Universidad de Alcalá

Escuela Politécnica Superior

GRADO EN INGENIERÍA EN SISTEMAS DE TELECOMUNICACIÓN

Trabajo Fin de Grado

Clasificación acústica de aves mediante la extracción de características del sonido a través de los parámetros MFCC

ESCUELA POLITECNICA SUPERIOR

Autor: Marcos De Rodrigo Talavera

Tutor: Roberto Gil Pita

2018

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

GRADO EN INGENIERÍA EN SISTEMAS DE TELECOMUNICACIÓN

Trabajo Fin de Grado

Clasificación acústica de aves mediante la extracción de características del sonido a través de los parámetros MFCC

Autor: Marcos De Rodrigo Talavera

Tutor/es: Roberto Gil Pita

TRIBUNAL:

Presidente: Manuel Utrilla Manso

Vocal 1º: Manuel Rosa Zurera

Vocal 2º: Roberto Gil Pita

FECHA: 16 de Julio, 2018

Índice general

Resumen

Actualmente, la tecnología tiende cada vez más a unirse al ámbito de la Inteligencia Artificial, donde los propios sistemas toman decisiones. En las últimas décadas se han desarrollado diversos sistemas fruto de una combinación entre el procesado de audio y los algoritmos de Aprendizaje Automático, principalmente aplicaciones orientadas a voz y música. Sin embargo, el reconocimiento y clasificación de especies animales es aún un campo relativamente inexplorado.

En este trabajo, se aportará una visión general de todo el proceso llevado a cabo para realizar clasificación automática de audio, utilizando el procesado digital de audio denominado Extracción de Características, y los algoritmos de clasificación del Aprendizaje Automático Supervisado basados en Reconocimiento de Patrones.

Palabras clave: Extracción de Características, Aprendizaje Automático, Reconocimiento de patrones, Clasificador acústico.

Abstract

Nowadays, technology tends to join the Artificial Inteligence field, where systems take decisions. In the last decades, diverse systems have been developed as a combination of audio processing and Machine Learning algorithms, mainly aplications focused on voice and music. However, animal species sound recognition and classification is yet a relatively unexplored field.

This project will provide a general overview of the whole process carried out to perform automatic audio classification, using digital audio processing denominated as Feature Extraction, and the Supervise Machine Learning classification algorithms based on Pattern Recognition.

Keywords: Feature Extraction, Machine Learning, Pattern Recognition, Acoustic classifier.

Introducción

La clasificación y reconocimiento de señales de audio es actualmente un campo de gran interés. Durante los últimos años se han desarrollado diversas aplicaciones resultantes de una combinación entre procesado de audio y aprendizaje automático. Entre las más desarrolladas se encuentran aquellas enfocadas al ámbito de la voz humana, como por ejemplo el reconocimiento del hablante, que permite reconocer a distintos individuos a través de la voz, o el reconocimiento y transcripción del habla en aplicaciones de mensajería.

Un campo menos desarrollado dentro de la clasificación automática de señales de audio es el reconocimiento de sonidos ambientales, en el caso de este trabajo el sonido de especies animales, más concretamente, el de aves, el cual tiene diversas aplicaciones, como la detección de eventos sonoros, el seguimiento de migraciones o la clasificación automática de especies que pueblan una determinada zona.

En este trabajo se ha desarrollado una herramienta que permite la clasificación de diferentes especies de aves y se pretende dar una visión general de todo el proceso de clasificación, desde la extracción de características del audio hasta los algoritmos de aprendizaje automático utilizados para la clasificación.

La extracción de características de las señales de audio es una parte fundamental de la clasificación, ya que serán estas las que nos permitirán diferenciar especies distintas. En este trabajo son varias las características que se extraen de las distintas señales, siendo la más relevante los coeficientes cepstrales de la escala de Mel, que han demostrado su capacidad de aportar información útil y relevante.

Los algoritmos de clasificación emplean las características extraídas para determinar a que especie pertenece la señal de dichas características. En el caso de este proyecto, la herramienta cuenta con un algoritmo de clasificación conocido como “K-Nearest-Neighbors”, que contrastará las “K” muestras vecinas más cercanas de la base de datos con las de la señal en cuestión, como se verá a lo largo del documento.

Este proyecto recoge en sus últimos apartados una serie de simulaciones reales llevadas a cabo con la herramienta desarrollada, donde se puede observar el funcionamiento real del sistema ante diferentes escenarios.

Resumen extendido

Una forma resumida de exponer el funcionamiento de la herramienta es la siguiente: se le suministra al sistema una base de datos de audio (en formato .mp3) con la que trabajar, el sistema extrae las características de todos los audios y forma dos grupos, uno dedicado a entrenamiento y otro a evaluación. El sistema contrasta cada ejemplo del grupo de evaluación contra el grupo de entrenamiento completo para decidir a qué especie pertenece cada uno de los audios a evaluar. La forma en la que toma la decisión es comparando las etiquetas de los "K" vecinos más cercanos del grupo de entrenamiento, entendiendo por vecinos más cercanos aquellos que más se aproximan en el valor de sus características. Si la mayoría de los vecinos pertenecen a una determinada especie, el sistema determinará que el audio evaluado pertenece a dicha especie. Una vez determinada la especie, se comparará este resultado con la etiqueta del audio evaluado para determinar si la clasificación ha sido acertada o no y de esta forma determinar la probabilidad de error del sistema.

Para la herramienta de clasificación acústica de aves en primer lugar será necesario aportar una base de datos de audio extensa, ya que será necesario dividir esta en dos grupos, uno para entrenar al sistema y otro para evaluar el comportamiento de este y determinar si se comporta de la forma esperada. Los audios que formen la base de datos deberán estar correctamente ordenados en directorios y se cargarán al sistema de forma secuencial.

Una vez cargada la base de datos, el sistema procederá a extraer las características de cada uno de los audios. Las características que se extraerán de los audios son principalmente los coeficientes cepstrales de la escala de Mel y otros como la media del tono y su desviación estándar. La forma de extraer estas características, así como su modo de empleo se explicarán con más detalle a lo largo del documento.

A partir de este punto el sistema no trabaja más con los audios, si no que utilizará las características extraídas de ellos como base de datos. Serán estas características las que se repartan entre el conjunto de ejemplos de entrenamiento y el conjunto de ejemplos de evaluación. La forma de dividir la base de datos será asignando una primera parte de esta base de datos a un grupo y el resto al otro, ya que si, por ejemplo, se repartiesen alternadamente podría ocurrir que evaluásemos las características de la primera parte de

un audio con las características de la segunda parte de ese mismo audio, con unas características probablemente idénticas, obteniendo unos falsos resultados positivos.

El conjunto de datos dedicados a entrenamiento irá etiquetado con la especie correspondiente (método denominado aprendizaje automático supervisado) y servirán para que el sistema los almacene en una base de datos contra las que posteriormente se contrastarán los datos del grupo de evaluación. El conjunto de datos destinados a evaluación irá también etiquetado con la especie a la que pertenecen, aunque no se tendrá en cuenta esta etiqueta hasta una vez clasificados, para así evaluar si la clasificación ha sido acertada o no. Si los resultados no son los esperados, existe la posibilidad de asignar mayor porcentaje de la base de datos al conjunto de datos de entrenamiento.

A la hora de determinar a qué especie pertenece un ejemplo del conjunto de evaluación, se empleará el criterio de vecindad o distancia, también conocido como "K-Nearest-Neighbors". Este criterio consiste en comparar las etiquetas de las muestras del conjunto de entrenamiento más cercanas al ejemplo a evaluar para así determinar la especie a la que pertenece dicho ejemplo. La forma de determinar que muestras son las más cercanas a la muestra a evaluar es mediante la distancia euclídea entre los valores de dichas muestras. Se tomará la etiqueta que se repita más veces como la correcta, ponderando más la etiqueta de la muestra que más cerca se encuentre en caso de empate.

La herramienta nos permitirá representar en un plano bidimensional las distintas características extraídas en forma de nube de puntos, a lo que denominaremos espacio de características, donde podremos ver como se agrupan las muestras correspondientes a distintas especies e imprimir en un documento las probabilidades de error y de acierto del sistema para la base de datos aportada, en base a una serie de parámetros de los que hablaremos más adelante en este documento.

Aprendizaje automático supervisado

El aprendizaje automático de las máquinas tiene como objetivo desarrollar algoritmos que permitan aprender sobre un conjunto de observaciones, de tal manera que sea posible hacer predicciones, y de esta forma, tomar decisiones automáticamente, dotando así al sistema de una inteligencia artificial.

Estos algoritmos tienen típicamente dos fases fundamentales:

La **fase de entrenamiento** consiste principalmente en proporcionar ejemplos al sistema, de los cuales éste debe aprender. En este trabajo se empleará el aprendizaje automático supervisado, que consiste en etiquetar dichos ejemplos, es decir, sabemos de antemano su clasificación.

La **fase de evaluación** es llevada a cabo una vez el sistema ha sido entrenado. Esta fase consiste en proporcionar al sistema un subconjunto de ejemplos, que el propio sistema debe etiquetar. Conociendo las verdaderas etiquetas de los datos proporcionados será posible verificar los resultados de la evaluación, caracterizando así al sistema en función del número de aciertos y fallos. En esta fase pueden surgir dos tipos de errores: falsas alarmas, probabilidad de aceptar erróneamente muestras pertenecientes a otra clase, y falsos rechazos, probabilidad de rechazar erróneamente muestras pertenecientes a dicha clase.

Es importante mencionar que todo algoritmo de aprendizaje automático basado en reconocimiento de patrones requiere un procesado previo de la información con la que se trabajará. Este procesado consiste en extraer las características relevantes de la información a tratar, las cuales serán usadas por el sistema tanto en la fase de entrenamiento como en la fase de evaluación.

En el caso de la clasificación de aves, en un primer lugar, será necesario que el sistema aprenda cómo es el sonido de una determinada especie. Posteriormente, estas características serán introducidas al sistema de aprendizaje indicándole de que especie se trata. Finalmente, para evaluar el comportamiento del sistema se deberán introducir datos de la especie en cuestión, así como datos de otras especies.

Bases de datos

En el proceso general de desarrollo de un clasificador, el primer paso es recopilar una base de datos. La elección de la base de datos a utilizar es un aspecto crucial del clasificador, ya que se trata de los ejemplos de los cuales éste debe aprender. Una vez definida la base de datos, es necesario procesar cada uno de los ejemplos para extraer las características más significativas. Todas las características seleccionadas conforman, para todos los ejemplos de la base de datos, el conjunto de datos de entrenamiento y de evaluación, es decir, prescindimos de la muestra de audio en cuestión y trabajamos solo con las características extraídas de ésta.

La principal base de datos estará compuesta por audios extraídos de la página web xeno-canto.org. Se trata de una base de datos orientada al canto de aves, la cual es además rica en detalles acerca de las muestras de audio, aportando información acerca de la fecha y posición geográfica donde fue tomada la muestra, datagramas de los audios, así como especificaciones acerca del origen del audio, si el canto de dicha muestra es de alarma, de apareamiento, etc.

Extracción de características

Tal y como se ha comentado anteriormente, es necesario depurar la información a clasificar para crear el conjunto de datos de entrenamiento. Para ello, es necesario realizar un pre-procesado de cada audio con el objetivo de extraer la información más relevante que permita su clasificación. Por ello, la extracción de características es una parte fundamental dentro de cualquier sistema de reconocimiento de patrones.

El proceso de obtención de características del audio que describan al mismo en diferentes aspectos es denominado “Feature Extraction”. De tal forma que cada descriptor aporte algún tipo de información sobre el audio. Al realizar el pre-procesado del audio, cada audio será descrito mediante un conjunto de características, es decir, cada audio será descrito con un conjunto de valores, formando así un ejemplo del conjunto de datos de entrenamiento. Es importante destacar que, a la hora de definir un conjunto de datos de entrenamiento, todos los ejemplos que forman el mismo deben estar descritos por las mismas características.

Existen diferentes tipos de características, y cada una de ellas aporta una información distinta. Si bien es normal que entre distintas características exista cierta correlación, lo deseable es que la correlación entre estas sea mínima, ya que, si dos características tienen alta correlación entre sí, estarán aportando la misma información, pudiendo ser redundante una de ellas. Por el contrario, cuanta menos correlación exista entre dos características, más información distinta se tendrá para describir el audio. Es importante evitar la redundancia entre características para que el funcionamiento del clasificador sea más eficiente.

Para visualizar a simple vista la correlación entre dos características, es habitual representar las mismas en forma de scattering o dispersión. Esta representación consiste básicamente en enfrentar el valor de cada característica sobre un eje bidimensional para un gran conjunto de audios, conformando así una nube de puntos denominado espacio de características. Puesto que cada característica representa una dimensión, el scattering siempre viene dado en forma bidimensional, es decir, para dos características en concreto. Por ejemplo, en el caso de clasificación de distintas especies de aves, cada una de las especies se representa por el valor de las dos características en cuestión, formando cada una de ellas un eje (eje bidimensional), pudiendo observar distintas concentraciones de puntos para las características de distintas especies. No hay que perder de vista que se trata de una representación bidimensional, es decir, aunque dos nubes de puntos correspondientes a dos especies distintas se solapasen, sería posible que, incluyendo una nueva característica, y, por tanto, una nueva dimensión, estas especies quedasen separadas entre sí en el plano tridimensional.

En caso de realizar un scattering con la misma característica en cada eje, la correlación sería del 100%, por lo que el resultado sería una línea recta en la que veríamos cómo los puntos correspondientes a distintas especies de ave forman distintos grupos a lo largo de la línea.

Como paso previo a la extracción de las características mencionadas, es habitual realizar algún tipo de mejora en el audio a tratar, por ejemplo, eliminar el ruido y componente continua o aplicar un preénfasis a la señal.

Mel-Frequency Cepstral Coefficients

Los MFCC, o Mel Frequency Cepstral Coefficients, son coeficientes para la representación del sonido basados en la percepción auditiva humana. Estos surgen de la necesidad, en el área del reconocimiento de audio automático, de extraer características de las componentes de una señal de audio que sean adecuadas para la identificación de contenido relevante, así como obviar todas aquellas que posean información poco valiosa como el ruido de fondo, emociones, volumen, tono, etc. y que no aportan nada al proceso de reconocimiento.

Estos coeficientes representan la amplitud del espectro de la señal de manera compacta, esto los ha vuelto la técnica de extracción de características más usada en el reconocimiento de sonidos y fueron introducidos por Davis y Mermelstein en los años 80 y han sido el estado del arte desde entonces.

Para su cálculo, primeramente, se aplica un filtro de pre-énfasis a la señal, lo que equivale a un filtro tipo FIR de primer orden con coeficiente de preénfasis “alpha” para realzar la alta frecuencia, lo cual proporciona una mejor relación señal ruido (SNR). Posteriormente se divide la señal en tramas. Esto es consecuencia de que toda señal de audio cambia constantemente en el tiempo, lo cual dificulta enormemente la extracción de características que la puedan diferenciar de otras señales o la identifiquen como similar a aquellas que claramente no lo son para un ser humano. Debido a esto y con el objetivo de simplificar su tratamiento, se asume que en pequeños periodos de tiempo sus características no varían mucho y por tanto se le pueden realizar todo un conjunto de procesamientos con el objetivo de extraer características “estáticas” para cada pequeño tramo de la señal, las cuales, en su conjunto, representarían la señal completa.

El siguiente paso es calcular la potencia espectral de cada trama. Este paso viene motivado por la cóclea humana (un órgano del oído), la cual vibra en diferentes puntos dependiendo de la frecuencia de los sonidos recibidos. Dependiendo del punto de la cóclea que vibra, diferentes nervios informan al cerebro de que ciertas frecuencias están presentes. Nuestra estimación del periodograma realiza una función similar, identificando que frecuencias se encuentran en la trama.

La estimación del periodograma espectral todavía contiene mucha información irrelevante para el reconocimiento automático del audio. En particular, la cóclea no puede discernir la diferencia entre dos frecuencias muy cercanas. Es por esta razón que tomamos

agrupaciones del periodograma y las sumamos para obtener una idea de cuanta energía existe en diferentes regiones de la frecuencia. Esto es llevado a cabo por el banco de filtros de Mel, que son de forma triangular y el ancho de los filtros aumenta con la frecuencia: el primer filtro es muy estrecho y nos da una indicación de cuanta energía existe cerca de los 0Hz. Según aumenta la frecuencia los filtros se vuelven más anchos ya que nos interesan menos las variaciones que se producen a dichas frecuencias. Solo estamos interesados en conocer de forma aproximada cuanta energía existe en cada región de la frecuencia. La escala de Mel nos indica exactamente como espaciar nuestro banco de filtros y como de anchos hacerlos. A diferencia de la escala lineal de frecuencias utilizada en el cómputo de la FFT, la escala de Mel es una escala perceptual de tonalidades equidistantes, es decir, es proporcional al logaritmo de las frecuencias lineales, asemejándose así a la percepción humana. En la expresión (1) se presenta la ecuación que permite pasar de frecuencias a frecuencias de Mel.

$$Mel(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \quad (1)$$

Una vez tenemos las energías de nuestro banco de filtros, tomamos su logaritmo. Esto viene también motivado por la audición humana, ya que no percibimos el sonido de forma lineal si no logarítmica. Generalmente, para duplicar el volumen de un sonido que percibimos, necesitamos aumentar 8 veces la cantidad de energía de este. Esto implica que grandes variaciones en la energía pueden no sonar muy diferentes si el sonido es ya alto para empezar. El logaritmo nos permite usar sustracción de la media cepstral, una técnica de normalización del canal.

El último paso es el computo de la transformada de coseno discreta (DCT) del logaritmo de las energías del banco de filtros. Son dos los motivos por los que se realiza este paso. Debido a que los filtros del banco de filtros se encuentran solapados, las energías de estos están correladas entre sí. La DCT decorrela las energías, lo que significa que las matrices diagonales de covarianza pueden ser usadas para modelar las características de la trama. Pero hay que tener en cuenta que solo 13 de los 26 coeficientes de la DCT se mantienen. Esto es debido a que los coeficientes más altos de la DCT representan cambios rápidos en las energías del banco de filtros y resulta que estos cambios rápidos degradan el funcionamiento del reconocimiento automático de audio, por lo que conseguimos cierta mejora quitándolos.

Un resumen del proceso de obtención de los coeficientes MFCC sería el siguiente:

- Se separa la señal en tramas más pequeñas.
- A cada trama se le aplica la transformada discreta de Fourier (DFT) y se obtiene la potencia espectral de la señal.
- Se aplica el banco de filtros correspondientes a la escala de Mel al espectro obtenido en el paso anterior y se suman las energías obtenidas en cada uno de ellos.
- Se toma el logaritmo de todas las energías de cada frecuencia de Mel.
- Por último, se le aplica la transformada de coseno discreta (DCT) a estos logaritmos.

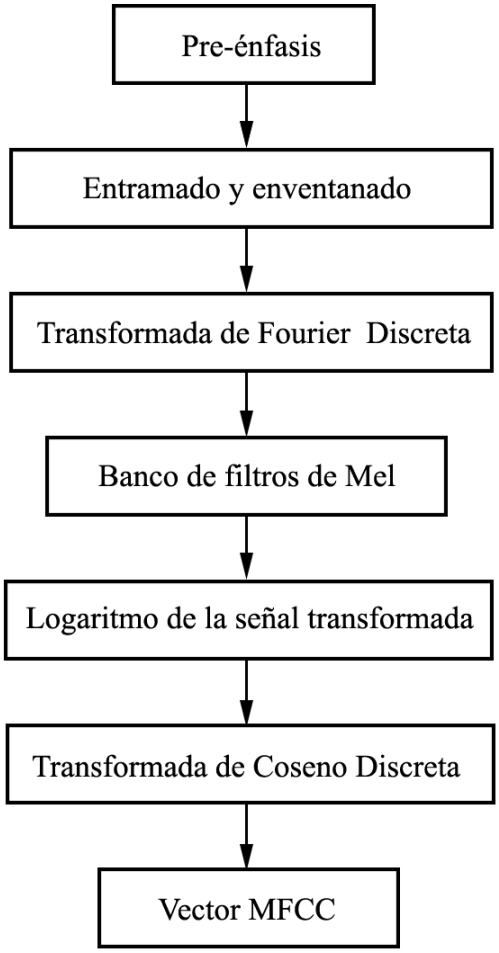


Figura 1. Pasos para la obtención de los coeficientes MFCC.

Clasificación

Los algoritmos de clasificación supervisada se utilizan en problemas en los cuales se conoce a priori el número de clases y los representantes de cada clase. Básicamente consiste en que, para clasificar automáticamente una nueva muestra, se tiene en cuenta la información que se pueda extraer de un conjunto de ejemplos.

Estos algoritmos operan usualmente sobre la información suministrada por un conjunto de ejemplos de entrenamiento que son asumidos como representantes de las clases, y los mismos poseen una etiqueta de clase correcta. A este conjunto de ejemplos se le denomina conjunto de entrenamiento (training set) y es el empleado para la clasificación de nuevas muestras.

El objetivo de estos algoritmos es determinar cuál es la clase, de las que se tiene conocimiento, a la que debe pertenecer una nueva muestra.

K-Nearest Neighbors

Cómo algoritmo de clasificación, se empleará el conocido como criterio de vecindad o distancia “K-Nearest-Neighbors”, un tipo de algoritmo estadístico basado en los ejemplos de entrenamiento. Al contrario que otros algoritmos, éste requiere menos tiempo de cómputo durante la fase de entrenamiento, pero requiere más tiempo en el proceso de clasificar una muestra. Está basado en el principio de cercanía de los ejemplos con características similares, siendo el algoritmo más básico el vecino más cercano (nearest neighbor). En este caso, el algoritmo se basa en los “K” vecinos más cercanos para predecir la especie de la muestra en cuestión.

El algoritmo consiste en determinar un espacio de características n-dimensional (n características) en el cual se encuentren todas las muestras de entrenamiento, todas ellas etiquetadas con su especie correspondiente. Durante el proceso de clasificación de una nueva muestra, ésta se proyecta en dicho espacio n-dimensional y se obtiene la distancia relativa entre la muestra “y” y cada uno de los ejemplos o muestras de entrenamiento “x”. Esta distancia se puede medir con diferentes métricas, como la distancia euclídea o el discriminante lineal de mínimos cuadrados.

El valor de “K” influye en gran medida en el resultado de la clasificación. Sin embargo, no existe un método que permita elegir un valor adecuado, salvo la realización de pruebas con distintos valores de “K”.

