uc3m | Universidad Carlos III de Madrid

Grado en Ingeniería de Sistemas Audiovisuales
2018-2019

Trabajo Fin de Grado

“Detección del trastorno específico del lenguaje en niños mediante el análisis acústico de sus voces”

Alberto Cuevas González

Tutor/es

Ascensión Gallardo Antolín

01/07/2019, Universidad Carlos III, campus de Leganés.

AGRADECIMIENTOS

Me gustaría agradecer a las personas que me han apoyado o ayudado en alguna manera en la realización de este trabajo.

En primer lugar, a Ascen, mi tutora, por mostrarme su confianza en que podía acabar el TFG dentro de plazo y por toda la información, consejos y ayuda que me ha proporcionado para poder completar el trabajo. También por la proposición de este trabajo, que dentro de que es un trabajo el cual requiere esfuerzo para la correcta realización del mismo, es un trabajo bonito de realizar.

También agradecer a mi familia y amigos por darme los ánimos suficientes para poder concluir el trabajo.

En definitiva, a todos vosotros, gracias.

RESUMEN

El síndrome específico del lenguaje, también conocido por sus siglas en inglés SLI (*Specific Language Impairment*), es un síndrome que se estima que afecta en torno al 7 u 8 por ciento de la población total de niños en el mundo. Este síndrome, se caracteriza por que aquel niño que lo sufre tiene dificultades en el aprendizaje del lenguaje sin tener ninguna otra deficiencia que pueda desembocar en problemas en el habla o lingüísticos.

La problemática que existe en la actualidad para diagnosticar el SLI es que no se basa en medidas objetivas, sino que se diagnostica subjetivamente por parte de pediatras y pedagogos expertos en el tema.

El objetivo de este trabajo es que se pueda crear un sistema basado en el aprendizaje máquina que sea capaz de determinar con la mayor probabilidad de acierto posible la existencia o no del SLI en niños mediante el análisis de sus voces.

Este sistema se ha desarrollado a partir de una base de datos con audios de niños con y sin el síndrome específico del lenguaje. El sistema consta básicamente de dos etapas: extracción de características acústicas y clasificador.

En la primera etapa, se extraen un conjunto de parámetros acústicos que representan las características más relevantes de la voz de cada niño. En concreto, se han utilizado los parámetros mel-cepstrales (*Mel Frequency Cepstrum Coefficients*, MFCC) y se han probado varias variantes, como la inclusión de la log-energía y de los parámetros delta-MFCC, los cuales son las derivadas de los parámetros MFCC y modelan su evolución temporal.

La segunda etapa consiste en un clasificador binario basado en máquinas de vectores soporte (*Support Vector Machine*, SVM) con diferentes funciones Kernel.

En cuanto a la parte experimental, se han realizado varios conjuntos de pruebas en distintas condiciones: dependencia e independencia de locutor, y audios limpios y contaminados con ruido. Para cuantificar el funcionamiento del sistema, se han utilizado las medidas de precisión, *recall* y *F-score*.

El sistema ha obtenido altas prestaciones con habla limpia, tanto para el caso dependiente como independiente de locutor. Con respecto al habla ruidosa, como era de esperar, se observa una degradación del funcionamiento del sistema a bajas relaciones señal a ruido (*Signal-to-Noise Ratio*, SNR), especialmente para el caso independiente de locutor. No obstante, para SNRs medias y altas, se obtiene un *F-score* superior a 0.9 para el caso independiente de locutor y con la utilización de los parámetros MFCC y sus derivadas y el Kernel gaussiano.

ABSTRACT

Specific Language Impairment (SLI) is a syndrome that is estimated to affect about 7 to 8 percent of the world's total child population. This syndrome is characterized by the fact that a child who suffers from it has difficulties in learning language without having any other impairment that could lead to problems in speech or language.

The current problem in diagnosing SLI is that it is not based on objective measures, but is diagnosed subjectively by paediatricians and pedagogues who are experts in the subject.

