. Cooley and J. Tukey, *An Algorithm for the Machine Calculation of Complex Fourier Series*, 1965, Math. Comp. 19:297-301.

[45] Spectrogram Computation with Signal Processing Toolbox. MathWorks. [Online]. Available: https://es.mathworks.com/help/signal/ug/spectrogram-computation-with-signal-processing-toolbox.html

[46] Espectrograma utilizando la transformada de Fourier de tiempo corto. MathWorks. [Online]. Available: https://es.mathworks.com/help/signal/ref/spectrogram.html#bultmx7_sep_mw_c056db1e-cade-47af-bf56-37cd76eee5db

[47] E. Flórez, S. Cardona, L. Jordi, "Selección de la ventana temporal en la transformada de Fourier en tiempos cortos utilizada en el análisis de señales de vibración para determinar planos en las ruedas de un tren," *Revista Facultad de Ingeniería Universidad de Antioquia*, 2008. [Online]. Available: http://www.scielo.org.co/scielo.php?pid=S0120-62302009000400013&script=sci_arttext

[48] L. Colomer. (2016) Acústica musical. Capítulo 10. Análisis espectral de los sonidos musicales. [Online]. Available: https://cursodeacusticamusical.blogspot.com/2016/02/capitulo-10-analisis-espectral-de-los.html

[49] A. A. Moreno, "Clasificación de imágenes usando redes neuronales convolucionales en Python," Master's thesis, Universidad de Sevilla, 2019.

[50] D. J. Matich, "Redes Neuronales: Conceptos Básicos y Aplicaciones," Master's thesis, Universidad Tecnológica Nacional –, 2001.

[51] V. Pastor Ruiz, "Desarrollo de Redes Neuronales para resolución de problemas de estructuras," Master's thesis, Universidad Politécnica de Madrid, 2021.

[52] J. de la Iglesia López, "Redes neuronales artificiales en el contexto de la visión artificial," Master's thesis, Universidad Complutense de Madrid, 2023.

[53] P. Isasi Viñuela e I. Galván León, *Redes de Neuronas Artificiales. Un enfoque práctico*. Pearson Educación, S.A., Prentice Hall, 2004.

[54] Y. Aljure Jiménez, "Clasificación de Flores con Redes Neuronales Convolucionales," Master's thesis, Universidad de Antioquía, 2019.

[55] G. Kalmutskyy, "Simulación de un sistema de clasificación robotizado de propósito general utilizando técnicas de Deep-Learning y visión artificial en Python," Master's thesis, Universidad de Almería, 2020/2021.

[56] (2024) El descenso del gradiente: La brújula del Machine Learning. Verne Technology Group. [Online]. Available: https://www.vernegroup.com/actualidad/tecnologia/descenso-gradiente-brujula-machine-learning/

[57] J. J. Cabrera Mora, "Entrenamiento, optimización y validación de una CNN para la localización de un robot móvil mediante tareas de clasificación y regresión," Master's thesis, Universidad Miguel Hernández de Elche, 2021.

[58] trainingOptions. Opciones para entrenar una red neuronal de deep learning. MathWorks. [Online]. Available: https://es.mathworks.com/help/deeplearning/ref/trainingoptions.html#bu59f0q_sep_mw_564dc803-5731-47e9-b875-1c827f0cc8fe

[59] M. Fahmy Amin, *Confusion Matrix in Three-class Classification Problems: A Step by Step Tutorial*, 2023, Journal of Engineering Research, Volume 7, Issue 1, Article 26. [Online]. Available: https://digitalcommons.aaru.edu.jo/cgi/viewcontent.cgi?article=1115&context=erjeng

[60] A. Martín Pérez , *Diseño de un sistema basado en bosques aleatorios para la detección de tumores cerebrales mediante imágenes hiperespectrales*, Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación. [Online]. Available: https://oa.upm.es/66908/1/TFG_ALBERTO_MARTIN_PEREZ.pdf

[61] L. D. Gómez Silva, "Implementación de un sistema de visión artificial de detección y evasión de obstáculos para un robot móvil tipo oruga," Master's thesis, Universidad Autónoma de Bucaramanga, 2023.

[62] J. I. Morocho Jiménez, "Detección de tumores cutáneos malignos y benignos utilizando una red neuronal convolucional," Master's thesis, Escuela Politécnica Nacional, 2019.