Desarrollo e implementación

En este proyecto se ha realizado un programa orientado a la clasificación de señales de audio. Se ha diseñado una herramienta que puede instalarse en cualquier sistema operativo que cuente con el software Matlab y que permite entrenar a un clasificador y analizar los resultados. Este programa se ha dividido en dos módulos, un primer módulo encargado de la extracción de características de los distintos audios y un segundo módulo que realizará una representación bidimensional de las características extraídas y procederá al cálculo de la probabilidad de error del sistema.

Herramienta de clasificación

En este apartado veremos el funcionamiento básico de la herramienta, de que elementos se compone la interfaz, así como una breve introducción a los procesos que ejecuta la herramienta.

La herramienta está dividida en dos módulos independientes que realizan tareas distintas y es por eso que veremos cada uno por separado.

Módulo de Extracción de características

En la figura 2 se muestra la interfaz gráfica del programa desarrollado en MatlabR2016b. Esta interfaz pertenece al módulo de extracción de características y su función es la de extraer las características clave de los distintos audios en base a unos parámetros especificados por la interfaz.

Esta interfaz cuenta con tantas pestañas como especies vayamos a clasificar, en el caso de este proyecto serán 5 las especies a clasificar, siendo estas: gorrión común, jilguero europeo, mirlo común, paloma torcaz y mirlo.

Dentro de cada una de las pestañas deberemos introducir las rutas absolutas de los directorios donde se encuentran las bases de datos de audios de la especie en cuestión y los directorios donde las características de dichos audios extraídas serán guardadas. Por cada uno de los ficheros de audio de la base de datos se creará un fichero .mat en el directorio de resultados.

La interfaz cuenta también con una serie de campos que podremos modificar y que serán parámetros que afecten a la extracción de características del audio. En cada

pestaña podremos indicar el rango de frecuencias que nos interesan para la especie en cuestión, de forma que la herramienta extraiga las características de solo ese rango de frecuencias del audio y así evitemos posibles interferencias de otros ruidos ambientales o incluso de otras especies. También en cada pestaña encontraremos un campo llamado "Tag", que será la etiqueta que le pondremos a las características extraídas y que se utilizaran posteriormente en el módulo de clasificación.

Por último encontramos los campos de "Sampling frequency" y "Number of Mel filterbanks", que serán comunes a todas las especies. El primer campo indica la frecuencia de muestreo con la que se leerán los audios y servirá también para el calculo de las tramas de análisis de los MFCC como veremos más adelante. El segundo campo corresponde al número de bancos de filtros de Mel que emplearemos y es un valor crucial para el computo de los MFCC ya que indica cuantos coeficientes extraeremos de cada audio.

El botón "Feature Extraction" ejecutará el módulo de extracción de características pasando como argumentos los campos que vimos en la interfaz. Se extraerán las características de todos los audios de los directorios que se indicaron en base a estos parámetros y por último se almacenarán estas características en los directorios indicados.

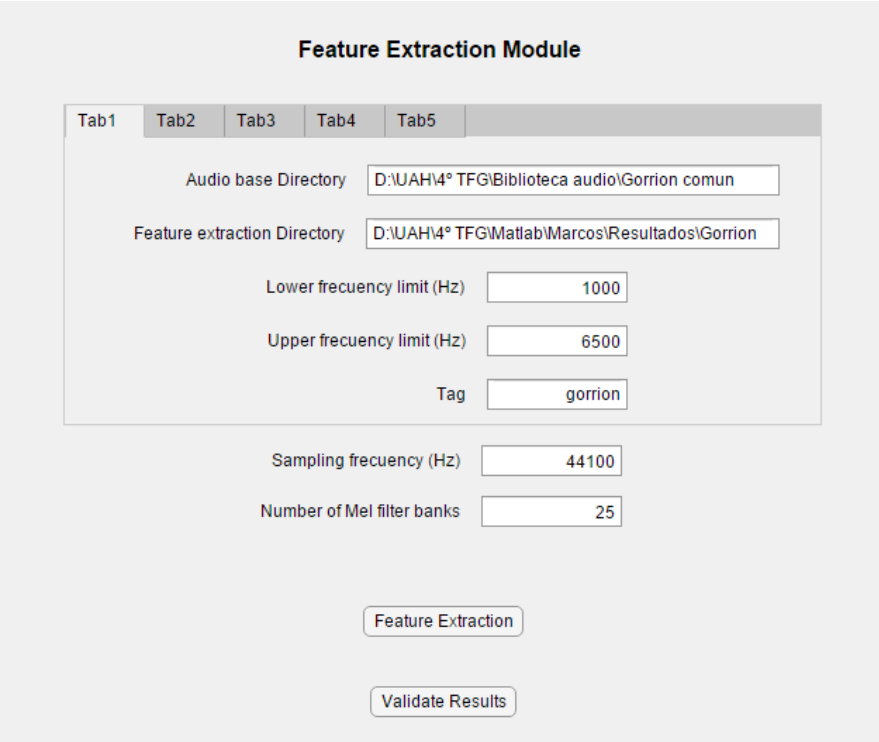


Figura 2. Interfaz módulo de extracción de características.

Una vez extraídas las características de todos los audios se procederá a hacer click en el botón de “Validate Results”, que abrirá una nueva ventana con el módulo de clasificación.

Módulo de Clasificación

Dentro de esta interfaz encontramos una primera parte dedicada a la representación bidimensional de los coeficientes MFCC, donde podremos seleccionar como ejes los coeficientes que elijamos para poder ver la agrupación en forma de scattering de las distintas características de los audios. Esta representación bidimensional nos permitirá ver a simple vista como se agrupan las distintas características y así poder escoger con más cuidado los coeficientes con los que queremos clasificar los distintos audios.

La segunda parte de la interfaz del módulo de clasificación estará dedicada a la obtención de las distintas probabilidades de error en función de con qué parámetros contemos. Se podrá escoger el número de coeficientes MFCC que serán comparados, así como el número de vecinos con los que se compararán las muestras. Estos parámetros son críticos para determinar la probabilidad de error, por lo que se permite seleccionar varios y la herramienta realizará simulaciones con todos ellos, dándonos una estimación de la probabilidad de error.

Será posible indicar además del número de coeficientes y de vecinos, si se incluye o no la media y desviación estándar del pitch de los audios, así como la media y desviación estándar de los máximos de energía de estos.

Por último, se introducirá la ruta absoluta de un fichero .txt donde se guardarán los resultados de estas simulaciones. Si el fichero ya existe será sobrescrito y si no será creado. Este fichero almacenará las distintas probabilidades de error obtenidas en las simulaciones acompañadas de los parámetros que se emplearon en su obtención.

Los resultados obtenidos de las simulaciones nos darán una idea del funcionamiento de la herramienta y serán estas probabilidades de error las que caractericen al sistema.

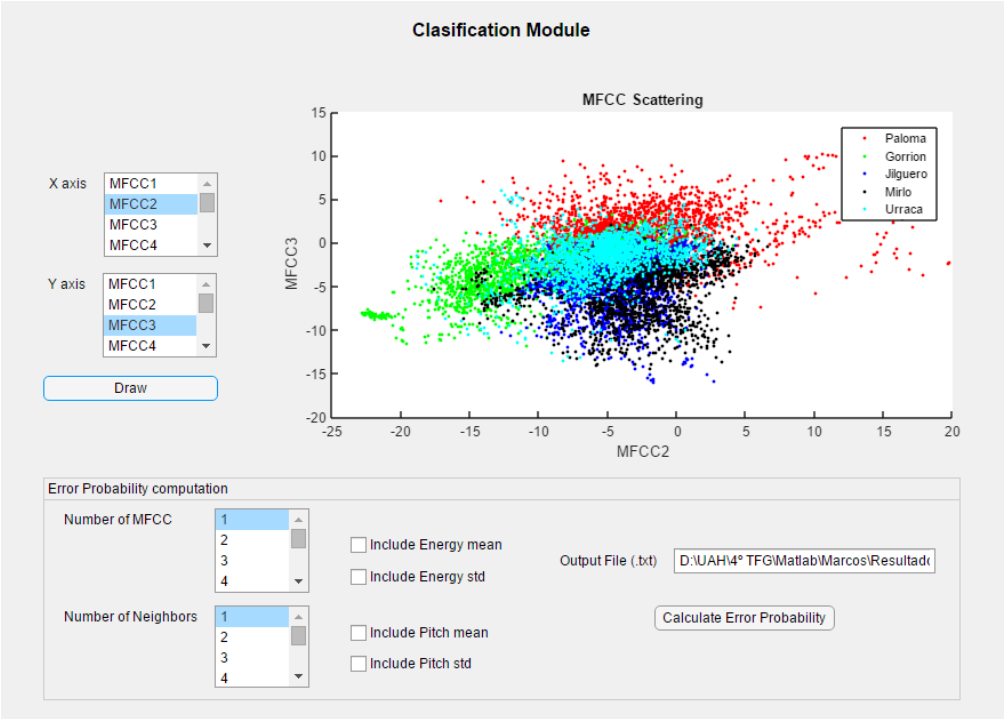


Figura 3. Interfaz del módulo de clasificación.

Ejecución y funcionamiento

En este apartado se verá en más detalle la parte técnica del funcionamiento de la herramienta.

Módulo de Extracción de características

Este es el primer módulo llamado al arrancar la aplicación y como ya se comentó anteriormente es el encargado de extraer las características de una base de datos de audio a partir de unos parámetros especificados por la interfaz.

Una vez completados todos los campos se procede a pulsar el botón de "Feature Extraction" o extracción de características. Este botón llama a la función "FeatureExtractionButtonPushed", la cuál se encarga de extraer las características de los distintos audios en base a los campos previos. La forma en la que la herramienta extrae las características es la siguiente:

La aplicación buscará en primer lugar los distintos archivos .mp3 que encuentre en el directorio indicado, los cargará al sistema e irá informando por consola según estos sean cargados al espacio de trabajo de la herramienta.

Cada audio será leído junto con su frecuencia de muestreo y si esta es distinta de la frecuencia que se indico en la interfaz se remuestreará la señal, de esta forma nos aseguramos de que todos los audios sean tratados igual independientemente de su frecuencia de muestreo original. Estos audios una vez cargados al espacio de trabajo de la herramienta serán tratados como vectores, y en este punto, hay que tener en cuenta que los distintos audios podrían estar compuestos por una o dos pistas, en cuyo caso nos quedaríamos solo con una de ellas.

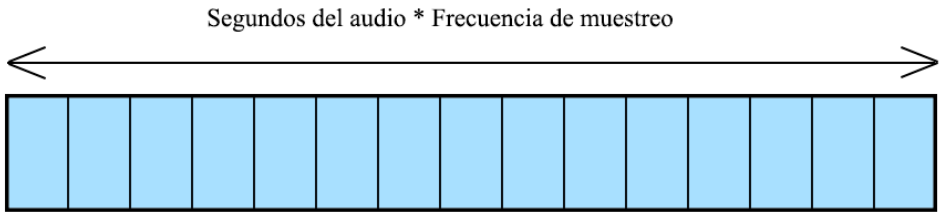


Figura 4. Vector audio.

Como veremos más adelante en el módulo de clasificación, no nos interesan los valores instantáneos de las distintas características del audio, si no su media y desviación

estándar, es decir, como varían estas características en el tiempo, ya que en un momento dado las características de dos especies distintas podrían ser muy similares y es gracias a la observación de su variación que somos capaces de diferenciarlas.

Debido a que los vectores audio pueden ser de duración indeterminada, dividimos estos en tramas de 5 segundos, ya que no resultaría interesante extraer la media de las características de un audio largo, que podría variar mucho en el tiempo y la media de estas características no reflejaría dichos cambios. La herramienta almacenará en una matriz el audio completo, de forma que cada fila de la matriz se corresponde a 5 segundos del audio. De esta forma en vez de tratar con vectores audio de longitud indeterminada, la herramienta trabajará con matrices audio y extraerá características de cada fila, es decir, extraerá características por cada 5 segundos de audio, como veremos más adelante.

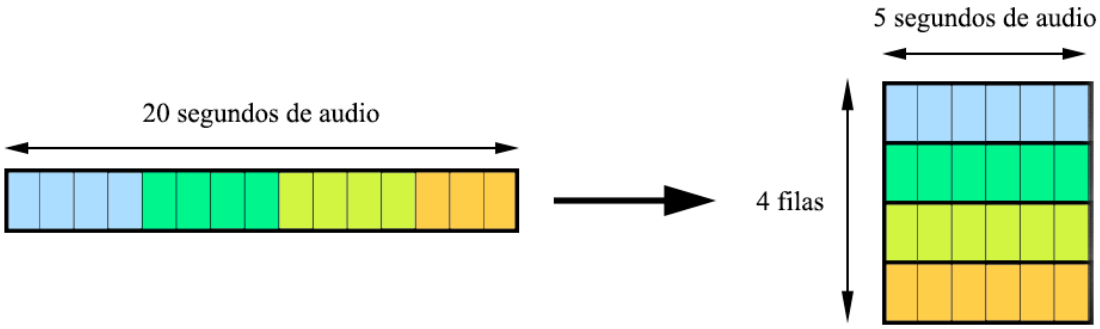


Figura 5. Matriz audio.

Una vez obtenida esta matriz audio, se procede a la extracción de características de cada fila de la matriz. Este proceso es llevado a cabo de la siguiente manera:

Dentro de un bucle que recorrerá las distintas filas de la matriz audio, se calcularán los coeficientes MFCC, el pitch y los máximos de energía de la señal. Una vez obtenidas estas características se calcularán sus medias y desviaciones estándar y se almacenaran estas en variables que irán acumulando los valores de las distintas tramas de 5 segundos, de modo que una vez procesado el audio completo tendremos una matriz que contendrá la media de los coeficientes MFCC del audio completo, otra matriz que contendrá sus desviaciones estándar y vectores que contendrán las medias y desviaciones estándar del pitch y de los máximos de energía del audio completo. Estas serán las variables que se almacenarán en el directorio de resultados en un fichero .mat.

A continuación veremos de que forma se obtienen estas características.

Cómputo de los MFCC

El cómputo de los coeficientes se lleva a cabo mediante la invocación de una función a la que se le pasan como argumentos los 5 segundos de audio de los que se extraerán los coeficientes, la frecuencia de muestreo, la duración de la trama de análisis en milisegundos y la duración del desplazamiento de tramas también en milisegundos, el coeficiente de preénfasis, el tipo de ventana que emplearemos para enventanar las tramas, el rango de frecuencias que seleccionamos en la interfaz, el número de bancos de filtros de Mel y el número de coeficientes MFCC que queremos obtener.

En la primera parte de la función realizamos una serie de cálculos preliminares. Entre ellos nos aseguramos de que el número de inputs de la función sea correcto, si el valor máximo de la señal de audio es igual o inferior a uno lo multiplicamos por 2^{15} para así explotar las muestras al rango de 16 bits, calculamos el número de muestras que entran en una trama de análisis "Nw" y el desplazamiento de tramas "Ns". Definimos también la longitud de la transformada rápida de Fourier (aunque en el marco teórico nos referíamos a la transformada discreta de Fourier, en la aplicación real se emplea la transformada rápida) y la longitud de las partes únicas de esta transformada (la mitad más uno).

El siguiente bloque de la función lo componen los handles de funciones que nos serán útiles a lo largo de la ejecución de esta función. Estos son la transformación del dominio espectral al dominio de Mel y viceversa y la transformada de coseno discreta de tipo II.

El tercer bloque de esta función será el que extraiga las características del audio, en concreto los coeficientes MFCC, las energías de los bancos de filtros y las tramas enventanadas.

El primer paso en el procesado del audio para la extracción de sus características será someter la señal a un preénfasis, lo que equivale a un filtro tipo FIR de primer orden con coeficiente de preénfasis "alpha" para realzar la alta frecuencia. Este realce proporciona una mejor relación señal ruido (SNR), además de introducir aproximadamente +6 dB/octava para compensar la atenuación de la alta frecuencia. La respuesta de este filtro tiene la siguiente forma:

$$H(z) = 1 - \alpha z^{-1} \qquad 0{,}9 \leq \alpha \leq 1 \qquad (2)$$

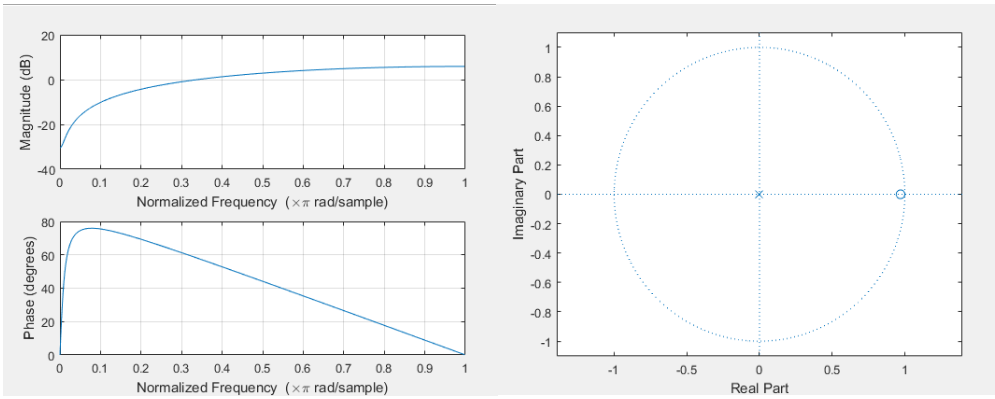


Figura 6. Respuesta en frecuencia del filtro de preénfasis.

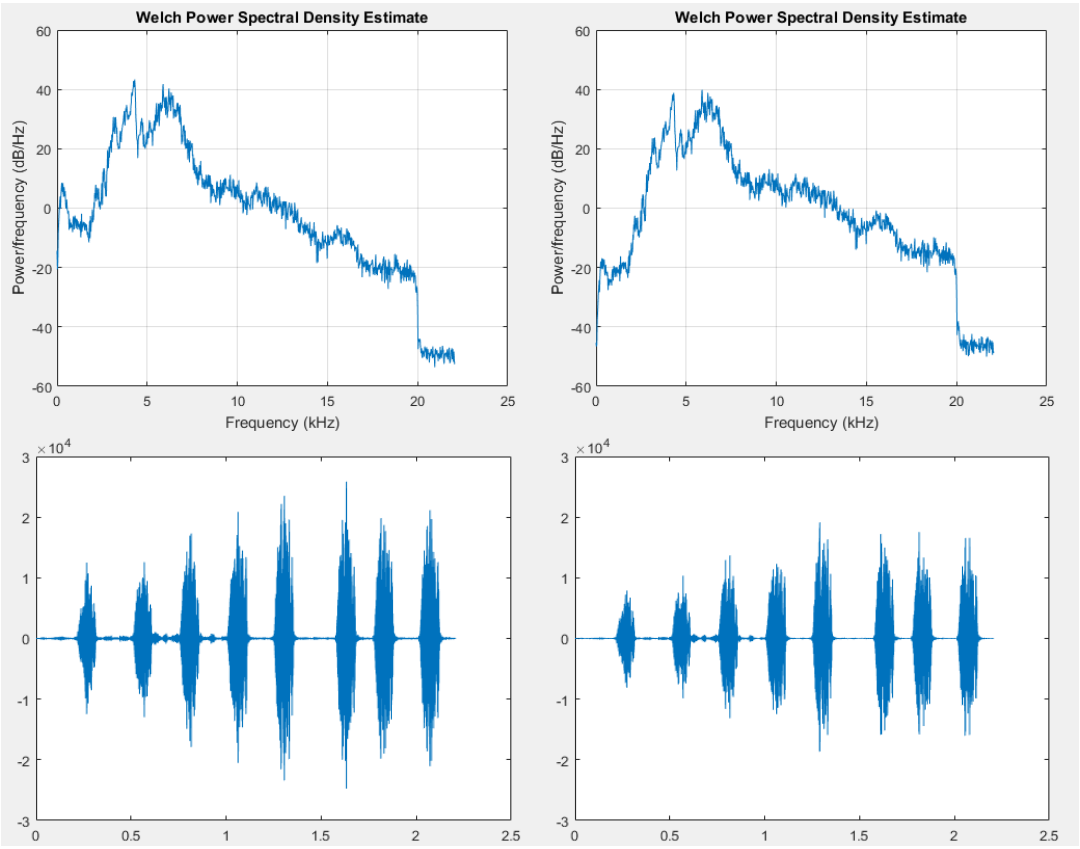


Figura 7. Ejemplo de señal a la izquierda, posterior al preénfasis a la derecha.

Posteriormente, se realiza un enventanado de la señal, dividiendo esta en tramas o frames de menor duración y con cierto solapamiento entre sí, aplicando una ventana (hamming, hanning…) a cada una de las tramas. En la siguiente figura se puede observar este proceso, dividiendo la señal de 5 segundos en bloques de tamaño “L” (longitud de la trama), con cierto solapamiento entre sí “O” (overlapping). A cada trama se le multiplica

por una ventana muestra a muestra (una ventana Hamming en nuestro caso, ya que introduce menos distorsión que otras ventanas), de forma que las muestras de los extremos de cada trama tengan menor peso que las muestras centrales, ya que, de lo contrario, se producirían efectos indeseados en alta frecuencia. Además, dado que existe solapamiento entre las tramas, no se tiene pérdida de la información por el efecto de atenuación de los extremos.