The aim of this work is to create develop a system based on machine learning techniques that is capable of determining with the greatest probability of success the existence or not of SLI in children through the analysis of their voices.

This system has been developed from a database with audios of children with and without the specific language syndrome. It basically consists of two stages: extraction of acoustic characteristics and classifier.

In the first stage, a set of acoustic parameters that represent the most relevant characteristics of each child's voice are extracted. Specifically, Mel Frequency Cepstrum Coefficients (MFCC) have been used and several variants have been tested, such as the inclusion of log-energy and delta-MFCC parameters, which are the derivatives of MFCCs and model their temporal evolution.

The second stage consists of a binary classifier based on Support Vector Machines (SVM) with different Kernel functions.

As for the experimental part, several sets of tests have been carried out under different conditions: dependence and independence of the speaker, and clean and noise-contaminated audios. In order to quantify the performance of the system, precision, recall and F-score measurements have been used.

The system has obtained high performance with clean speech, both for the dependent and independent speaker cases. With respect to noisy speech, as was to be expected, a degradation of the functioning of the system at low signal-to-noise ratio (SNR) is observed, especially for the independent speaker case. However, for medium and high SNRs, an F-score higher than 0.9 is obtained for the independent speaker case and with the use of the MFCC parameters and their derivatives and the Gaussian kernel.

ÍNDICE GENERAL

1. Introducción 11
1.1. Planteamiento del problema 11
1.2. Estructura del documento 11
1.3. Marco regulador 12
1.4. Entorno Socioeconómico 12
2. Estado del arte 13
3. Diseño de la Solución 15
3.1. Introducción 15
3.2. Base de datos 16
3.2.1. Base de datos original 16
3.2.2. Adecuación y división de la base de datos en subgrupos. 20
3.2.3. Justificación de la elección de la base de datos y su distribución. 22
3.3. Extracción de parámetros. 23
3.3.1. Justificación de la elección de los parámetros MFCC. 25
3.4. Sistema de clasificación 25
3.4.1. Grupos de train y modelo de entrenamiento 25
3.4.2. Justificación de la elección de los grupos train y de los modelos de entrenamiento 26
3.4.3. Grupos de test 27
3.4.4. Justificación de la elección de los grupos de test 30
4. Implementación 32
4.1. Implementación del manejo de la base de datos. 32
4.2. Implementación de la extracción de los parámetros MFCC. 32
4.3. Implementación del entrenamiento de los distintos modelos. 34
4.4. Implementación de la realización de la clasificación. 35
4.5. Implementación de la obtención y visualización de los resultados. 35
5. Pruebas y resultados 39
5.1. Protocolo experimental 39
5.2. Resultados 41
5.2.1. Comentario de resultados 50
6. Líneas futuras. 52
7. Conclusiones 53
8. Presupuesto 54
9. Anexos 55
10. Referencias 57
Apéndice: English summary 58

ÍNDICE DE FIGURAS

- **Figura 1**: Diseño de la solución
- **Figura 2**: Distribución de audios en base de datos original
- **Figura 3**: Extracción de los MFCC
- **Figura 4**: Tabla Excel
- **Figura 5**: Tabla Excel (2)
- **Figura 6**: Resultados Excel
- **Figura 7**: Gráfico pruebas dependientes del locutor con modelo polinómico grado 3
- **Figura 8**: Gráfico pruebas independientes del locutor con modelo gaussiano grado 3
- **Figura 9**: lectura.m
- **Figura 10**: escritura_mfcc.m
- **Figura 11**: matriz.m