[63] G. Gutierrez y A. Santiago, "Diseño de un sistema de control de calidad para tomates cherry usando biosensores y Deep Learning con Googlenet en Matlab," Master's thesis, Universidad Ricardo Palma, 2023.

[64] J. J. Carballo Pacheco, "Clasificación de imágenes médicas con técnicas de Deep Learning," Master's thesis, Universidad de Extremadura, 2022.

[65] L. R. Barba Guamán, "Uso de técnicas deep learning para reconocimiento de objetos en áreas rurales," Master's thesis, Universidad Politécnica de Madrid, 2021.

[66] M. Avilés Camarmas, A. Berzosa Tordesillas y J. Granizo Egido, "Detección de vehículos en movimiento en vídeos mediante técnicas de aprendizaje profundo," Master's thesis, Universidad Complutense de Madrid, 2020-2021.

[67] R. Rodríguez Abril, "Googlenet," *Un artículo de La Máquina Oráculo*. [Online]. Available: https://lamaquinaoraculo.com/deep-learning/googlenet/

[68] audioread. Leer un archivo de audio. MathWorks. [Online]. Available: https://es.mathworks.com/help/matlab/ref/audioread.html

[69] R. Sánchez, "Teoría de la señal: Representar el espectro frecuencial de un archivo audio con Matlab," 2016. [Online]. Available: http://rubensm.com/representar-el-espectro-frecuencial-de-un-archivo-audio-con-matlab/

[70] FFT Transformada rápida de Fourier. MathWorks. [Online]. Available: https://es.mathworks.com/help/matlab/ref/fft.html

[71] R. Coetsee. MATLAB Onramp. Curso online de MathWorks. [Online]. Available: https://matlabacademy.mathworks.com/es/details/matlab-onramp/gettingstarted

[72] P. Bourke, "Fast Fourier Transform," 1993. [Online]. Available: https://paulbourke.net/miscellaneous/dft/

[73] M. Juan Monguillot, "Sistema de detección de instrumentos musicales en señales de audio," Master's thesis, Universitat Oberta de Catalunya, 2020.

[74] Clasificar una imagen con GoogLeNet. MathWorks. [Online]. Available: https://es.mathworks.com/help/deeplearning/ug/classify-image-using-googlenet.html

[75] Configurar parámetros y entrenar una red neuronal convolucional. MathWorks. [Online]. Available: https://es.mathworks.com/help/deeplearning/ug/setting-up-parameters-and-training-of-a-convnet.html

[76] R. Coetsee. Deep Learning Onramp. Curso online de MathWorks. [Online]. Available: https://matlabacademy.mathworks.com/es/details/deep-learning-onramp/deeplearning

[77] R. Coetsee. Deep Learning with MATLAB. Curso online de MathWorks. [Online]. Available: https://matlabacademy.mathworks.com/es/details/deep-learning-with-matlab/mldl

[78] (2022) Frecuencias de afinación del piano. [Online]. Available: https://es.wikipedia.org/wiki/Frecuencias_de_afinaci%C3%B3n_del_piano

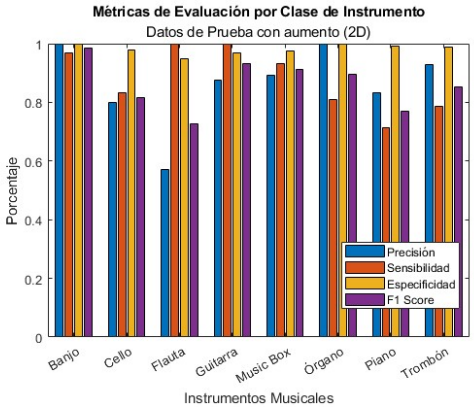
Apéndices

A

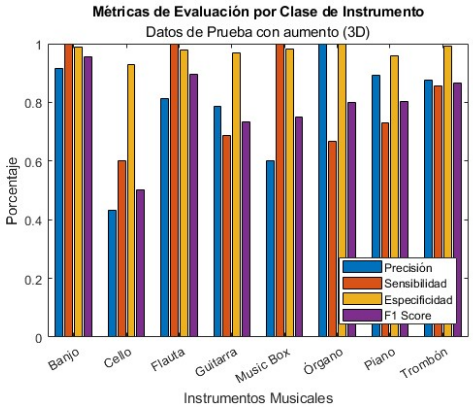
Resumen estadístico del Conjunto de Prueba

Siguiendo el mismo enfoque que en el apartado 6.3.5, se presentan las gráficas de barras junto a diferentes tablas que contienen el resumen estadístico de las métricas para el **Conjunto de Prueba** de cada uno de los experimentos realizados. Por tanto, se podrán comparar los resultados obtenidos de manera más eficiente.