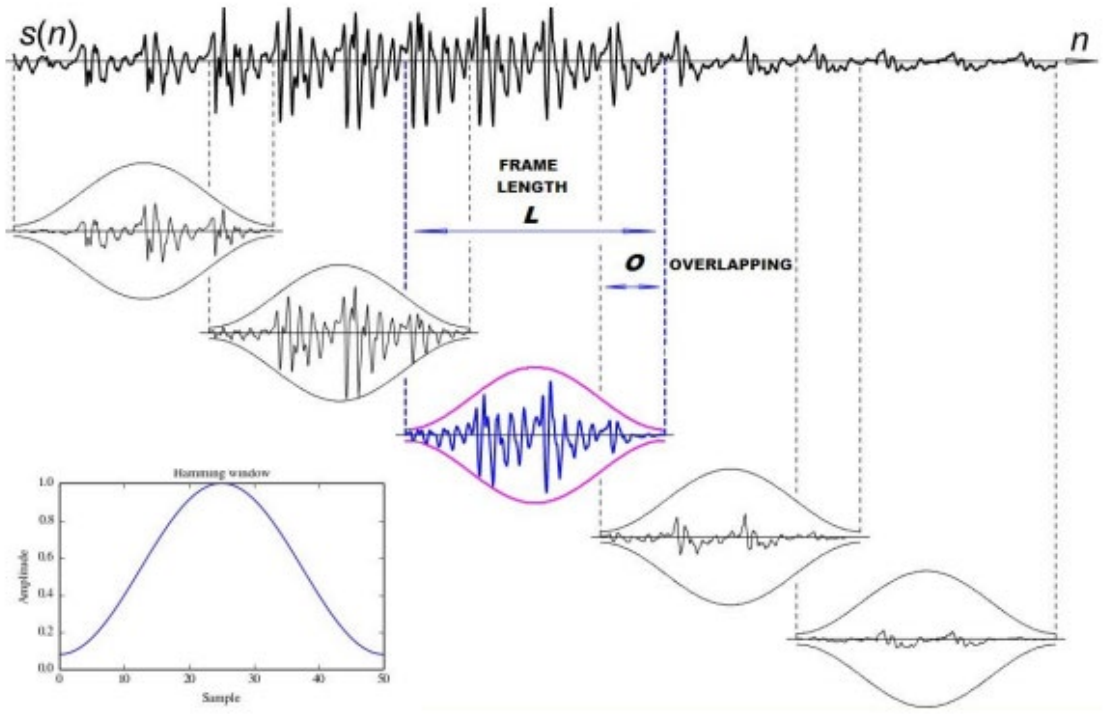


Figura 8. Enventanado y solapamiento entre tramas.

Estos frames se consiguen a través de la función “vec2frames”, que nos devuelve una matriz con los bloques enventanados.

Una vez obtenidos los frames se procede a calcular la magnitud del espectro (el módulo al cuadrado) obtenido mediante la transformada rápida de Fourier (FFT) con la longitud de la transformada calculada en los pasos preliminares.

A continuación se procede a través de la función “trifbank” a generar la matriz que contiene los “M” filtros triangulares, uno por fila. Estos filtros triangulares se encuentran uniformemente espaciados en la escala Mel definida por las funciones que transforman la frecuencia al dominio de Mel.

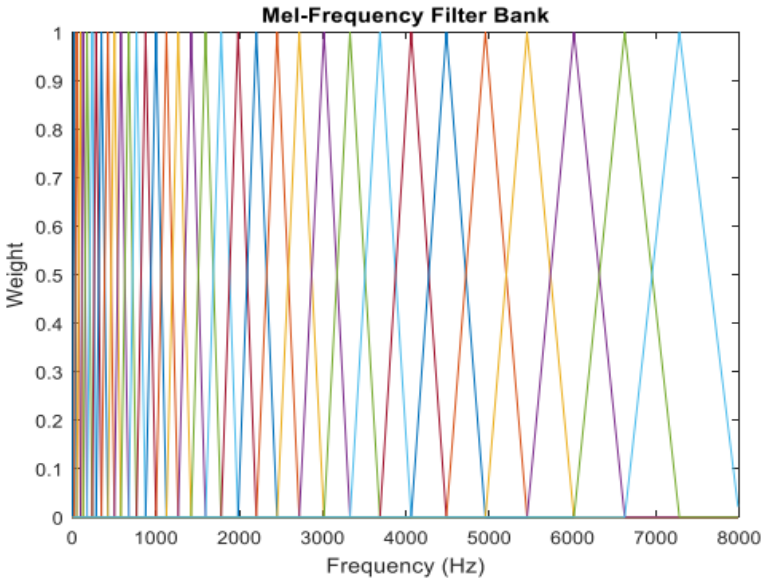


Figura 9. Banco de filtros de Mel.

Una vez obtenidos los bancos de filtros se procede al cálculo de las energías de los bancos de filtros, esto se consigue mediante el producto escalar de la matriz que contiene los filtros triangulares por la magnitud de la parte única del espectro obtenido anteriormente.

A continuación se genera la matriz de la transformada discreta del coseno (DCT) mediante la función que definimos en el segundo bloque de la función "mfcc" y que nos devolverá una matriz MxM. Y por último obtenemos los coeficientes MFCC aplicando la transformada discreta del coseno a los logaritmos de las energías de los bancos de filtros.

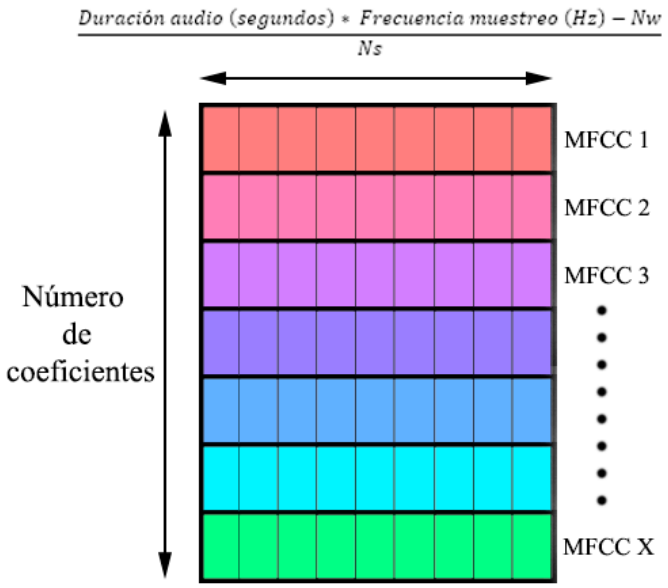


Figura 10. Matriz de coeficientes MFCC.

Una vez conseguida la matriz de coeficientes MFCC de cada 5 segundos de audio, se calcula la media y desviación estándar de cada uno de los coeficientes y se almacenan en dos nuevas matrices (una contiene la media mientras que la otra contiene la desviación estándar). Estas matrices contendrán en cada columna la media y desviación estándar de cada uno de los coeficientes para cada 5 segundos del audio completo.

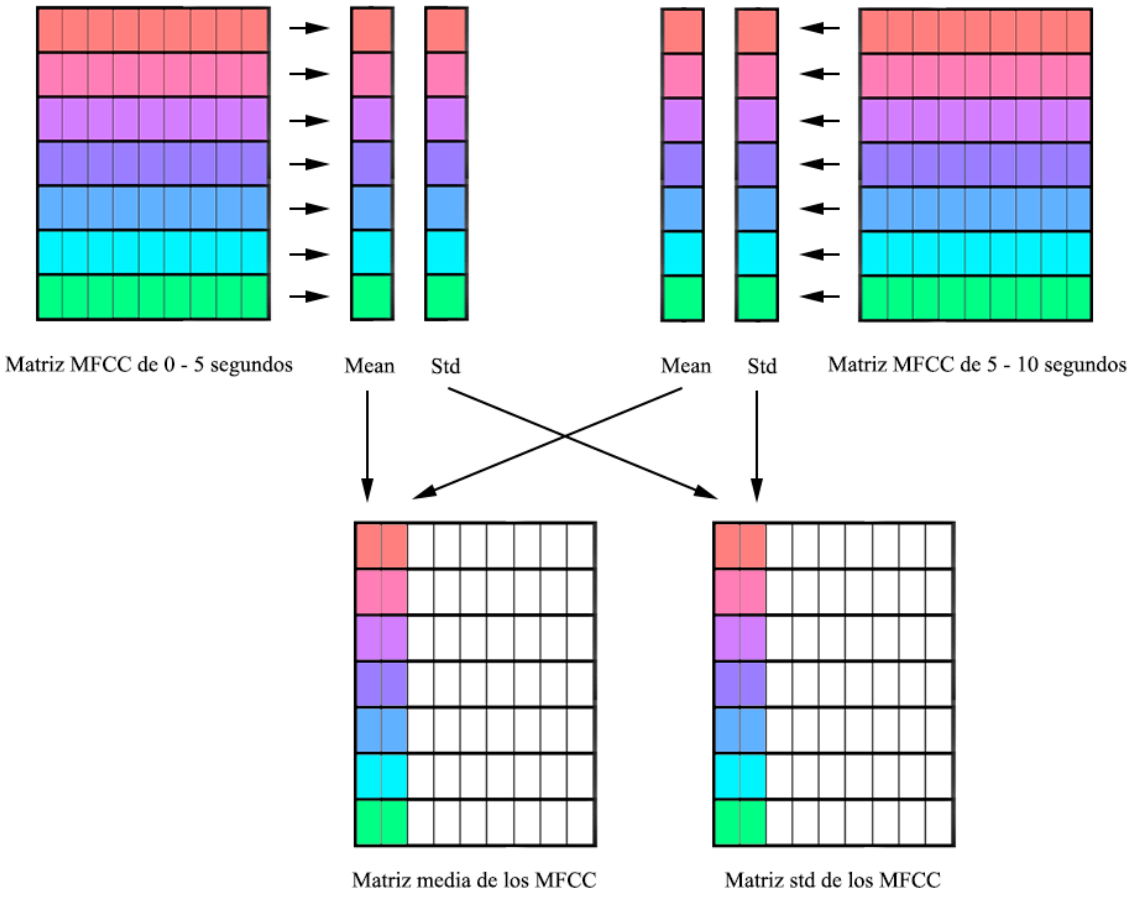


Figura 11. Cálculo de las matrices media y desviación de los coeficientes.

Cálculo del Pitch

El cálculo del pitch se lleva a cabo mediante la función “ExtraerPitch”, la cual toma como parámetros 5 segundos de audio de la matriz de audio y la frecuencia de muestreo de dicha señal y devuelve la media y la desviación estándar del pitch.

El primer paso para el calculo del pitch es remuestrear la señal a 8 KHz en caso de que la frecuencia de muestreo que se pasó como parámetro sea distinta ya que no necesitamos resoluciones mayores que esa para el cálculo del pitch.

A continuación se divide la señal de audio en pequeñas tramas de 20 milisegundos y se procede al cálculo de los coeficientes de predicción lineal. Estos coeficientes se calculan de la siguiente forma:

En primer lugar se calcula la autocorrelación de la trama de audio y se crea una matriz Toeplitz con dichos valores y un vector que contiene los términos independientes. Los coeficientes de predicción lineal se obtendrán del producto escalar de la inversa de la matriz de Toeplitz por el vector de términos independientes.

Estos coeficientes suponen el denominador del filtro H(z) que corresponde al modelo del tracto vocal, que es un filtro todo polos. El numerador de dicho filtro corresponde a la ganancia y la forma del filtro resultante sería la siguiente:

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{3}$$

Con el filtro LPC inverso y la señal de audio podemos obtener la señal de error, que además corresponde con la excitación del modelo de producción de voz.

Una vez obtenida esta excitación calculamos su autocorrelación y eliminamos las primeras muestras de esta, ya que queremos ver donde se produce el máximo de la autocorrelación y de no hacerlo siempre ocurriría el máximo en la primera muestra, que corresponde con la energía de la señal. Una vez obtenido el máximo de la autocorrelación, si este supera un 20% de la energía de la señal de excitación, se determinará que se trata de un sonido sonoro y se procede al calculo de su periodo o pitch, que será el índice de la muestra donde se detectó un máximo de la autocorrelación. De no superar un 20% de la energía de la señal, se entiende que se trata de un sonido sordo y por tanto su periodo será 0.

Este proceso se repite con las distintas tramas de 20 milisegundos y se almacenan en un vector. Una vez procesadas todas las tramas se procede al cálculo de la media y de la desviación estándar de dicho vector, que serán las variables que devuelva la función.

Cálculo de los máximos de energía

Para el cálculo de los máximos de energía se procede de forma similar a las funciones anteriores. Es la función "ExtraerMaximosEnergia" la que se encarga de devolver la media y la desviación estándar de los máximos de energía de la señal.

Esta función recibe por parámetros 5 segundos de audio de la matriz de audio y la frecuencia de muestreo de la señal. En primer lugar divide la señal de audio en pequeñas tramas de 20 milisegundos y para cada una de ellas obtiene el módulo del espectro.

Se almacena en un vector los máximos de energía encontrados en las distintas tramas y posteriormente se calculan su media y desviación estándar, que serán las variables que devuelva la función.

Módulo de Clasificación

Una vez extraídas todas las características de los distintos directorios, se procede a pulsar el botón de "Validate Results", el cual lanzará el módulo de clasificación.

Este módulo esta dividido en dos bloques. Uno primero dedicado a la representación bidimensional de los coeficientes MFCC en forma de scattering, donde podremos seleccionar que coeficientes queremos representar, lo que nos permitirá hacernos una idea de como se agrupan las características de distintas especies. Y un segundo bloque dedicado al cómputo de la probabilidad de error del sistema, donde podremos indicar una serie de parámetros y la herramienta realizará simulaciones con todos ellos, almacenando las distintas probabilidades de error obtenidas así como los parámetros empleados para su calculo en un fichero de texto en el directorio que seleccionemos.

Al arrancar este módulo se inicializan algunas variables con los valores que empleamos en el módulo de extracción de características, como el número de coeficientes que calculamos, para así reflejar en la interfaz gráfica los distintos valores que podemos seleccionar. También en el arranque de este módulo se comprueba si existe un fichero .mat que contiene todas las características extraídas de las distintas especies. De existir dicho fichero será cargado al espacio de trabajo de la herramienta y de lo contrario se cargarán uno por uno los ficheros de todas las especies y se creará un fichero .mat que agrupe todos estos.

Una vez ha arrancado el módulo ya se habrán cargado los distintos archivos necesarios para su funcionamiento y podremos ejecutar cualquiera de los dos bloques independientemente.

El primero de los bloques como se ha comentado con anterioridad, esta dirigido a la representación bidimensional de los coeficientes MFCC. Vemos que la interfaz nos proporciona dos listas que contienen los distintos coeficientes y donde podremos seleccionar que coeficiente queremos asignar a cada eje. Una vez seleccionados pulsaremos el botón “Draw” que invocará a la función “DrawButtonPushed”.

Esta función concatenará las matrices de los coeficientes MFCC de cada especie y también creará un vector que servirá para indicarnos a que especie corresponde cada una de las columnas de dicha matriz, esto se hará a través de la etiqueta o tag que creamos en la interfaz del módulo de extracción de características.

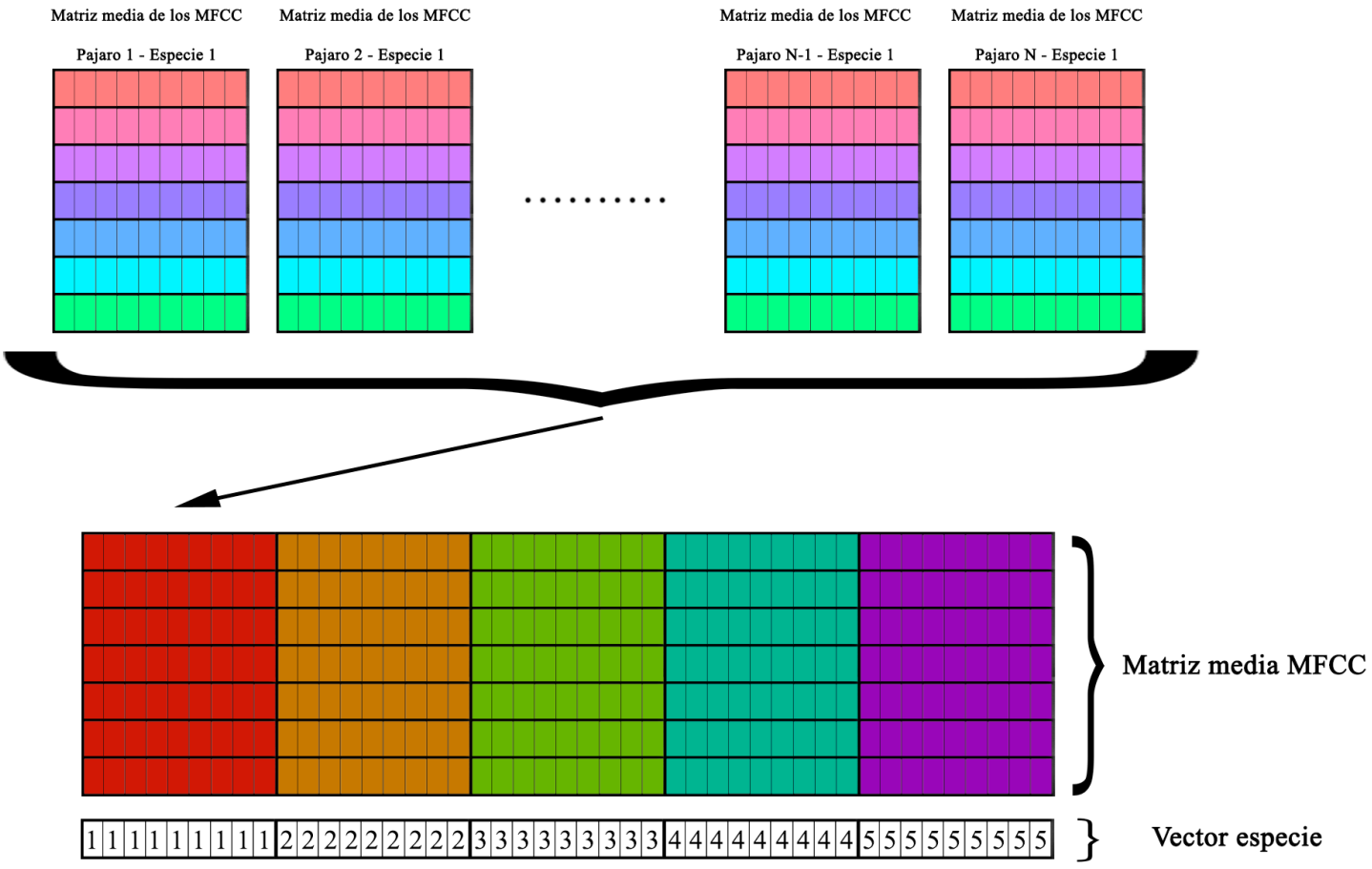


Figura 12. Concatenado de matrices de las diferentes especies.

Una vez tenemos la matriz con los distintos valores de los coeficientes y el vector que nos indica a que especie pertenece cada uno de esos valores, la herramienta procede a dibujar en la interfaz los distintos valores de los coeficientes seleccionados, empleando un color distinto para cada especie, acompañados de una leyenda que nos indicará que color corresponde a cada especie.

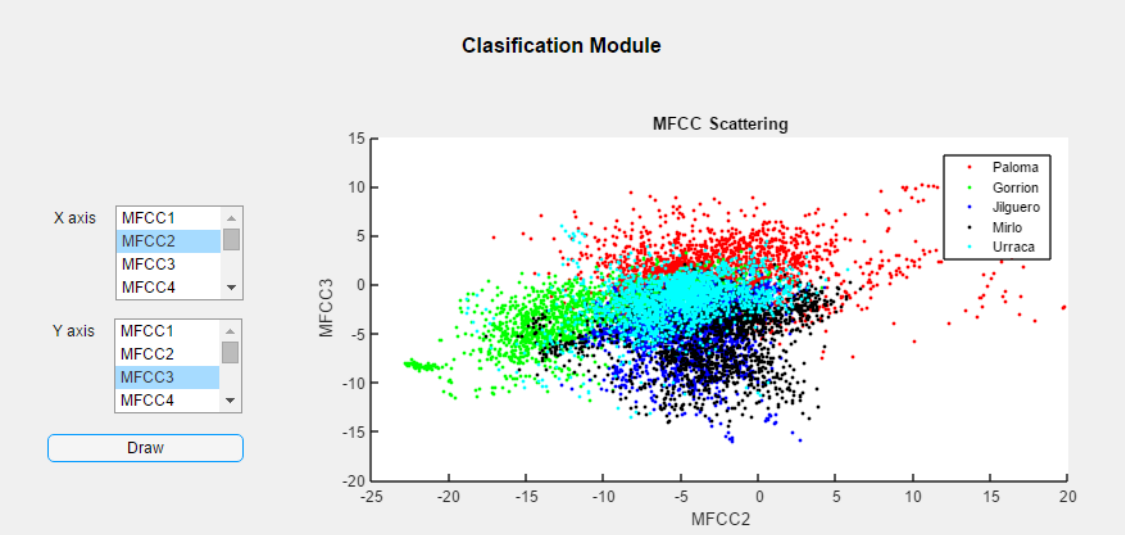


Figura 13. Ejemplo de representación de los coeficientes.

El segundo bloque, dedicado al cálculo de la probabilidad de error del sistema, es algo más complejo. Se seleccionarán a través de la interfaz que coeficientes queremos tener en cuenta para el cálculo, así como unos checkbox que nos permitirán incluir otras variables como el pitch o los máximos de energía de la señal.

Una vez hemos seleccionado con que parámetros deseamos realizar las simulaciones, especificamos el directorio y archivo de texto donde queremos almacenar los resultados de estas simulaciones y pulsamos el botón de “Calculate Error Probability”, que invocará la función “CalculateErrorProbabilityButtonPushed”.

Esta función, de forma análoga a la función de dibujar, concatenará las matrices de los coeficientes MFCC de cada especie, solo que además en esta función también concatenaremos los vectores que recogen los valores de la media y desviación estándar del pitch y de los máximos de energía de la señal de cada especie.