ÍNDICE DE TABLAS

- **Tabla 1**: Contenido por locutor de la base de datos.
- **Tabla 2**: Identificadores vocales.
- **Tabla 3**: Identificadores consonantes.
- **Tabla 4**: Identificadores monosílabo.
- **Tabla 5**: Identificadores bisílabos.
- **Tabla 6**: Identificadores trisílabos.
- **Tabla 7**: Identificadores palabras de 4 sílabas.
- **Tabla 8**: Identificadores palabras de 5 sílabas.
- **Tabla 9**: Contenido por locutor de la base de datos.
- **Tabla 10**: Pruebas
- **Tabla 11**: Resultados dependientes del locutor y función gaussiana
- **Tabla 12**: Resultados dependientes del locutor y función polinómica de grado 2
- **Tabla 13**: Resultados dependientes del locutor y función polinómica de grado 3 (MFCC)
- **Tabla 14**: Resultados dependientes del locutor y función polinómica de grado 3 (DMFCC)
- **Tabla 15**: Resultados dependientes del locutor y función lineal
- **Tabla 16**: Resultados independientes del locutor y función gaussiana (MFCC)
- **Tabla 17**: Resultados independientes del locutor y función lineal
- **Tabla 18**: Resultados independientes del locutor y función gaussiana (DMFCC)
- **Tabla 19**: Resultados independientes del locutor y función polinómica grado 2
- **Tabla 20**: Resultados independientes del locutor y función polinómica grado 3

ÍNDICE DE ECUACIONES

- **Ecuación 1**: Preénfasis.
- **Ecuación 2**: Transformada discreta de Fourier.
- **Ecuación 3**: Coeficientes MFCC
- **Ecuación 4**: Precisión
- **Ecuación 5**: Recall
- **Ecuación 6**: F-score

1. Introducción

En este apartado se realizará el planteamiento del problema y se detallarán aspectos necesarios en un trabajo de fin de grado como son el marco regulador del mismo y cómo puede afectar al entorno socioeconómico en el que se encuentra.

1.1. Planteamiento del problema

El presente documento, con el trabajo que detrás ha llevado, tiene como objetivo facilitar los procesos de detección del trastorno del síndrome específico del lenguaje (*Specific Language Impairment*, SLI) en niños mediante la creación de un sistema de detección de dicho síndrome a través del análisis de audios de voz de los posibles pacientes.

La posibilidad de que a través de varios audios grabados simplemente con un móvil por los padres a sus hijos y que estos audios sean enviados a un sistema remoto en el que esté ubicado el sistema de detección, supondría un ahorro de tiempo para médicos y logopedas que estarían más liberados para otras labores u otros pacientes. Para lograr este propósito, el sistema desarrollado debe tener una alta probabilidad de acierto.

En cualquier caso, es de resaltar que el objetivo de este tipo de sistemas no es sustituir al personal médico y otros especialistas, si no desarrollar una herramienta de ayuda al diagnóstico para este tipo de trastornos del lenguaje.

Con todo esto, el planteamiento del problema está enfocado a desarrollar una herramienta que pueda ayudar en el diagnóstico clínico del síndrome específico del lenguaje en niños mediante el análisis de sus voces.

1.2. Estructura del documento

La estructura de la memoria será la siguiente:

Hay una primera parte de introducción en la que se planteará el problema y se presentarán los objetivos del trabajo y también se detallarán aspectos importantes y necesarios como son el marco regulador, que aclara las normativas aplicables a la memoria, y el entorno socioeconómico que tiene como objetivo situar el trabajo en el contexto de actualidad.

A continuación, en el capítulo 2, **Estado del arte,** se hablará sobre hasta qué punto han llegado o cómo se han enfocado otras investigaciones relacionadas con el SLI.

Una vez llegados aquí, se empezará con la parte gruesa de la memoria y que trata sobre el diseño de la solución en el apartado 3, **Diseño de la Solución**, sobre cómo se ha implementado este diseño en el capítulo 4, **Implementación**, y los resultados obtenidos en el capítulo **5. Pruebas y resultados.**

A continuación, en los capítulos **6. Líneas futuras**. y **7. Conclusiones.**, se habla sobre líneas futuras y sobre las conclusiones de todo el trabajo respectivamente.

1.3. Marco regulador

En el presente apartado, se explicarán las regulaciones aplicables a este trabajo de fin de grado.