Experimento 1: Clasificación de imágenes con aumento



Conjunto de Prueba 2D



Conjunto de Prueba 3D

Figura A.1: **Experimento 1:** Gráficas de Métricas para imágenes de Prueba 2D y 3D con aumento de datos (creación propia a partir de Matlab)

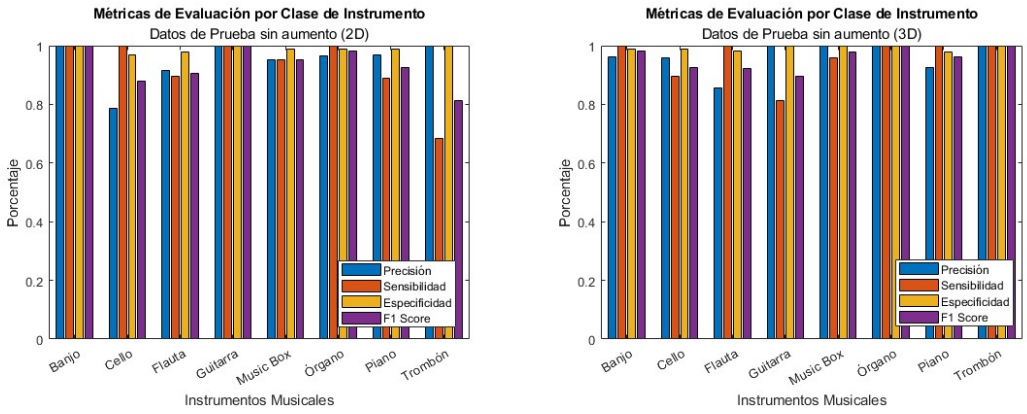
	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	86,27 %	88,06 %	98,13 %	86,11 %
Desv. Típica	12,88 %	10,19 %	1,64 %	8,11 %
Rango	[57,14-100]	[71,43-100]	[94,86-100]	[72,73-98,41]

Tabla A.1: Experimento 1: Resumen estadístico de la clasificación 2D

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	78,95 %	81,76 %	97,44 %	78,85 %
Desv. Típica	17,38 %	15,66 %	2,14 %	12,85 %
Rango	[43,33-100]	[60-100]	[92,77-100]	[50,32-95,65]

Tabla A.2: Experimento 1: Resumen estadístico de la clasificación 3D

Experimento 2: Clasificación de imágenes sin aumento



Conjunto de Prueba 2D Conjunto de Prueba 3D

Figura A.2: **Experimento 2:** Gráficas de Métricas para imágenes de Prueba 2D y 3D sin aumento de datos (creación propia a partir de Matlab)

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	94,85 %	92,78 %	98,95 %	93,26 %
Desv. Típica	6,72 %	10,18 %	1,04 %	6,09 %
Rango	[78,57-100]	[68,52-100]	[96,92-100]	[81,32-100]

Tabla A.3: Experimento 2: Resumen estadístico de la clasificación 2D

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	96,32 %	95,83 %	99,25 %	95,85 %
Desv. Típica	4,74 %	6,50 %	0,83 %	3,63 %
Rango	[85,71-100]	[81,25-100]	[97,88-100]	[89,66-100]

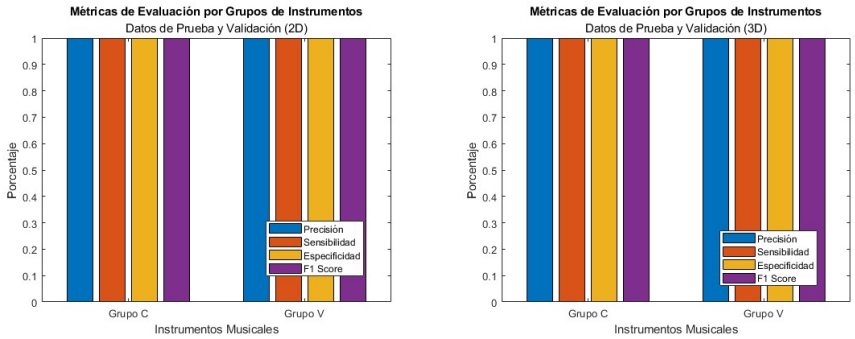
Tabla A.4: Experimento 2: Resumen estadístico de la clasificación 3D