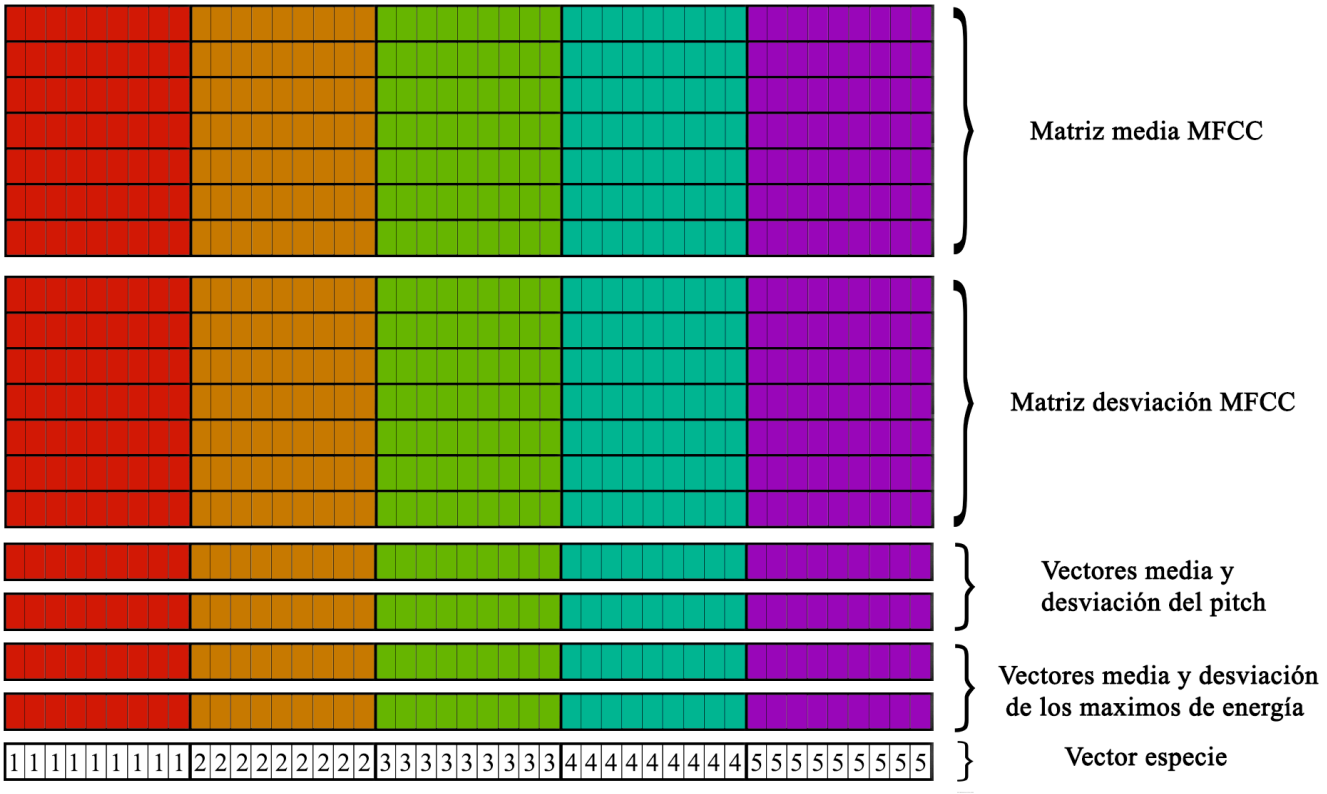


Figura 14. Matrices y vectores de características.

Una vez obtenemos esta matriz y estos vectores, procedemos a dividir estos entre el grupo de entrenamiento y el grupo de evaluación. La forma en la que se dividen estos es en base a una variable que indica que porcentaje de los datos va destinado a entrenar al sistema y que porcentaje esta destinado a comprobar el funcionamiento real de este. El reparto se hace de forma que el primer porcentaje de los datos va destinado a un grupo y el resto al otro, ya que si por ejemplo, repartiésemos alternadamente los datos, podría ocurrir que evaluásemos el funcionamiento de la herramienta con unos datos muy similares a los empleados para entrenar al sistema, dando lugar a unos falsos resultados favorables.

Se crean dos nuevas estructuras, una de diseño y otra de test, que estarán compuestas por una matriz que contendrán los coeficientes, el pitch y los máximos de energía de la señal y un vector que nos indicará a que especie pertenecen cada una de las columnas de dicha matriz. En resumen, estas dos estructuras contendrán todas las características extraídas de todos los audios.

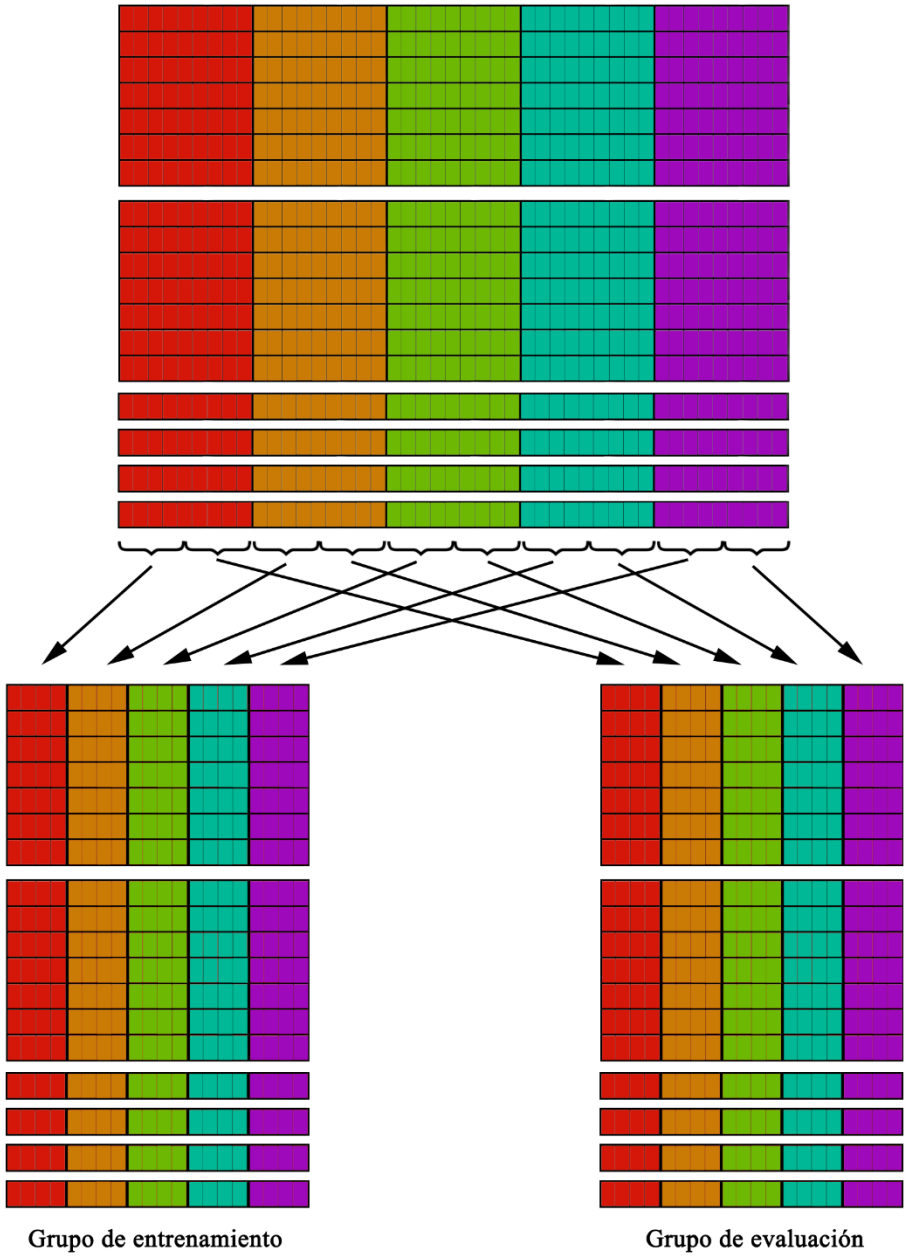


Figura 15. Reparto entre el grupo de entrenamiento y evaluación.

Posteriormente, en base a que parámetros fueron seleccionados en la interfaz, se crean unas matrices que contienen solo los datos que hemos seleccionado, en lugar de contener todos los datos (se copian las filas seleccionadas).

Con estas matrices ya preparadas, podemos proceder a realizar las distintas simulaciones. La herramienta irá iterando los distintos valores seleccionados y para cada uno de ellos empleará la función “knnsearch” para determinar a que especie pertenecen las características del grupo de evaluación.

El funcionamiento de esta función “knnsearch” es el siguiente: compara una a una las distintas columnas de la matriz de evaluación con la matriz entera de entrenamiento y almacena en una variable los distintos vecinos más cercanos. El número de vecinos más cercanos se indica también a través de la interfaz, ya que no hay forma de calcular el número óptimo de vecinos con los que comparar más que de forma empírica. Se empleará la distancia euclídea para determinar las columnas, y por tanto, las especies, más cercanas a la columna a evaluar.

Una vez conocidos los vecinos más cercanos a la muestra a evaluar, se realiza un histograma para hacer un recuento del número de especies vecinas, siendo la especie que se repite más veces la que la herramienta toma como resultado.

Estas decisiones se almacenan en un vector, que se comparará con el vector que contiene a que especie pertenece cada muestra, de esta forma podremos saber el número de fallos y de aciertos del sistema y podremos así calcular la probabilidad de error de la herramienta, la cual será almacenada en el fichero de texto de resultados que se indicó por la interfaz.

Simulaciones

En este apartado veremos los pasos a seguir para la obtención de resultados a partir de la herramienta desarrollada en este trabajo así como los resultados de las simulaciones llevadas a cabo durante el mismo. Introduciremos de nuevo la herramienta pero esta vez no desde el punto de vista técnico si no desde el punto de vista del usuario.

Cuando arranca la aplicación se nos presenta una interfaz gráfica como la que podemos observar en la siguiente figura.

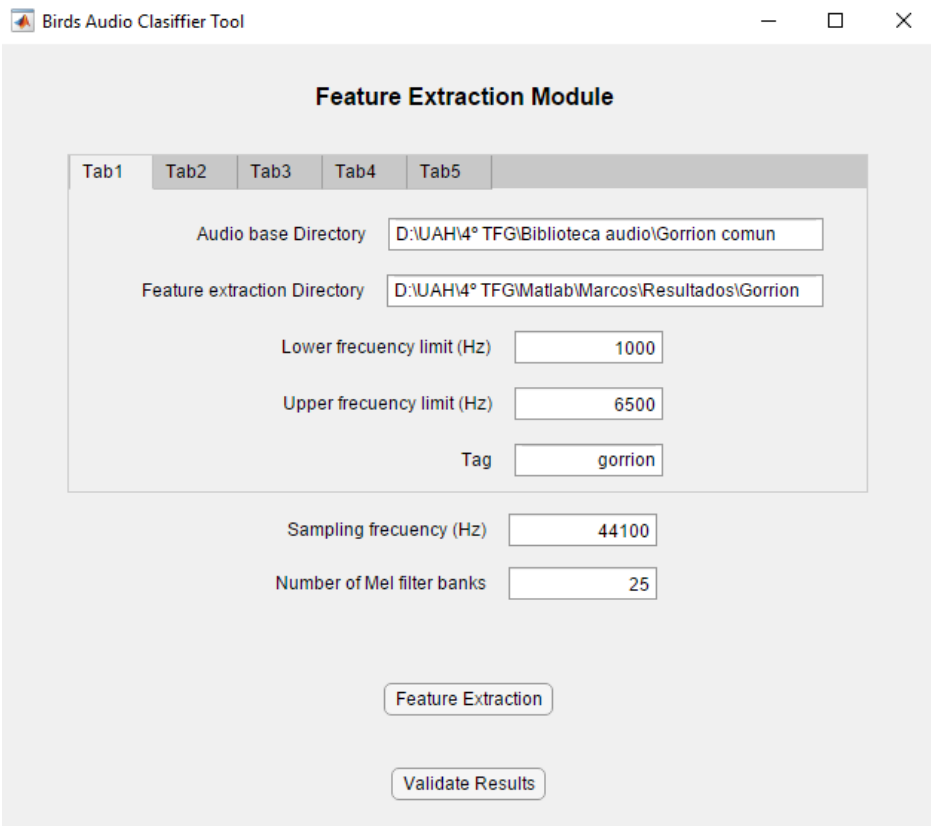


Figura 16. Interfaz del módulo de extracción de características de la herramienta.

En las simulaciones realizadas para este trabajo hemos usado como especies de prueba las siguientes: gorrión común, jilguero europeo, mirlo común, paloma torcaz y urraca común. Estas han sido las especies elegidas ya que son especies representativas de nuestro país y porque los sonidos que estas producen son muy diferentes entre algunas de ellas y muy similares entre otras. Con esto pretendemos ver como es el funcionamiento de la herramienta en cualquier situación, no solo ante sonidos fácilmente diferenciables.

Los primeros pasos a la hora de realizar simulaciones es hacernos con una base de datos de audio de cada una de las distintas especies. En este trabajo se ha utilizado como principal base de datos la página web www.xeno-canto.org, que nos proporciona una cantidad razonable de material con el que trabajar, además de contar con ciertas facilidades como la clasificación de audios en función de la calidad de la grabación.

Una vez obtenidos suficientes audios para las distintas especies y tras haber almacenados estos en sus respectivos directorios se procede a rellenar los campos de "Audio base Directory" y "Feature extraction Directory", que son los directorios donde se encuentran almacenados los audios de la especie en cuestión y el directorio donde serán almacenadas las características extraídas, respectivamente.

El siguiente paso es establecer unas frecuencias límite para cada una de las especies, para lo que conviene hacer un estudio previo de las señales de estas especies para observar entre que rango de frecuencias se encuentra cada una.

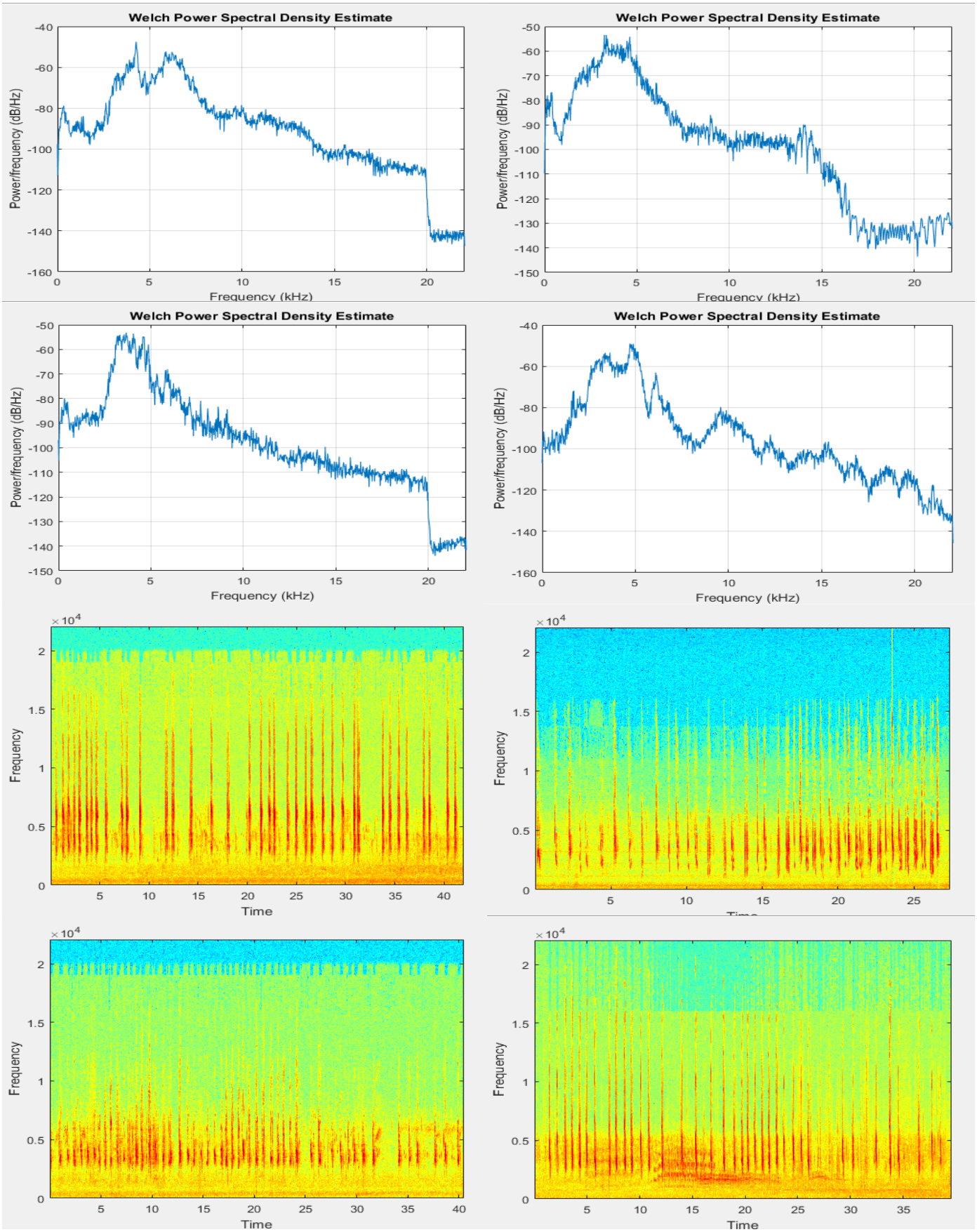
Estudio del espectrograma

Es común que la mayoría de los audios provengan de grabaciones realizadas en ubicaciones con ruido ambiental, ya sea en el campo o en una zona urbana, donde siempre existirán otros sonidos en la grabación además del de la propia especie. Para unos resultados lo más óptimos posibles, la manera de proceder sería limpiando los audios con un software de tratamiento de audio, como por ejemplo "Audacity", para así asegurarnos de que la grabación solo contiene aquellos sonidos de una especie en concreto.

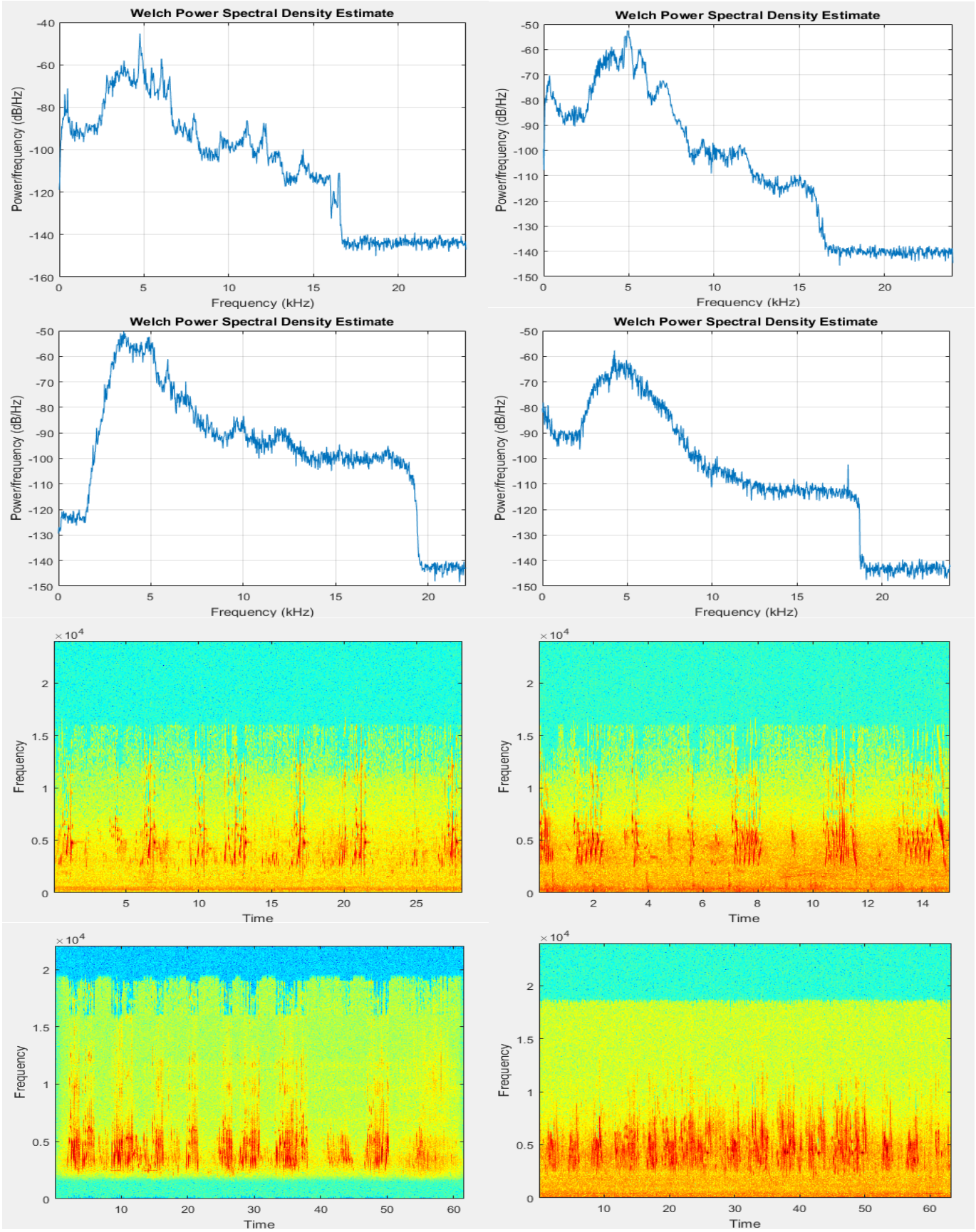
La tarea de limpiar los audios no resulta tan sencilla como pueda parecer, ya que no se puede automatizar, debido a que para ello haría falta un clasificador previo, lo cual sería una incongruencia. Por tanto es una tarea que solo puede llevarse a cabo por una persona y requeriría de largas horas de trabajo, teniendo en cuenta que podríamos contar con una base de datos de miles de audios de varios minutos cada uno.

Es debido a esto que se ha realizado un estudio previo de los espectrogramas y densidades espectrales de potencia de cada especie, para poder así observar cuales son las frecuencias más características en cada una y poder deshacernos del resto, mejorando los resultados de la herramienta sin tener que dedicarle horas al procesado previo de los audios.