Uno de los aspectos de regulación importante es la base de datos usada. En nuestro caso, la base de datos se llama "Speech databases of typical children with SLI" ***[1]*** y es pública y abierta, con lo que se respetan los derechos de autor y se respeta la protección de datos de las personas que prestan sus voces para los audios existentes en la base de datos. En este caso, los audios son de niños, con lo que los permisos que tiene la base de datos son de los padres o tutores legales de estos niños.

En este trabajo se han usado las herramientas se Matlab, Excel y Word. En cuanto a Matlab, la licencia la proporciona la Universidad Carlos III a todos los alumnos. En lo que al paquete Office se refiere, el cual incluye las herramientas Excel y Word, también es la Universidad Carlos III de Madrid la que proporciona la licencia.

En cuanto a la redacción de esta memoria, cabe decir, que ha sido necesaria la lectura y búsqueda de información de diversas webs o *papers* de profesionales expertos en el tratamiento de voz o en el síndrome específico del lenguaje. Todo ello, debe quedar y ha quedado legalmente referenciado según detalla el Real Decreto Legislativo 1/1996 del 12 de abril ***[2]***.

En cuanto a leyes nacionales, cabe destacar la Ley Orgánica 15/1999, de 13 de diciembre*, de Protección de Datos de Carácter Personal* y a su Reglamento de desarrollo, aprobado por Real Decreto 1720/2007, de 21 de diciembre ***[2]***.

1.4. Entorno Socioeconómico

El trastorno específico del lenguaje se define como un trastorno que retrasa el aprendizaje del lenguaje, sin que el paciente sufra ninguna discapacidad que justifique dicha dificultad, como puede ser algún problema auditivo o discapacidad intelectual.

Según el *National Institute on Deafness and other Communication Disorders* ***[3]***, este síndrome afecta en torno del 7 al 8 por ciento de los niños en el jardín de la infancia.

Este síndrome, no sólo afecta a las personas en su etapa infantil, sino que tiene consecuencias en la edad adulta, pues una persona que tiene problemas en la comunicación en la niñez le costará aprender conocimientos básicos en la edad de desarrollo, lo que puede dar lugar a la aparición de problemas sociales y sufrir acoso o bullying.

Esto hace que la consecución del objetivo de la creación de un sistema que diagnostique con alta precisión si un niño tiene el síndrome específico del lenguaje o no lo tiene, mediante grabaciones de audio, haría posible que el diagnóstico se realizara de forma más rápida y que, en caso de que fuera necesario, se pudiese iniciar el tratamiento lo antes posible.

En resumen, el desarrollo de un sistema de este tipo podría tener un impacto de gran ayuda para los distintos sistemas sanitarios y para las personas que pudieran padecer el síndrome.

2. Estado del arte

En este apartado, se explicará lo que se conoce en la actualidad sobre el síndrome específico del lenguaje, que rumbo están tomando las investigaciones sobre el mismo y que técnicas se están utilizando para el diagnóstico del SLI.

Estudios de 2019 como el que publica la web "neuronas en crecimiento" ***[4]*** describe el síndrome específico del lenguaje como un trastorno del lenguaje que se identifica por un aprendizaje lento y retrasado respecto a la edad biológica del niño en concreto, sin tener esto relación con alguna otra deficiencia auditiva, cognitiva o de conducta.

En la actualidad no hay unos procedimientos estipulados para diagnosticar a ciencia cierta si un niño tiene el SLI o no. Se diagnostica según medidas no cuantitativas y con un grado de subjetividad alto, pues el SLI, actualmente, se diagnostica cuando el niño, a causa de su nivel de habilidades lingüísticas, no cumple con las expectativas sociales y educativas acordes a su edad.

Es decir, claramente, sería conveniente disponer de algún sistema que pueda ayudar a diagnosticar el SLI de una manera más objetiva y con rigor estadístico, de forma que esta herramienta sea un soporte al diagnóstico para el personal de sanidad, tal y como un traumatólogo se ayuda de una radiografía para diagnosticar una fractura.