Experimento 3: Clasificación de imágenes en dos grupos

Del mismo modo, el resumen estadístico de este experimento es idéntico para los conjuntos 2D y 3D, así que se omite una de las tablas para evitar redundancias. Además, se puede observar también que los resultados coinciden con los ya obtenidos en el Conjunto de Validación (6.14).

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	**100** %	**100** %	**100** %	**100** %
Desv. Típica	0 %	0 %	0 %	0 %
Rango	[100-100]	[100-100]	[100-100]	[100-100]

Tabla A.5: Experimento 3: Resumen estadístico de la clasificación 2D y 3D



Conjunto de Prueba 2D Conjunto de Prueba 3D

Figura A.3: **Experimento 3:** Gráficas de Métricas para imágenes de Prueba 2D y 3D divididas en dos grupos (creación propia a partir de Matlab)

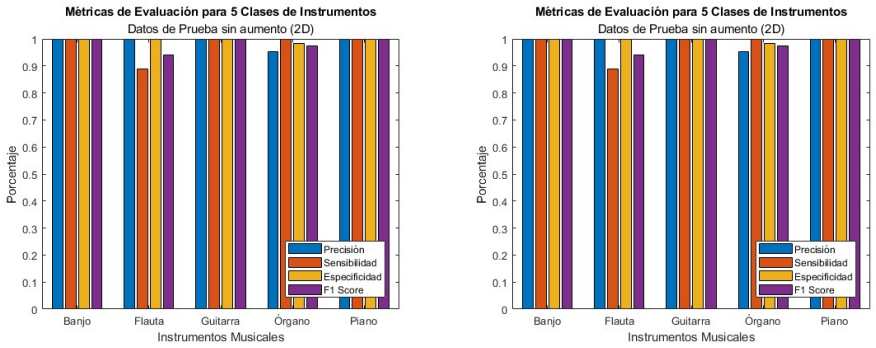
Experimento 4: Clasificación de imágenes en cinco clases

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	99,05 %	97,78 %	99,67 %	98,34 %
Desv. Típica	1,90 %	4,44 %	0,67 %	2,31 %
Rango	[95,24-100]	[88,89-100]	[98,33-100]	[94,12-100]

Tabla A.6: Experimento 4: Resumen estadístico de la clasificación 2D

	PRECISIÓN	SENSIBILIDAD	ESPECIFICIDAD	F1 SCORE
Media	98,75 %	99,17 %	99,71 %	98,93 %
Desv. Típica	2,5 %	1,67 %	0,57 %	1,36 %
Rango	[93,75-100]	[95,83-100]	[98,57-100]	[96,77-100]

Tabla A.7: Experimento 4: Resumen estadístico de la clasificación 3D



Conjunto de Prueba 2D Conjunto de Prueba 3D

Figura A.4: **Experimento 4:** Gráficas de Métricas para imágenes de Prueba 2D y 3D divididas en cinco clases (creación propia a partir de Matlab)

B

Algoritmo FFT

En el Algoritmo 1, se explicará con detalle el mecanismo de la Transformada Rápida de Fourier creada por Cooley y Tukey, tal y como ya se comentó en el Apartado 3.2.2.

Este algoritmo se basa en la estrategia de *Divide y Vencerás*, la cual se encargará de separar el conjunto inicial de datos de $\mathbb{C}^N$ en dos categorías: de índice par e impar, tomando N como una potencia de dos. A continuación, se calcularán dos DFTs de dimensión $\frac{N}{2}$, cuya unión representa la DFT de N puntos. Este argumento se repite de manera sucesiva, reduciendo así el número de operaciones realizadas.