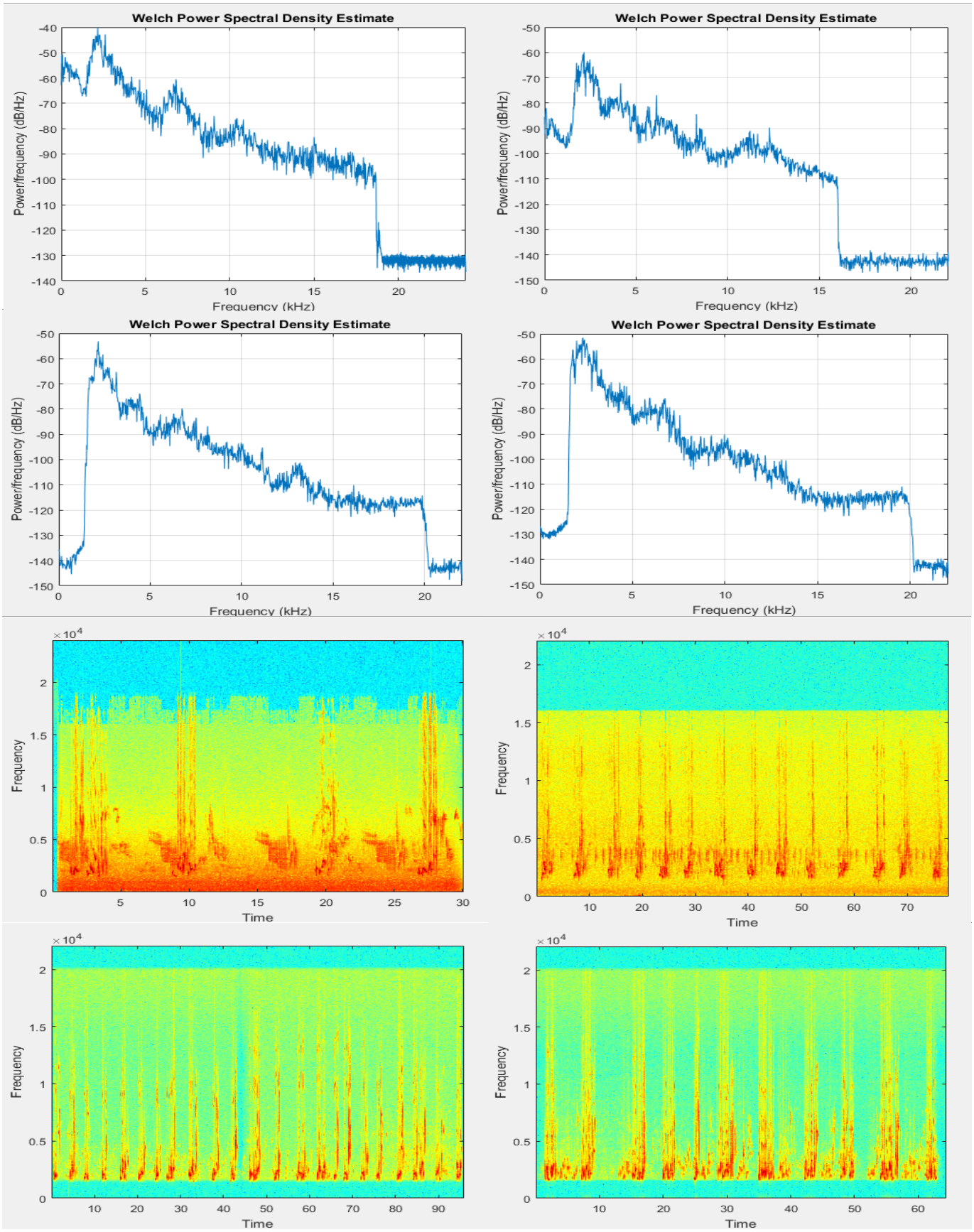
En el caso del gorrión común hemos podido observar que el rango de frecuencias más significativas se encuentra entre los 1000 y 7000 Hz gracias a la estimación de la densidad espectral de potencia.



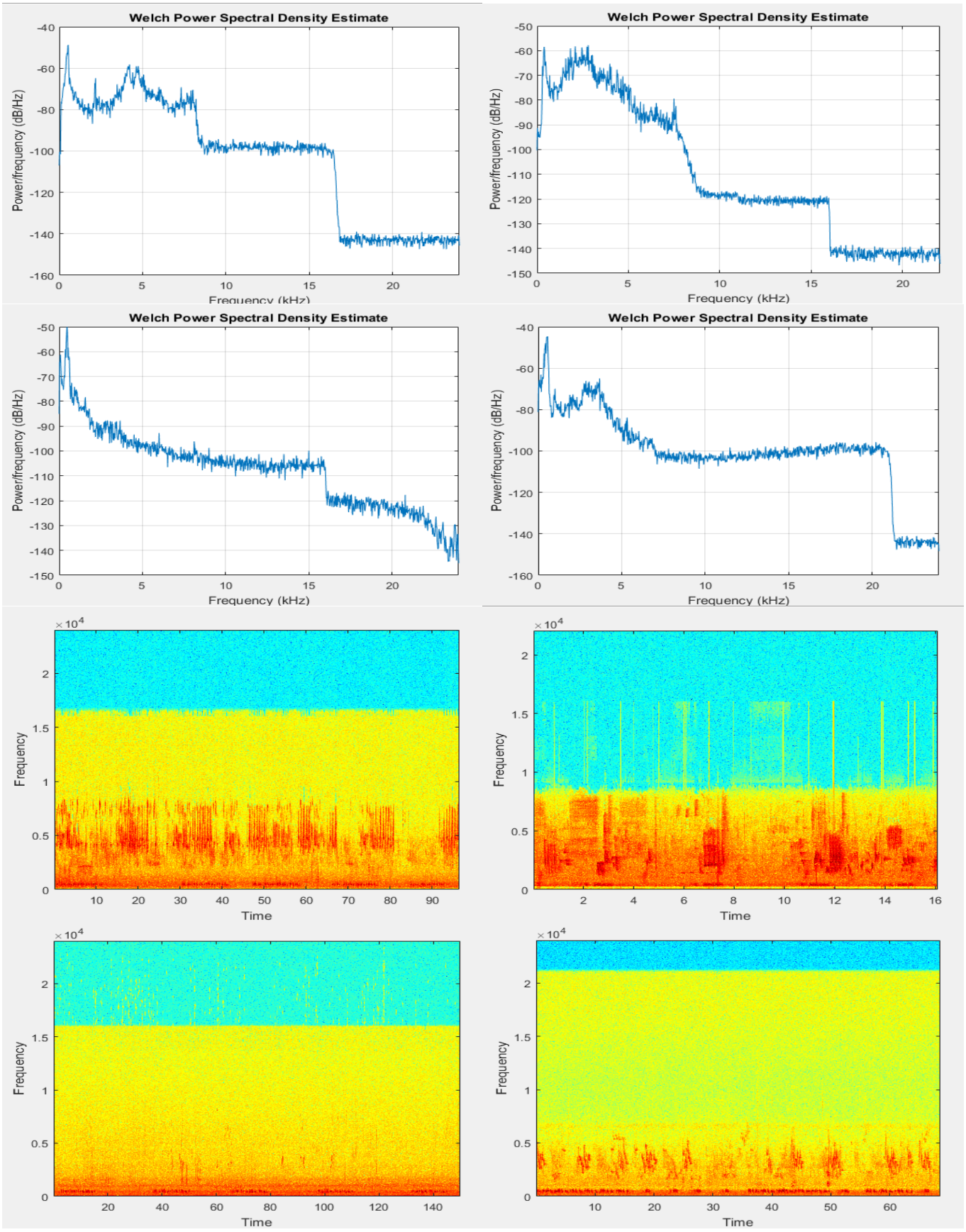
En el caso del jilguero europeo hemos observado que las frecuencias de más relevancia se encuentran en el rango de 2000 a 7000 Hz.



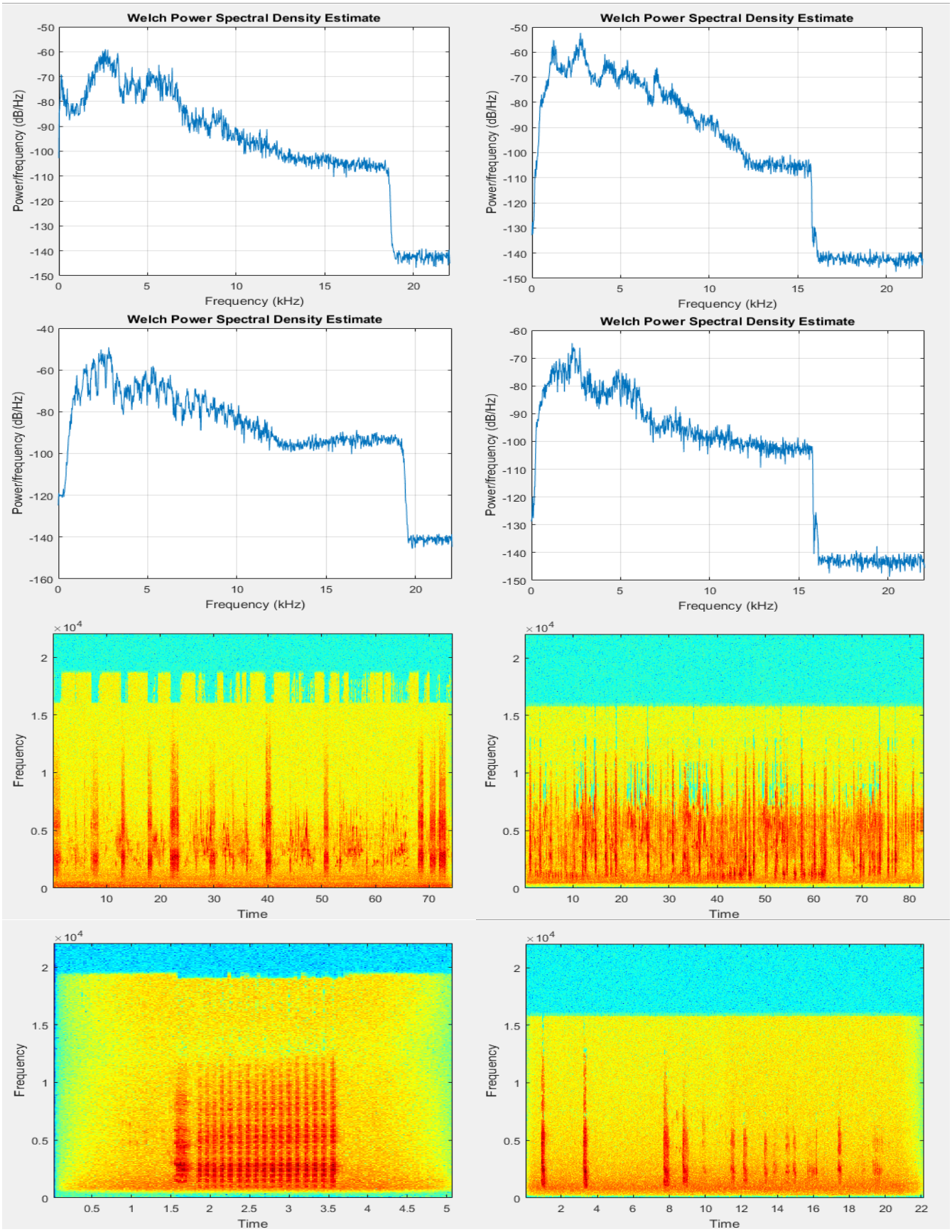
En el caso del mirlo común hemos observado que las frecuencias de más relevancia se encuentran en el rango de 1000 a 8000 Hz.



En el caso de la paloma torcaz hemos observado que las frecuencias de más relevancia se encuentran en el rango de 250 a 7000 Hz.



Y por último, en el caso de la urraca común hemos observado que las frecuencias de más relevancia se encuentran en el rango de 500 a 7000 Hz.



Tras seleccionar el rango de frecuencias más relevantes de cada especie procedemos a establecer la frecuencia que emplearemos para muestrear las señales de audio. Una frecuencia de muestreo alta devolverá un mayor número de muestras por audio, lo que supone aumentar la cantidad de datos con los que la herramienta cuenta para trabajar y se resume, en teoría, en unos mejores resultados.

Ahora, hay que tener en cuenta que los audios son remuestreados a la frecuencia de muestreo seleccionada en la interfaz, es por eso que seleccionar una frecuencia de muestreo alta podría suponer que se generasen muestras nulas intercaladas en aquellos audios que tuviesen una frecuencia de muestreo inferior a la indicada y, por tanto, esos audios podrían no ser representativos de la especie en cuestión. Por defecto la interfaz nos muestra una frecuencia de muestreo de 44100 Hz, que es la frecuencia de muestreo por defecto en CDs, aunque se ha observado que algunos de los audios empleados en este trabajo poseen una frecuencia de muestreo superior (48000 Hz) y otros emplean frecuencias de muestreo inferiores (hasta 16000 Hz).

Las simulaciones han sido realizadas con un valor por defecto de 25 filtros de Mel. Esto es debido a que el número de bancos de filtros indica el número de filtros triangulares que se encontraran equiespaciados entre las frecuencias dadas, es por eso que al aumentar el número de filtros estos se vuelven más estrechos y por tanto las características extraídas de estos variarán más que para un número de filtros menor y esto se ve reflejado en que la nube de puntos que representa las características de una especie sea más disperso, cuando lo apropiado sería tener una nube compacta y claramente definida, para así facilitar las labores de clasificación.

Resultados

Se han llevado a cabo diversas simulaciones para observar el funcionamiento de la herramienta ante diferentes escenarios que serán expuestos a continuación.

Simulación 1.1

Número de especies: 2 (Paloma y Gorrión)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Empleando únicamente los coeficientes MFCC

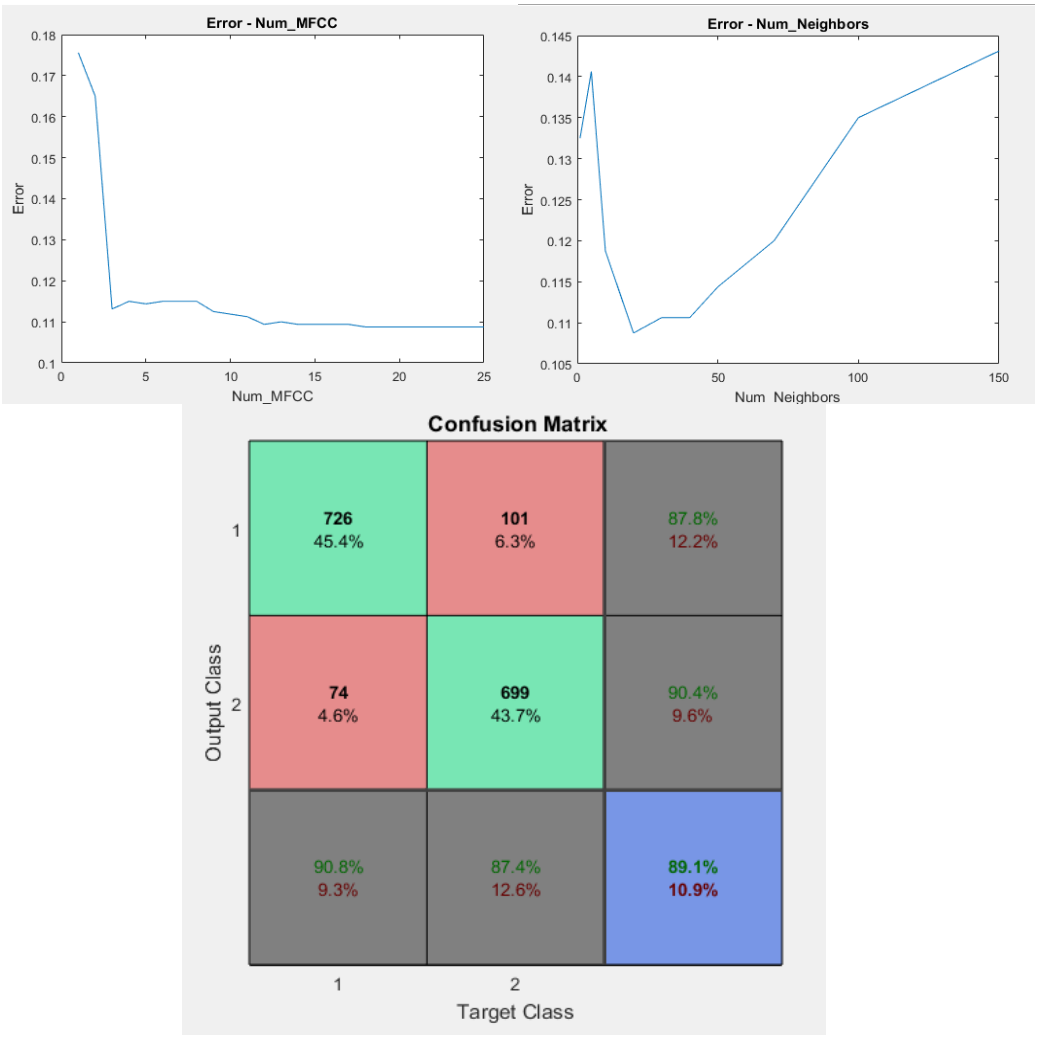


Figura 17. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 89,1 % de aciertos para 20 vecinos y 18 coeficientes MFCC.

Simulación 1.2

Número de especies: 2 (Paloma y Gorrión)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Incluyendo media y desviación estándar del pitch

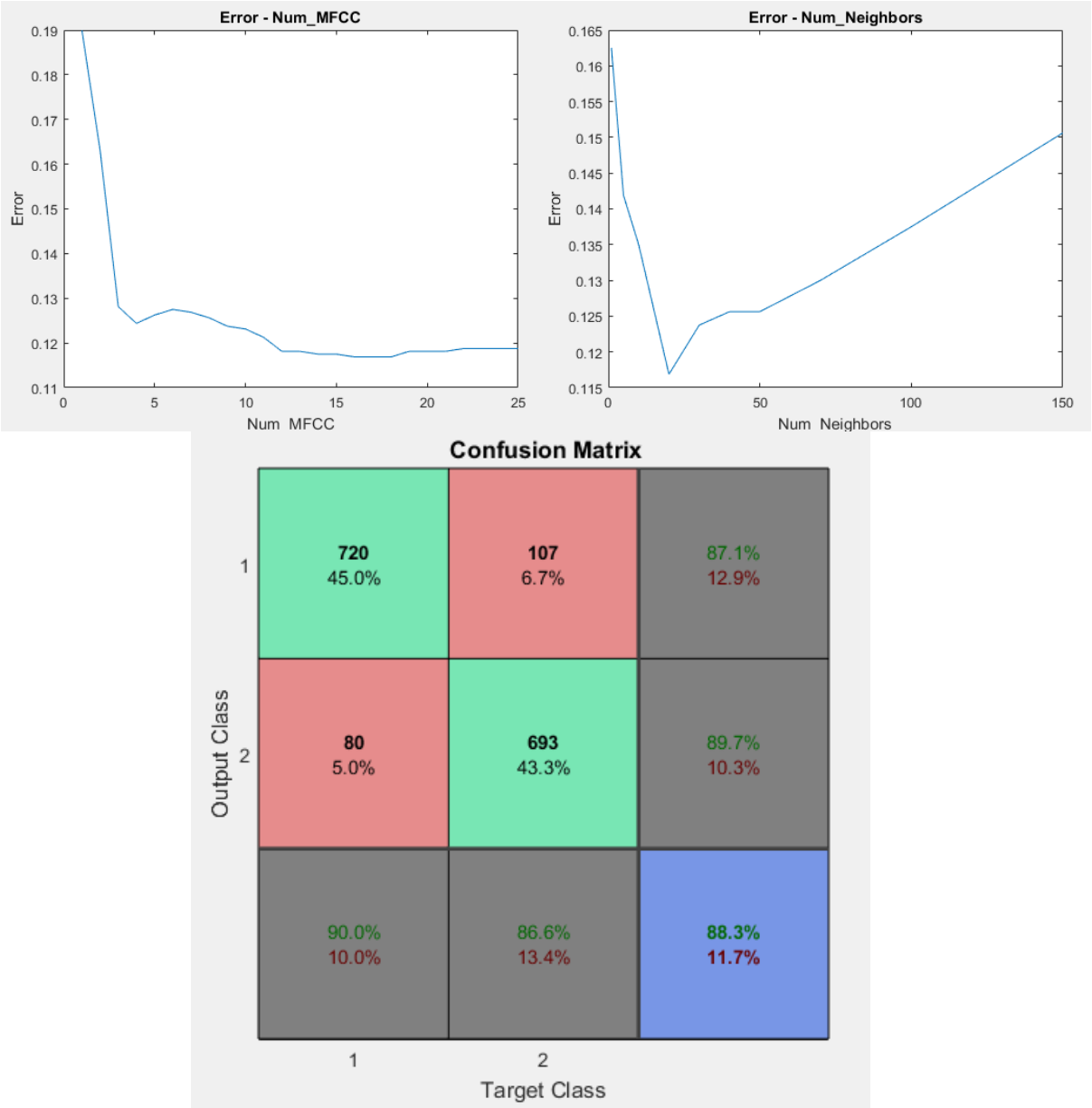


Figura 18. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 88,3 % de aciertos para 20 vecinos y 16 coeficientes MFCC.

Simulación 2.1

Número de especies: 3 (Paloma, Gorrión y Jilguero)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Empleando únicamente los coeficientes MFCC

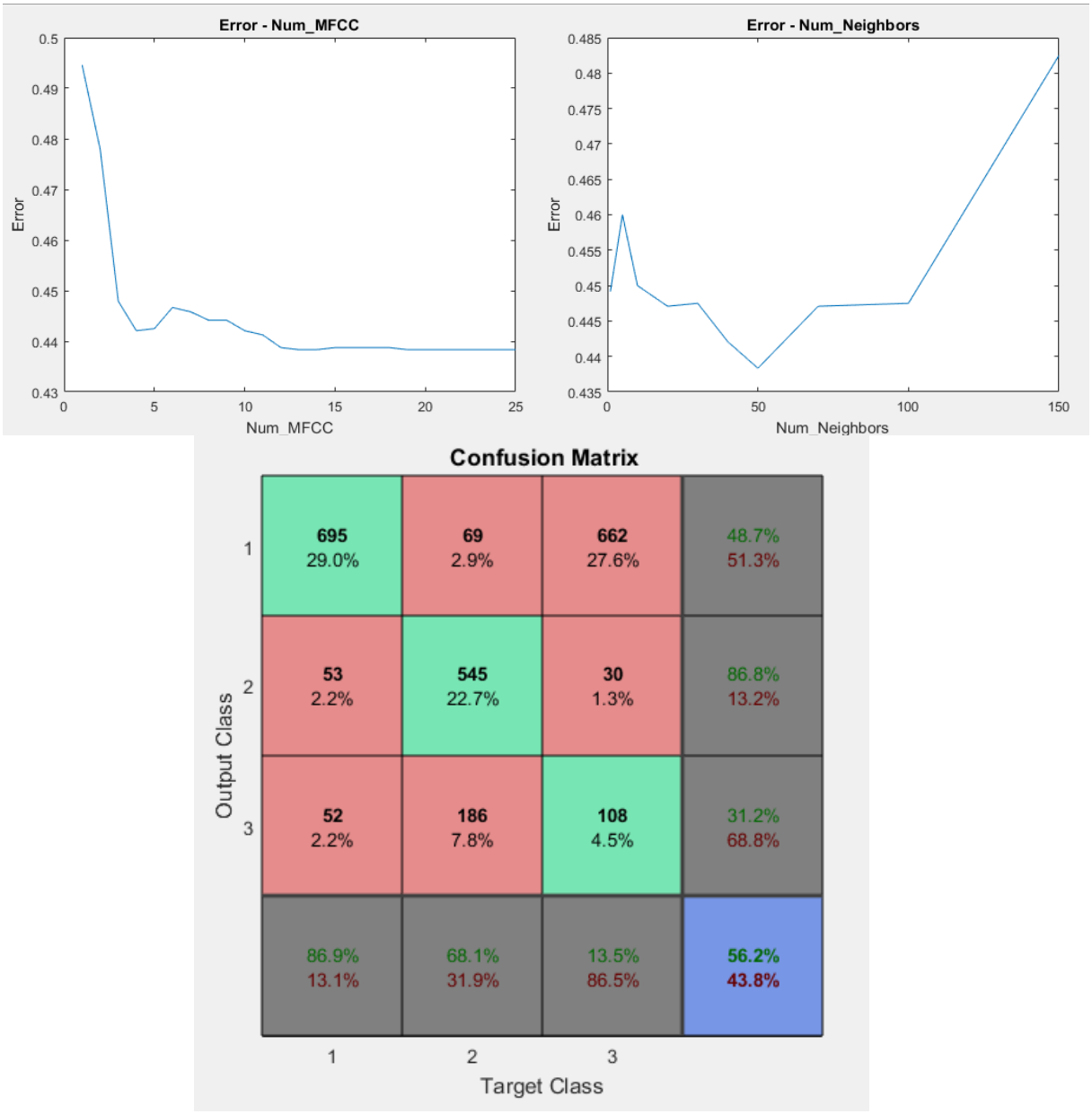


Figura 19. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 56,2 % de aciertos para 50 vecinos y 13 coeficientes MFCC.

Simulación 2.2

Número de especies: 3 (Paloma, Gorrión y Jilguero)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Incluyendo media y desviación estándar del pitch

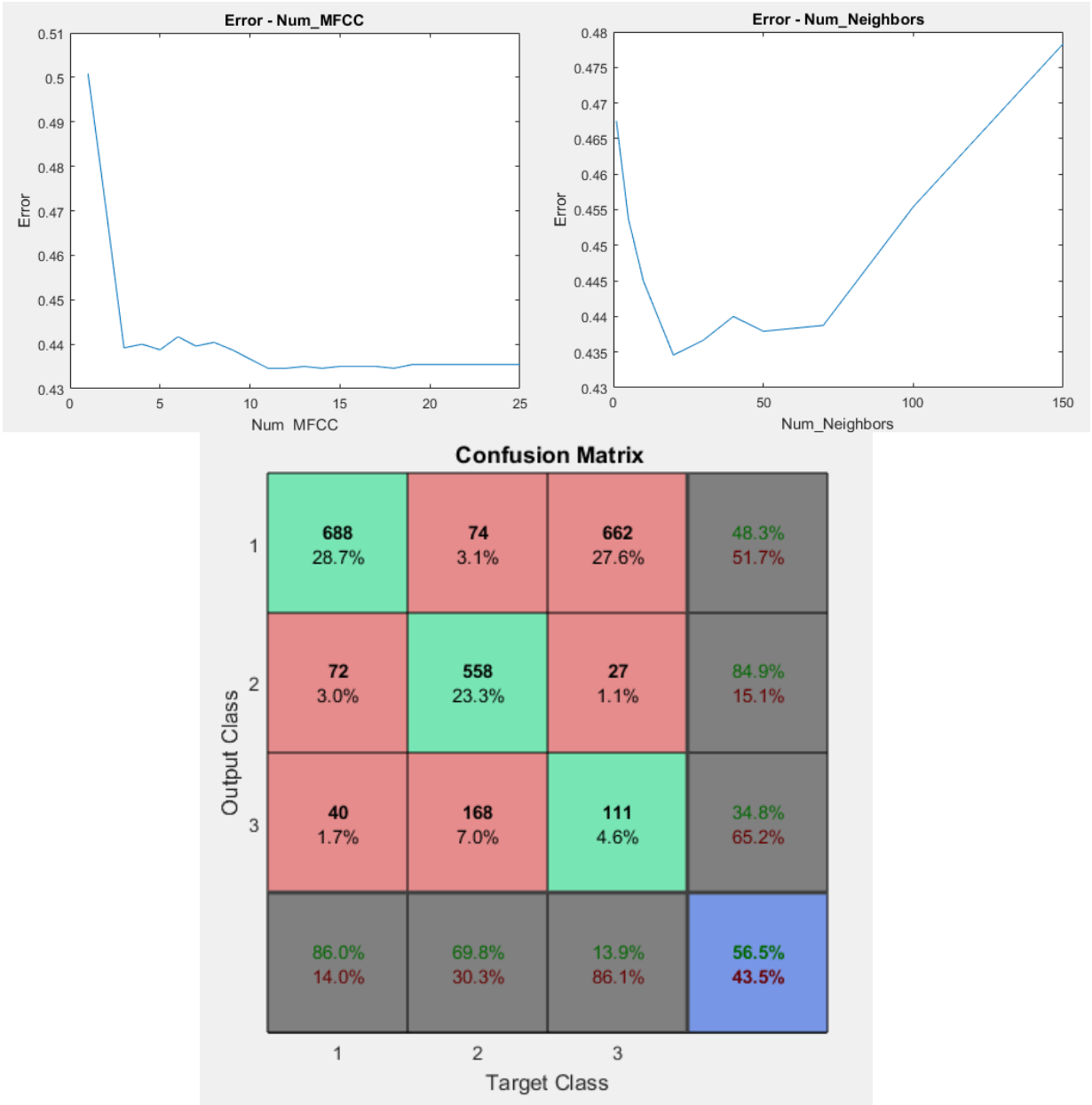


Figura 20. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 56,5 % de aciertos para 20 vecinos y 11 coeficientes MFCC.

Simulación 3.1

Número de especies: 3 (Paloma, Gorrión y Mirlo)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Empleando únicamente los coeficientes MFCC

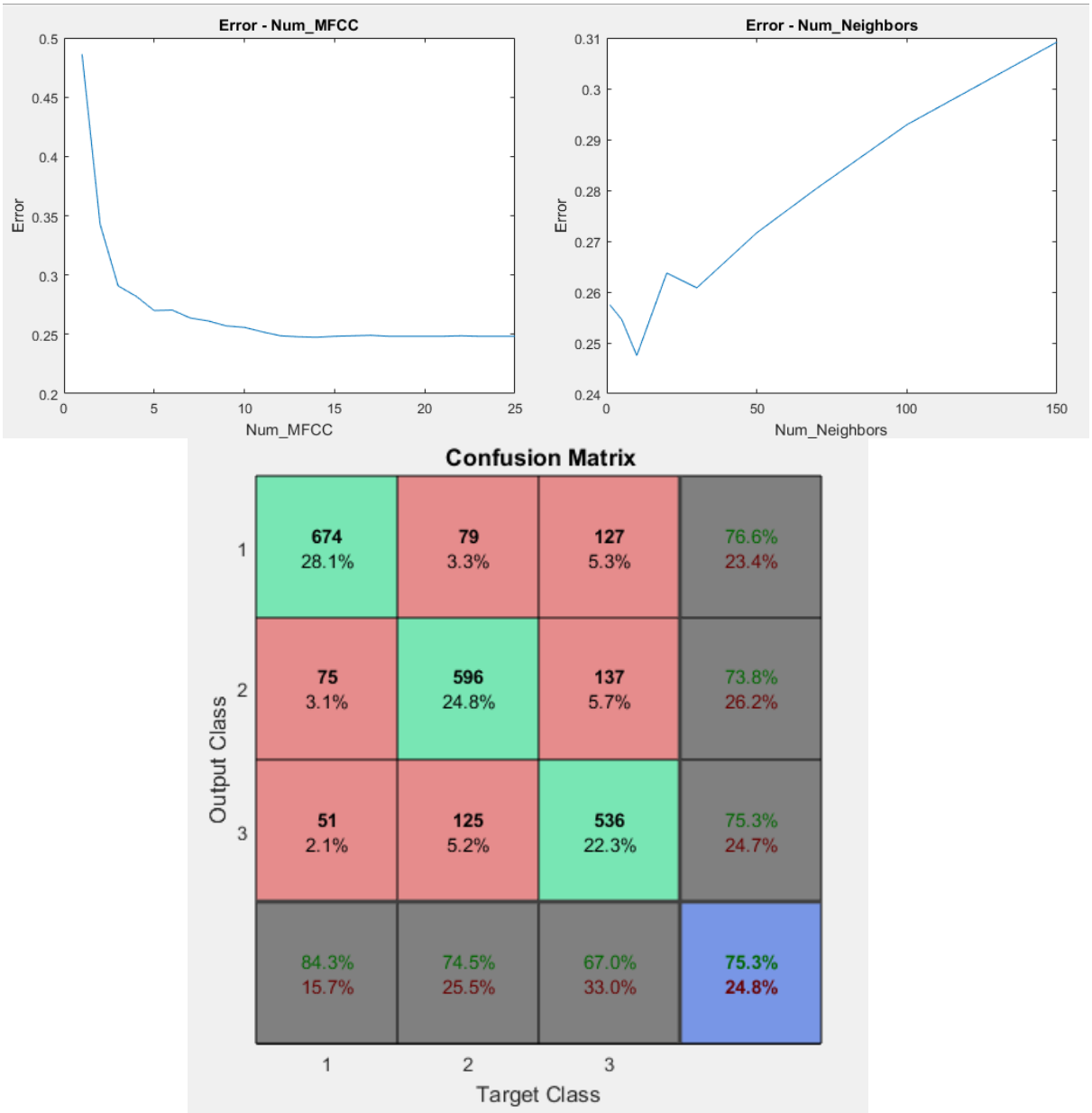


Figura 21. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 75,3 % de aciertos para 10 vecinos y 14 coeficientes MFCC.

Simulación 3.2

Número de especies: 3 (Paloma, Gorrión y Mirlo)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Incluyendo media y desviación estándar del pitch

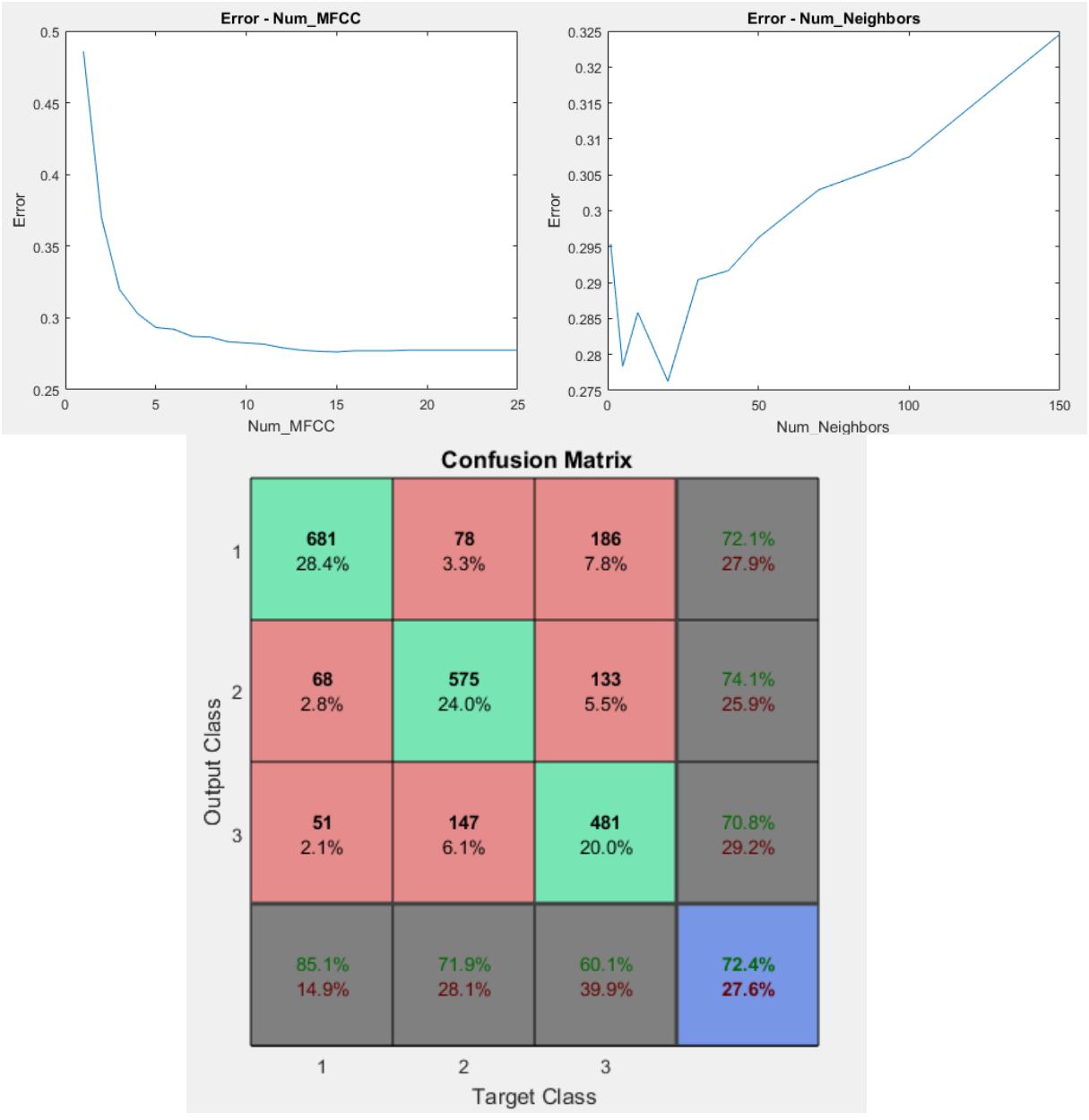


Figura 22. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 72,4 % de aciertos para 20 vecinos y 15 coeficientes MFCC.

Simulación 4.1

Número de especies: 4 (Paloma, Gorrión, Jilguero y Mirlo)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Empleando únicamente los coeficientes MFCC

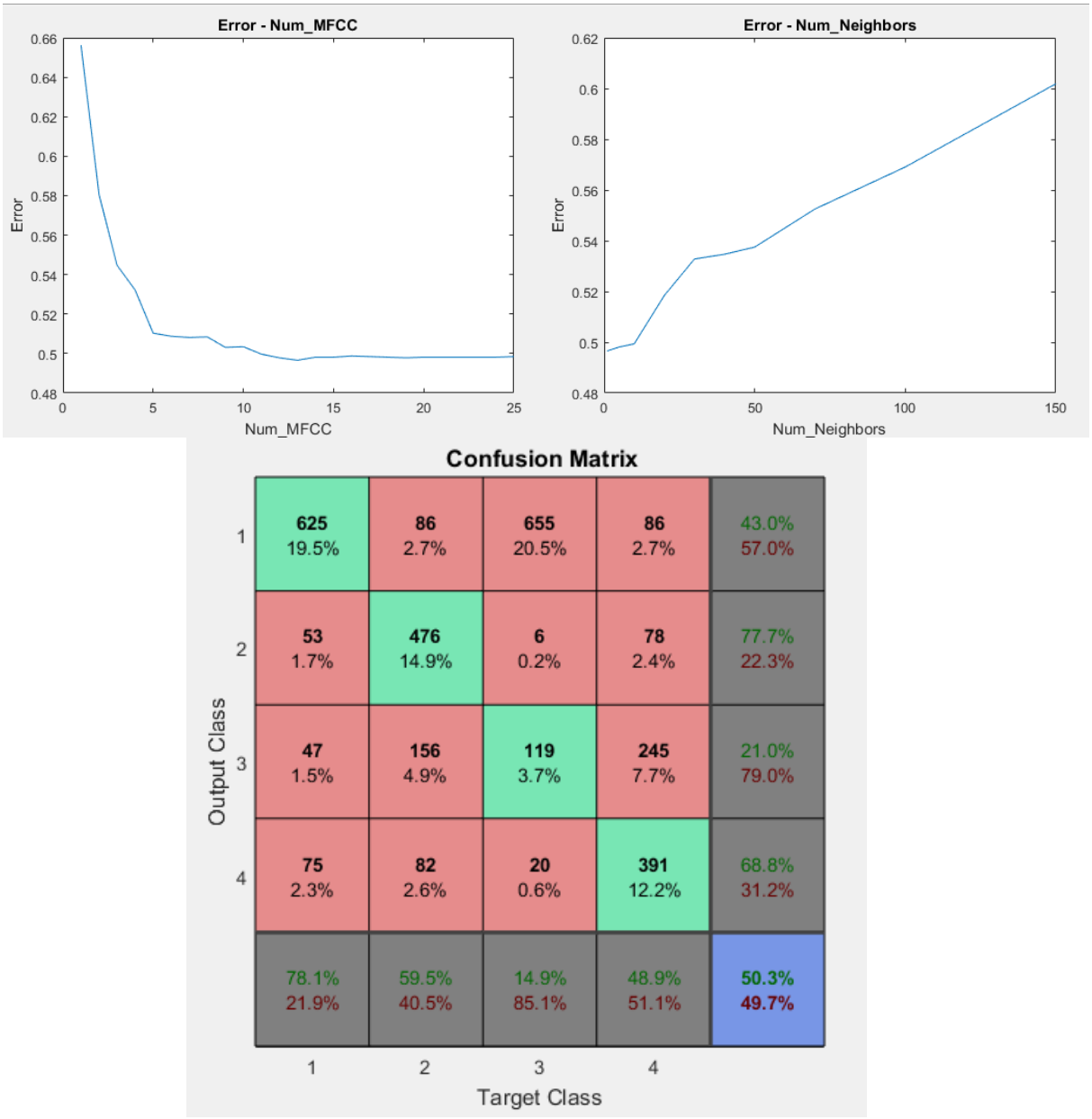


Figura 23. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 50,3 % de aciertos para 1 vecino y 13 coeficientes MFCC.

Simulación 4.2

Número de especies: 4 (Paloma, Gorrión, Jilguero y Mirlo)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Incluyendo media y desviación estándar del pitch

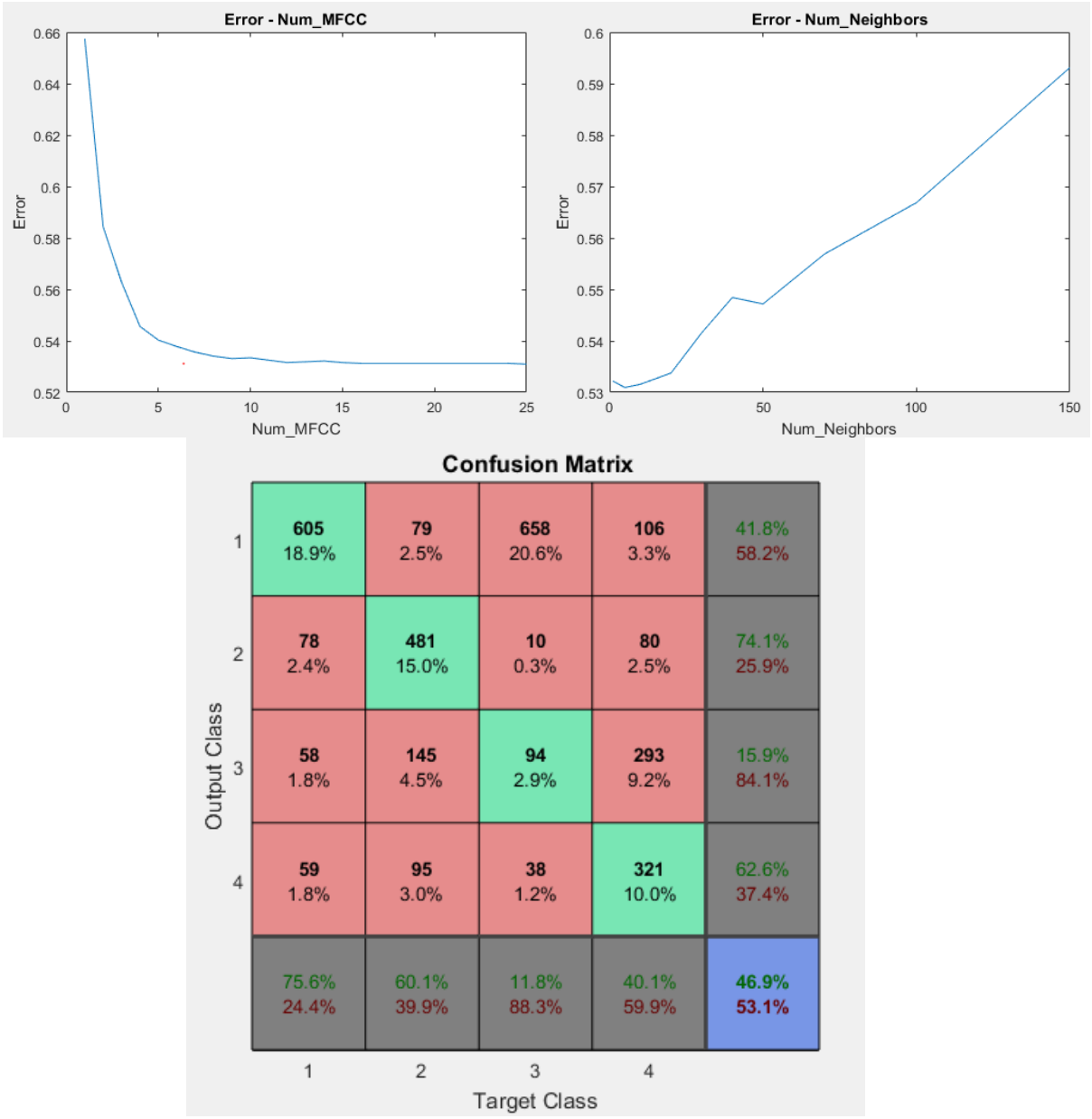


Figura 24. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 46,9 % de aciertos para 5 vecinos y 16 coeficientes MFCC.