La herramienta de la FFT aprovecha las propiedades de simetría y periodicidad del término W_N, notación definida a lo largo del Apartado 3.2.1. Por tanto, se tiene en cuenta que:

$$W_N^{\alpha+N} = W_N^{\alpha} \qquad W_N^N = 1$$

$$W_N^{\alpha+\frac{N}{2}} = -W_N^{\alpha} \qquad W_N^2 = W_{\frac{N}{2}}$$

para cualquier valor de α entero.

Tras visualizar el algoritmo, se puede apreciar con mayor claridad que las dos submatrices de la izquierda son totalmente idénticas. Además, multiplicando cualquiera de ellas por W_N^u siendo u el número de fila, se obtiene la submatriz de arriba a la derecha. Y la de abajo a la derecha es básicamente la opuesta de la anterior.

Algoritmo 1 *Transformada Rápida de Fourier (FFT)*

1: Se considera que el número de muestras de la señal es una potencia de dos entera $\rightarrow$ $N = 2^k \quad \forall k \in \mathbb{N}$

2: Si N no cumple con el tamaño indicado en el paso anterior, se utiliza el método de *Rellenado de ceros*, añadiendo tantos ceros como sea necesario al final de la señal hasta que su longitud alcance la potencia de dos más próxima. Esto no causa cambios en el espectro de Fourier.

3: Reestructurar la matriz $f(k)$ anterior $\rightarrow$ Colocar en primer lugar los elementos de índice par y, acto seguido, los de índice impar.

$$\begin{pmatrix} F(0) \\ F(1) \\ \vdots \\ F(\frac{N}{2}-1) \\ F(\frac{N}{2}) \\ F(\frac{N}{2}+1) \\ \vdots \\ F(N-1) \end{pmatrix} = C(W_N) \begin{pmatrix} f(0) \\ f(2) \\ \vdots \\ f(N-2) \\ f(1) \\ f(3) \\ \vdots \\ f(N-1) \end{pmatrix}$$

4: El paso 3 implica un cambio de orden en las columnas de la matriz $A(W_N)$ $\rightarrow$ Se obtiene una nueva matriz $C(W_N)$ de la forma:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^2 & \cdots & W_N^{N-2} & W_N & W_N^3 & \cdots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-2} & \cdots & W_N^{(\frac{N}{2}-1)(N-2)} & W_N^{(\frac{N}{2}-1)} & W_N^{3(\frac{N}{2}-1)} & \cdots & W_N^{(\frac{N}{2}-1)(N-1)} \\ 1 & 1 & \cdots & 1 & -1 & -1 & \cdots & -1 \\ 1 & W_N^2 & \cdots & W_N^{N-2} & -W_N & -W_N^3 & \cdots & -W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-2} & \cdots & W_N^{(N-1)(N-2)} & -W_N^{(\frac{N}{2}-1)} & -W_N^{3(\frac{N}{2}-1)} & \cdots & -W_N^{(\frac{N}{2}-1)(N-1)} \end{pmatrix}$$

5: $C(W_N)$ se compone de cuatro submatrices $\frac{N}{2} x \frac{N}{2}$ a partir de la relación:

$$\begin{pmatrix} T(W_{\frac{N}{2}}) & W_N^u T(W_{\frac{N}{2}}) \\ T(W_{\frac{N}{2}}) & W_N^{\widehat{u}} T(W_{\frac{N}{2}}) \end{pmatrix} \quad \text{con los índices } u = 0, ..., \frac{N}{2} - 1;\ \widehat{u} = \frac{N}{2}, ..., N-1$$

Además, $T(W_{\frac{N}{2}}) \in \mathbb{C}^{\frac{N}{2}}$ y $W_N^u\, T(W_{\frac{N}{2}}) = -W_N^{\widehat{u}}\, T(W_{\frac{N}{2}})$

6: Sean $E(k) = \sum f(2k)$ y $O(k) = \sum f(2k+1)$ $\forall k = 0, 1, .., \frac{N}{2} - 1$ tales que:
$F(u) = E(k) + W_N^u\ O(k) \quad y \quad F(u + \frac{N}{2}) = E(k) - W_N^u\ O(k)$

7: Reiterar el proceso hasta que $T(W_{N'}) \in \mathbb{C}^2 \ \rightarrow \ N' = 2$

Las páginas de referencia para el desarrollo del algoritmo son ([43], [41], [5]).