Simulación 5.1

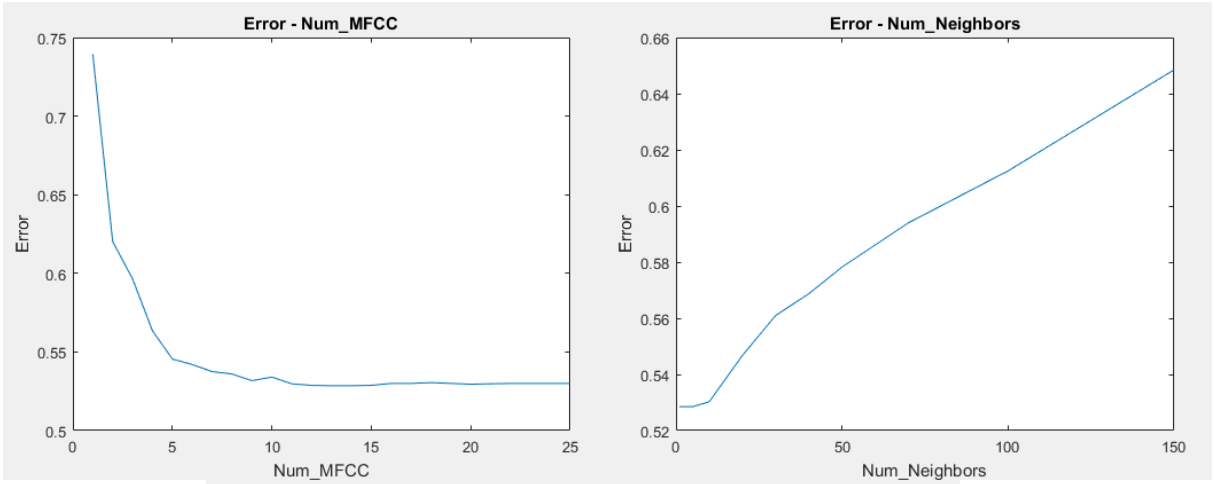
Número de especies: 5 (Paloma, Gorrión, Jilguero, Mirlo y Urraca)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Empleando únicamente los coeficientes MFCC



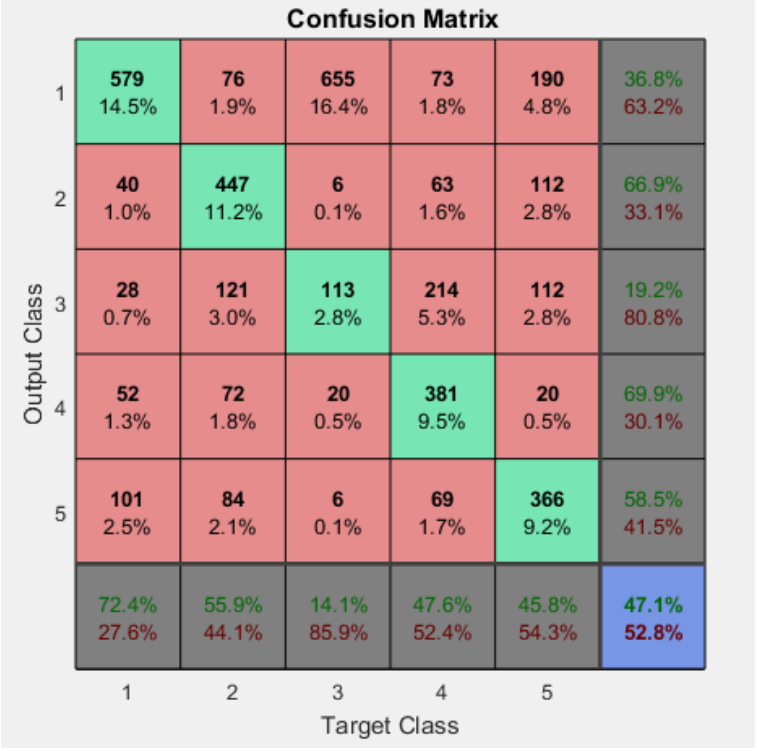


Figura 25. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 47,1 % de aciertos para 1 vecino y 13 coeficientes MFCC.

Simulación 5.2

Número de especies: 5 (Paloma, Gorrión, Jilguero, Mirlo y Urraca)

Frecuencia de muestreo: 44100 Hz

Número de filtros de Mel: 25

Porcentaje de DDBB dedicado a entrenamiento: 50%

Otros: Incluyendo media y desviación estándar del pitch

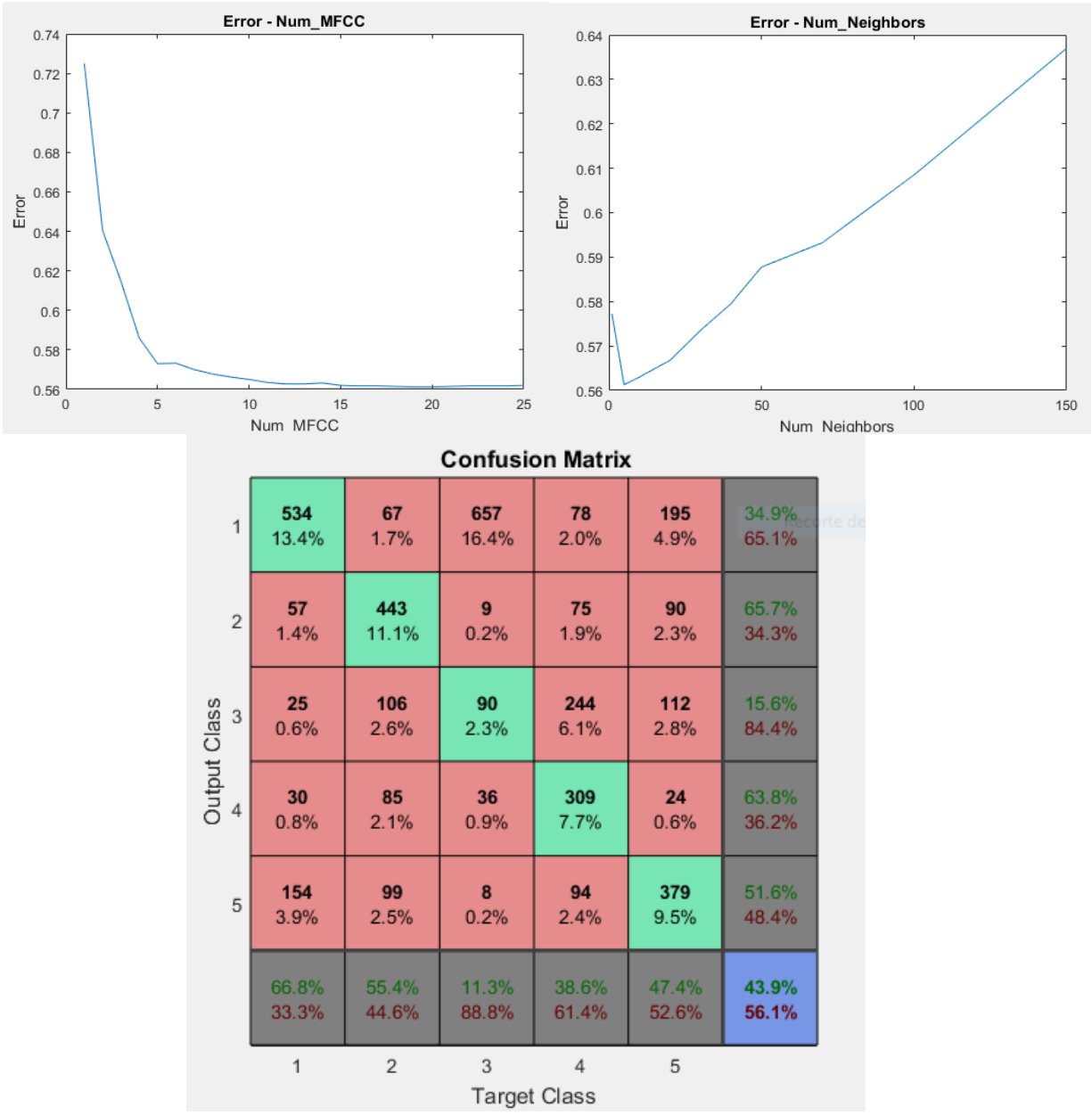


Figura 26. Arriba Izda.) Error – N.º de coeficientes; Arriba Dcha.) Error – N.º de vecinos Abajo) Matriz de confusión.

El mejor resultado obtenido para esta simulación ha sido de un 43,9 % de aciertos para 5 vecinos y 19 coeficientes MFCC.

N.º de Especies	N.º de MFCC	N.º de vecinos	Características adicionales	Aciertos
2	18	20	-	89,1 %
2	16	20	Pitch	88,3 %
3	13	50	-	56,2 %
3	11	20	Pitch	56,5 %
3	14	10	-	75,3 %
3	15	20	Pitch	72,4 %
4	13	1	-	50,4 %
4	16	5	Pitch	46,9 %
5	13	1	-	48,2 %
5	19	5	Pitch	43,9 %
Media	15	15		63 %

Tabla 1. Resultados de las simulaciones.

Analizando las simulaciones que se han llevado a cabo, puede observarse que en la mayoría de los casos los coeficientes MFCC han sido la característica clave a la hora de aportar información útil para la clasificación, mientras que el pitch y los máximos de energía no han aportado suficiente información relevante, lo que no quiere decir que no puedan obtenerse mejores resultados a partir de estos, si no que habría que adaptar estas características al proyecto en cuestión.

Se puede observar que las curvas de probabilidad de error descienden drásticamente de 1 a 5 coeficientes MFCC, lo que demuestra que estos descriptores aportan información útil y altamente decorrelada. Los siguientes coeficientes siguen aportando información relevante, ya que vemos que las curvas de probabilidad de error descienden en torno a un 0,2%, pero como se puede apreciar es un cambio insignificante en comparación con el descenso de alrededor de un 12,3% de 1 a 5 coeficientes. El porcentaje de error se estabiliza en un mínimo alrededor de los 15 coeficientes MFCC. Cabe mencionar que cuantas menos especies distintas participan en la clasificación, menor es la variación de la probabilidad de error ante diferente número de coeficientes, lo que implica que para clasificar un menor número de especies basta con solo unos pocos

coeficientes para obtener altas tasas de acierto, mientras que cuantas más especies participan, más coeficientes son necesarios para unos resultados óptimos.

El número de vecinos con el que comparamos las muestras es un aspecto tan crucial como subjetivo. En el caso de estas simulaciones, los ejemplos de entrenamiento de cada una de las especies están compuesto por 800 muestras de audio y se ha observado que los mínimos de la probabilidad de error se sitúan alrededor de 15 vecinos, lo que supone un 1,9% del número de muestras de la especie en cuestión y un 0,4% del número total de muestras del conjunto de entrenamiento. Para un menor número de especies un porcentaje ligeramente superior de vecinos mejora los resultados obtenidos, mientras que para un número mayor de especies tomar un número mucho menor de vecinos ha demostrado dar mejores resultados. Esto es debido a que cuantas más especies participan en la clasificación más se entremezclan las muestras en el espacio de características y por tanto, al comparar con más vecinos también tomamos erróneamente un mayor número de muestras pertenecientes a otras especies.

Como cabía de esperar, los resultados mejoran cuantas menos especies participan en la clasificación. De las matrices de confusión, podemos observar que es sobre todo la especie 3 (jilguero) la que presenta mayor número de errores, siendo clasificada la mayor parte del tiempo como la especie 1 (paloma), esto se debe a la similitud de sus espectrogramas y un análisis más exhaustivo permitiría adaptar las características extraídas para así mejorar los resultados obtenidos. El resto de las especies cuentan con un número de aciertos mucho mayor.

Todas las simulaciones se han llevado a cabo asignando la mitad de la base de datos al grupo de entrenamiento y la otra mitad al grupo de evaluación. Se ha procedido de esta forma para comprobar el funcionamiento real de la herramienta en un escenario genérico, destinando un porcentaje mayor de la base de datos al conjunto de entrenamiento los resultados mejorarían significativamente.

Manual de usuario

La herramienta desarrollada en este trabajo es intuitiva y de fácil manejo, aun así en este apartado veremos paso a paso el proceso de clasificación de diferentes especies.

El primer paso de este proceso es hacerse con una base de datos de audio extensa de cada una de las especies que queremos clasificar. Deberemos almacenar los audios en formato .mp3 en los directorios correspondientes a cada especie.

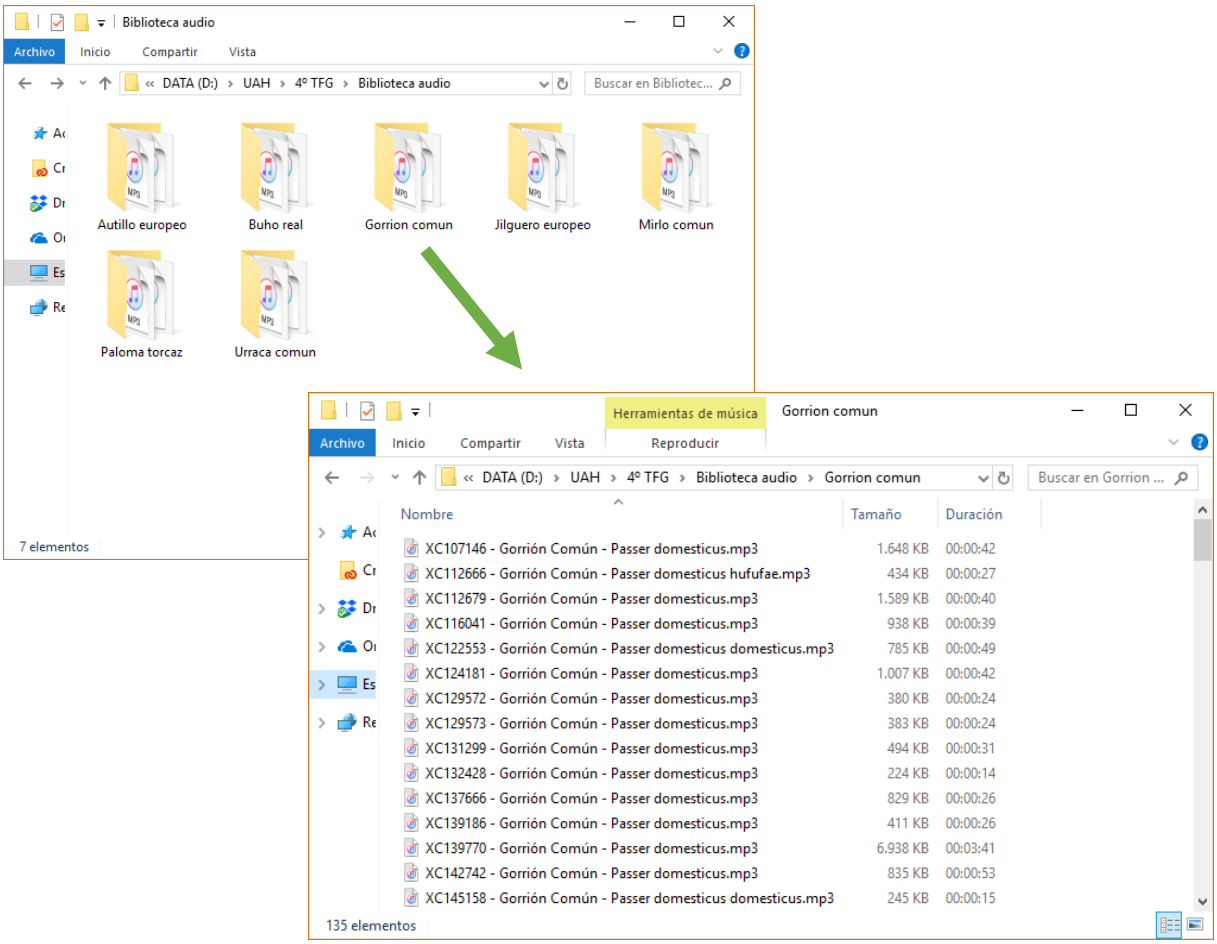


Figura 27. Directorios de las bases de datos.

Una vez copiados los distintos archivos de audio en sus directorios correspondientes se procede a instalar la herramienta. Para la instalación de la herramienta hacer doble-click en el archivo “HerramientaClasificacion.mlappinstall”. El instalador abrirá una ventana de Matlab como se muestra en la siguiente figura.

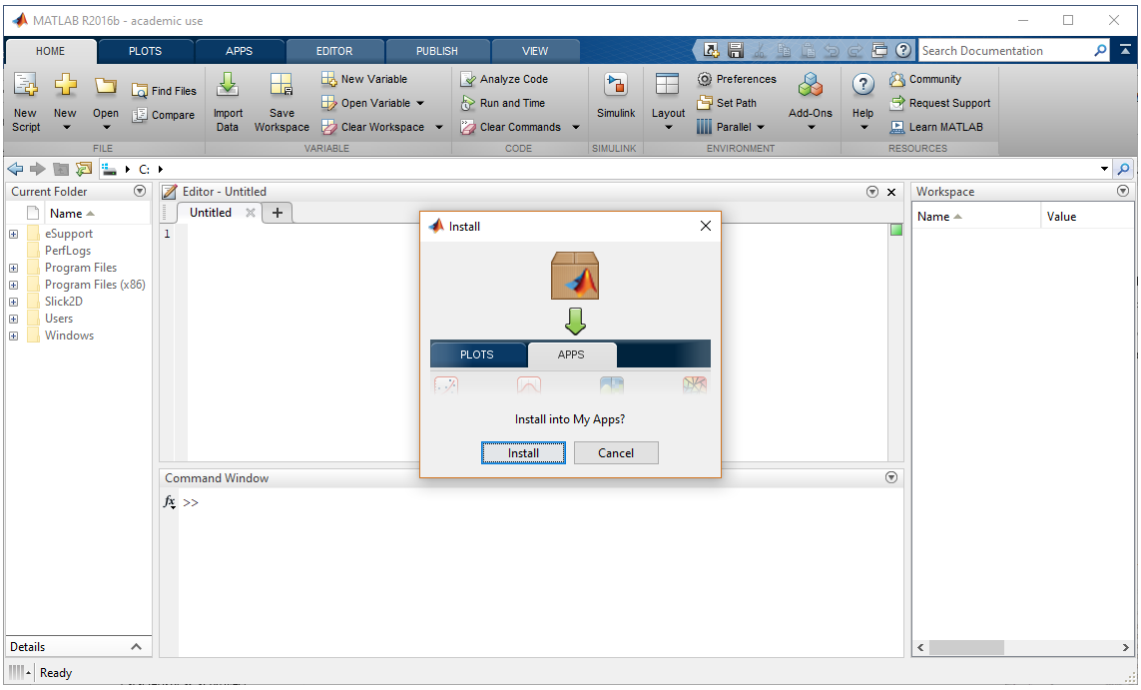


Figura 28. Pantalla de instalación.

Pulsando el botón “Install” la herramienta será instalada como una app más de Matlab y podremos acceder a ella a través de la pestaña “APPS”.

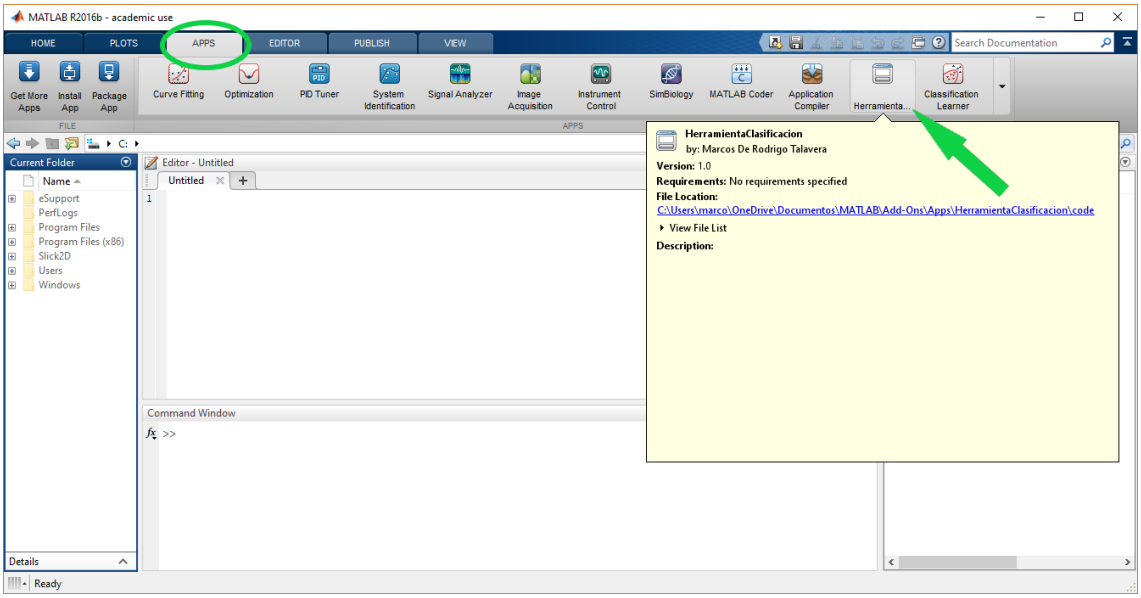


Figura 29. Localización de la aplicación.

Una vez instalada la herramienta procedemos a ejecutarla.

Se nos presentará una interfaz como la que podemos ver en la siguiente figura. Cada una de las pestañas corresponde a una especie diferente y en cada una debemos completar los siguientes campos:

“Audio base directory”: Ruta absoluta del directorio donde se encuentran los archivos .mp3 de la especie en cuestión.

“Feature extraction directory”: Ruta absoluta del directorio donde serán almacenadas las características extraídas de cada audio de la base de datos de la especie.

“Lower frequency limit (Hz)” y “Upper frequency limit (Hz)”: Rango de frecuencias del que queremos extraer características para dicha especie.

“Tag”: Etiqueta con el nombre de la especie.

“Sampling frequency (Hz)”: Será la frecuencia con la que se muestreen los audios de las distintas especies.

“Number of Mel filter banks”: Número de bancos de filtros de Mel, y por ende, número de coeficientes MFCC que obtendremos de las distintas especies.

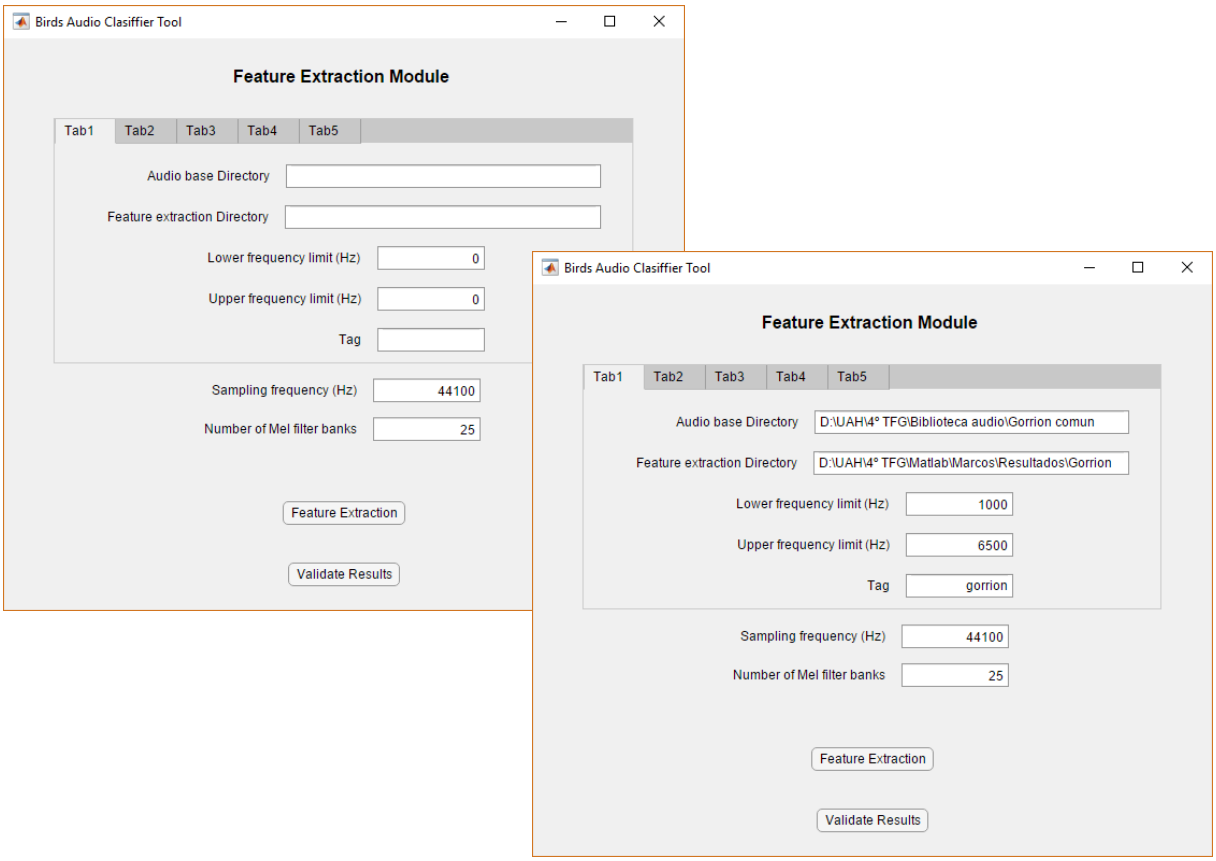


Figura 30. Interfaz del módulo de extracción de características de la herramienta.

Una vez completados los campos de cada una de las especies se pulsa el botón “Feature Extraction” y la herramienta procederá a la extracción de las diferentes características, almacenando los resultados en un archivo .mat por cada archivo .mp3 de las bases de datos, con el mismo nombre de fichero.

Cuando la herramienta ha terminado de extraer las características de las diferentes especies, se pulsa el botón “Validate Results”, que abrirá una nueva ventana con el módulo de clasificación.

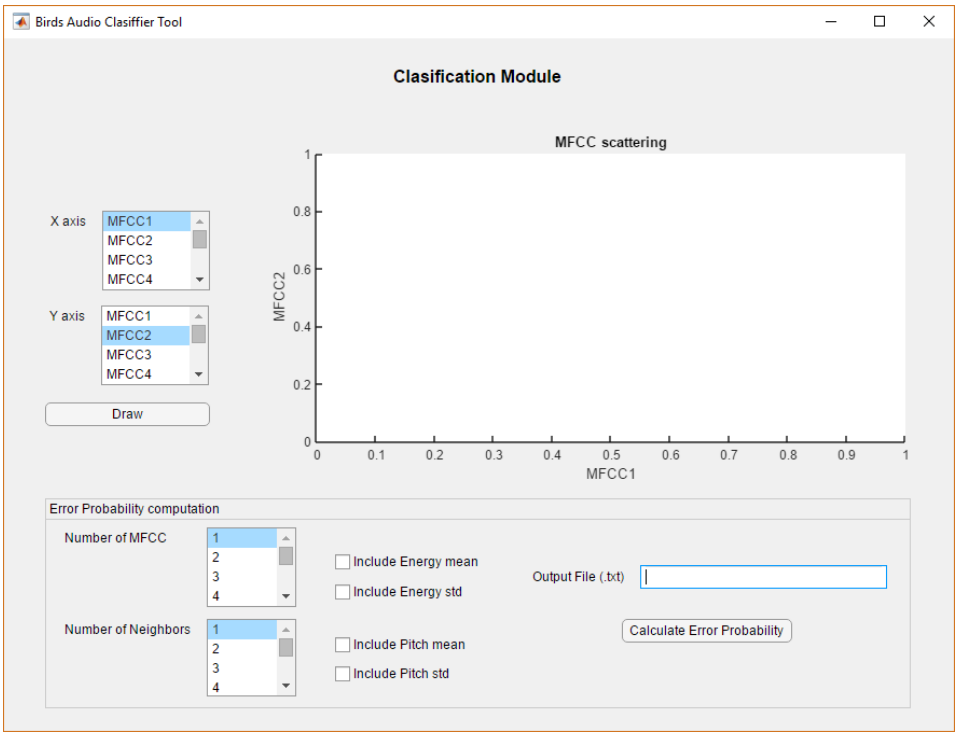


Figura 31. Interfaz del módulo de clasificación de la herramienta.

Esta interfaz cuenta con dos bloques. El primero de ellos dedicado a la representación bidimensional de las características de todas las especies involucradas, donde podremos seleccionar que coeficientes queremos representar en cada eje y el botón “Draw” dibujará estas características en el plano, utilizando los tags que rellenamos previamente para la leyenda.

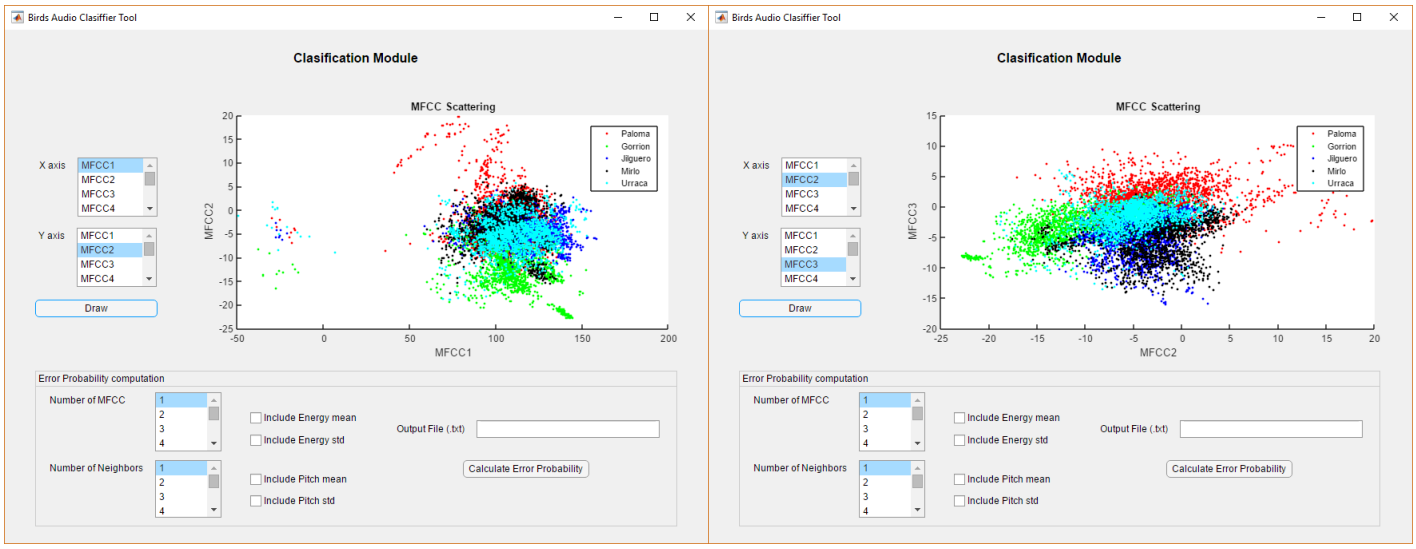


Figura 32. Ejemplo de representación de características.

El segundo realiza una serie de simulaciones con los parámetros que le indiquemos y almacena los resultados de dichas simulaciones en el fichero .txt que especifiquemos al pulsar el botón “Calculate Error Probability”, estos resultados serán la probabilidad de error del sistema en cada una de las simulaciones, acompañadas de los parámetros que se emplearon en su obtención. Entre los parámetros que podemos seleccionar para las simulaciones se encuentran:

“Number of MFCC”: Número de coeficientes que queremos tomar en cuenta para la clasificación. Si indicamos por ejemplo, el 10, se realizarán las simulaciones comparando nada más los 10 primeros coeficientes. Se permite seleccionar varios valores y la herramienta realizará simulaciones con todos ellos.

“Number of neighbors”: Número de vecinos del espacio de características con los que se compararán las muestras a clasificar. Se permite seleccionar varios valores y la herramienta realizará simulaciones con todos ellos.

“Include Energy mean” y “Include Energy std”: Marcar si queremos que las simulaciones se realicen incluyendo la media y/o la desviación estándar de los máximos de energía de la señal.

“Include Pitch mean” y “Include Pitch std” : Marcar si queremos que las simulaciones se realicen incluyendo la media y/o la desviación estándar del pitch de la señal.

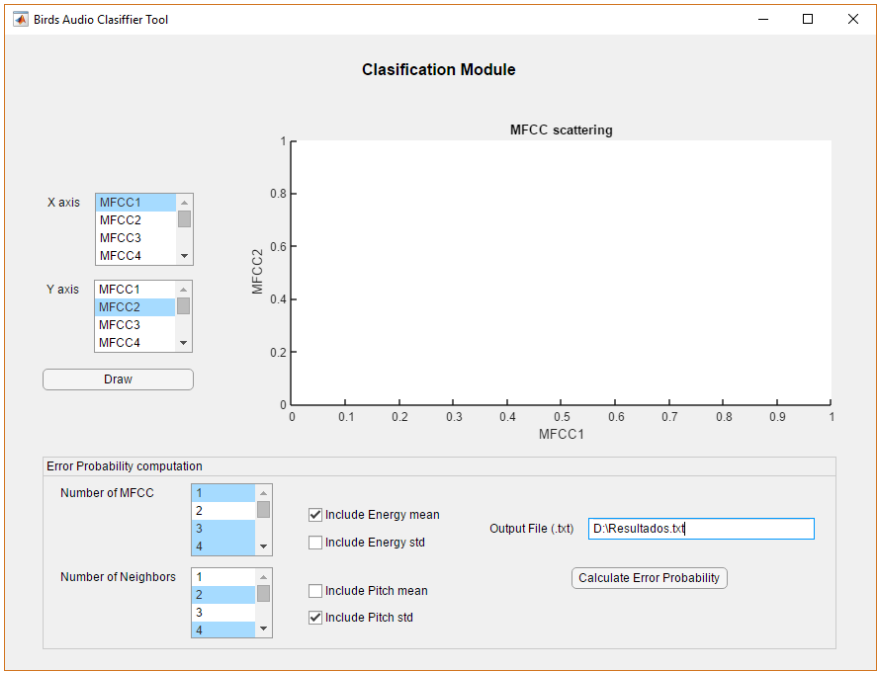
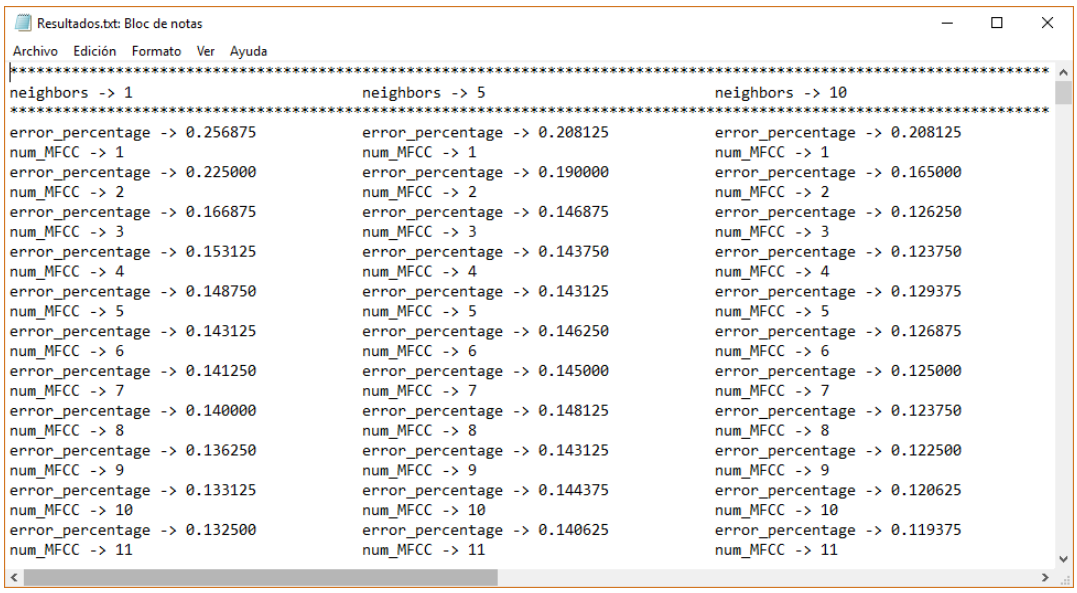


Figura 33. Ejemplo de parámetros para la realización de simulaciones.

Cuando la herramienta termina la realización de simulaciones, podemos observar los resultados obtenidos en cada una de estas en el fichero indicado.



Resultados.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
**********************************************************************************************************************
neighbors -> 1                          neighbors -> 5                          neighbors -> 10
**********************************************************************************************************************
error_percentage -> 0.256875            error_percentage -> 0.208125            error_percentage -> 0.208125
num_MFCC -> 1                           num_MFCC -> 1                           num_MFCC -> 1
error_percentage -> 0.225000            error_percentage -> 0.190000            error_percentage -> 0.165000
num_MFCC -> 2                           num_MFCC -> 2                           num_MFCC -> 2
error_percentage -> 0.166875            error_percentage -> 0.146875            error_percentage -> 0.126250
num_MFCC -> 3                           num_MFCC -> 3                           num_MFCC -> 3
error_percentage -> 0.153125            error_percentage -> 0.143750            error_percentage -> 0.123750
num_MFCC -> 4                           num_MFCC -> 4                           num_MFCC -> 4
error_percentage -> 0.148750            error_percentage -> 0.143125            error_percentage -> 0.129375
num_MFCC -> 5                           num_MFCC -> 5                           num_MFCC -> 5
error_percentage -> 0.143125            error_percentage -> 0.146250            error_percentage -> 0.126875
num_MFCC -> 6                           num_MFCC -> 6                           num_MFCC -> 6
error_percentage -> 0.141250            error_percentage -> 0.145000            error_percentage -> 0.125000
num_MFCC -> 7                           num_MFCC -> 7                           num_MFCC -> 7
error_percentage -> 0.140000            error_percentage -> 0.148125            error_percentage -> 0.123750
num_MFCC -> 8                           num_MFCC -> 8                           num_MFCC -> 8
error_percentage -> 0.136250            error_percentage -> 0.143125            error_percentage -> 0.122500
num_MFCC -> 9                           num_MFCC -> 9                           num_MFCC -> 9
error_percentage -> 0.133125            error_percentage -> 0.144375            error_percentage -> 0.120625
num_MFCC -> 10                          num_MFCC -> 10                          num_MFCC -> 10
error_percentage -> 0.132500            error_percentage -> 0.140625            error_percentage -> 0.119375
num_MFCC -> 11                          num_MFCC -> 11                          num_MFCC -> 11
```

Figura 34. Ejemplo de resultados de las simulaciones.

Conclusiones

En el presente trabajo se ha observado el funcionamiento de la herramienta para la clasificación automática de distintas especies de aves, desde la extracción de características hasta los algoritmos de clasificación de aprendizaje automático supervisado.

Las bases de datos juegan un papel crucial dentro de la clasificación, ya que son estas las que poseen los ejemplos con los que el sistema debe aprender, por lo que una buena base de datos es de vital importancia para llevar a cabo un buen entrenamiento del sistema. Otro aspecto importante referente a las bases de datos, como ya se ha comentado en apartados anteriores, es la determinación de ejemplos destinados a entrenamiento o a evaluación, ya que una evaluación que contenga datos de entrenamiento, es decir, que los datos de un mismo audio se hayan repartido entre ambos grupos, dará lugar a unos falsos resultados favorables.

Se ha podido observar en las simulaciones realizadas la importancia de una buena selección de características para la obtención de unos resultados óptimos. Los coeficientes MFCC han demostrado su gran capacidad de aportar información útil, siendo estos las características principales de todas las simulaciones realizadas. Por otro lado, características como el pitch y los máximos de energía de la señal han demostrado no ser de gran utilidad para la clasificación de aves, esto no quiere decir que no sea posible obtener mejores resultados a partir de ellas, si no que habría que adaptarlas a este proyecto en concreto para que aportasen información útil y relevante.

En cuanto al algoritmo de clasificación empleado, se ha observado que las probabilidades de error disminuyen y se estabilizan cuantos más coeficientes MFCC se toman en cuenta para la clasificación, con unos pocos coeficientes (alrededor de 5) las curvas de probabilidad de error presentan mínimos locales y cuantos más coeficientes se toman para la clasificación, estas curvas de error se estabilizan alrededor de dicho mínimo. En cuanto al número de vecinos con los que la herramienta compara los ejemplos de evaluación se ha observado que un menor número de vecinos da lugar a unos resultados más óptimos, especialmente cuantas más especies distintas se pretenda clasificar. Esto podría variar si se aumentase el número de muestras de cada especie, pero se recomienda no tomar un número de vecinos elevado ya que podría ocurrir que las características de

una muestra se encontrasen en los límites de la nube de puntos que representan las características de una especie en concreto, y al comparar con un número elevado de vecinos es probable que se tomasen más muestras de las especies colindantes que de la propia.

Por último, destacar que los objetivos del proyecto se han alcanzado con éxito, realizando una herramienta de clasificación de fácil manejo y siendo las probabilidades de error obtenidas en las distintas simulaciones de la herramienta relativamente bajas.

Futuras líneas de investigación

La clasificación automática es un campo muy útil y con un sinfín de aplicaciones. A lo largo de este proyecto y especialmente en el apartado de simulaciones, ha quedado patente lo complejo que puede llegar a ser este proceso y como pueden variar los resultados obtenidos en función de que parámetros se empleen.

Uno de los factores más importantes a la hora de realizar una correcta clasificación es el uso de determinadas características. Por ello, una de las líneas futuras más importantes para la obtención de mejores resultados será la incorporación de otras características, como el centroide espectral o la envergadura espectral, entre otros.

En el campo del aprendizaje automático existen una gran variedad de algoritmos que permiten realizar clasificaciones. Uno de los más utilizados actualmente para sistemas de audio son las redes neuronales, que consisten en un algoritmo que imita de alguna manera las redes neuronales del cerebro humano, por lo que, si una persona es capaz de discernir entre dos especies de ave, cabe pensar que una herramienta dotada de esta capacidad también pueda. En este trabajo se ha empleado únicamente el algoritmo de clasificación “K-Nearest-Neighbors” y como futura línea de investigación se plantea la incorporación de otros algoritmos de clasificación como las redes neuronales o los arboles de decisión, de esta forma la herramienta podrá realizar simulaciones con los distintos algoritmos para determinar cual de ellos obtiene mejores resultados.

Como hemos podido comprobar a la hora de extraer características de los audios que forman la base de datos, muchos de estos contienen sonidos interferentes provenientes de ruido ambiental o incluso de otras especies de ave. Con objeto de mejorar los resultados obtenidos por la herramienta de clasificación, una de las líneas futuras de investigación será el pre-procesado del audio, se podría incorporar un algoritmo de separación de fuentes para suprimir aquellos ruidos que afecten negativamente al funcionamiento del sistema.

Por último, sería interesante añadir una nueva funcionalidad a la herramienta que permitiese no solo clasificar los audios de la base de datos por especie, si no que además indicase el significado de los sonidos producidos por dichas especies. La mayoría de las especies emiten sonidos diferentes para comunicar un número limitado de mensajes como, por ejemplo, para indicar su necesidad de apareamiento, o para expresar alarma

ante un depredador cercano. Clasificar estos mensajes a partir de las características del sonido producido por una especie en concreto no resultaría costoso, ya que se podría emplear la misma herramienta de clasificación que se ha desarrollado en este proyecto aplicada en orden jerárquico, de forma que primero determinase la especie en cuestión y posteriormente procediese a identificar el mensaje. La posibilidad de conocer el significado de estos mensajes a través de la herramienta la convertiría en un traductor, ideal para aplicaciones de tiempo real.

Bibliografía

MFCC. Wikipedia, La enciclopedia libre. Fecha de consulta: enero 16, 2018 de https://es.wikipedia.org/wiki/MFCC.

James Lyons. (2009-2012). Mel Frequency Cepstral Coefficient (MFCC): Practical cryptography. http://www.practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/.

Abdel-rahman Mohamed. (2014). Deep Neural Network acoustic models for ASR (tesis doctoral). University of Toronto. Toronto.

Leif E. Peterson. (2009). K-nearest neighbor: scholarpedia.org. http://scholarpedia.org/article/K-nearest_neighbor.

Kilian Q.Weinberger, John Blitzer y Lawrence K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. University of Pennsylvania.

Andric Pasillas. (2017). Machine Learning: blog.adext.com. https://blog.adext.com/es/machine-learning-guia-completa.

Kamil Wojcicki. (2011). HTK MFCC Matlab: Mathworks. https://es.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab?focused=5199994&tab=function.

Martínez, G. y Aguilar, G. (2013). Reconocimiento de voz basado en MFCC, SBC y espectrogramas. Ingenius. N.º 10.

Álvarez, Lorena. Clasificación automática de señales sonoras aplicada a la mejora de la inteligibilidad de la voz en audífonos digitales (trabajo de fin de grado). Universidad de Alcalá de Henares.

López García, María. (2014). Análisis de estructura temporal de datos musicales para clasificación (trabajo de fin de grado). Universidad Carlos III de Madrid.

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITECNICA
SUPERIOR

Universidad
de Alcalá