En cuanto a las investigaciones interesantes e importantes que se están llevando a cabo en la actualidad sobre el SLI cabe destacar 3 de ellas ***[3]***:

- Investigación genética: Se ha detectado que una variante común en el sexto cromosoma humano puede tener una relación directa con el SLI. Esta variación en el sexto cromosoma también conlleva en muchos casos otros trastornos del lenguaje o del aprendizaje del lenguaje como pueden ser la dislexia o el autismo. Además, se están investigando otros genes posiblemente influyentes.

- Investigación sobre el bilingüismo: Se está detectando un mayor porcentaje de casos de niños que padecen SLI en niños con enseñanza bilingüe que en niños con un aprendizaje de una sola lengua.

- Investigaciones para el diagnóstico: Se está investigando para intentar obtener marcadores que den la alarma sobre un posible caso de SLI. Las investigaciones actuales se están centrando en la recopilación de datos para intentar obtener conclusiones. Estos datos son, entre otros, la capacidad del niño de realizar un seguimiento con los ojos, las condiciones neurofisiológicas y otras mediciones acordes al desarrollo cognitivo.

Aparte de estas tres líneas de investigación, hemos de mencionar el estudio en el que nos hemos basado, en el que se utiliza la misma base de datos que se ha usado para este trabajo y que también investiga un método basado en características acústicas de la voz para la detección del SLI. Este estudio se puede encontrar en la referencia ***[5]*** y termina con la siguiente conclusión, la cual en este trabajo se intenta ampliar (traducido del inglés):

“Los métodos descritos en este estudio fueron desarrollados para analizar los trastornos del habla en niños, específicamente en niños con problemas de lenguaje. La investigación se llevó a cabo durante 10 años. La descripción se centra en la clasificación, la recopilación de datos y el análisis de datos de estos niños. Para el análisis, sólo se utilizaron las habilidades del habla de los niños con SLI y se compararon con los niños sin SLI. El principal beneficio de este estudio incluye los métodos que se desarrollaron para clasificar a los niños con SLI basándose en el procesamiento directo de la base de datos”.

El primer método que se usa en este estudio, llamado análisis de errores, se basa en el número de errores de pronunciación en las elocuciones producidas por los niños. Una ventaja significativa es que su función no requiere métodos computacionales complejos y puede ser realizada por cualquier persona. Su desventaja fundamental es que no es un sistema automático. La tasa de éxito en la distinción entre niños con SLI y niños sin SLI fue del 93,81%. El segundo método se basa en el análisis de las características de la señal de habla de los dos grupos de locutores, con y sin SLI. En este caso, la tasa de éxito fue del 96,94%, y sólo tres de cada 98 participantes fueron clasificados como incorrectos. El tercer enfoque, basado en la duración de las elocuciones, verificó las hipótesis sobre diferencia de velocidad de procesamiento y respuesta para una serie de tareas entre los niños con y sin SLI. Los niños con SLI tienen una mayor duración de palabras que los niños sin SLI, es decir, siendo, en este caso, la diferencia del 27,51%.

Es importante remarcar que en este estudio el protocolo experimental no está completamente detallado, por lo que no nos ha sido posible replicar sus experimentos. En particular, no se indica los locutores ni los ficheros utilizados para entrenamiento y test, aunque es de suponer que los experimentos son dependientes del locutor.

En cualquier caso, según los autores del estudio, los resultados obtenidos demuestran que es posible identificar y distinguir claramente a los niños con SLI de los niños sin SLI.

3. Diseño de la Solución

3.1. Introducción

En este capítulo 3 de la memoria se planteará el diseño de la solución, sin entrar en los detalles de implementación que se desarrollarán en el capítulo siguiente, **Implementación**.

El diseño de la solución se enfoca con el siguiente objetivo: diseñar un sistema automático para la detección del síndrome específico del lenguaje en niños mediante el análisis de su voz y el uso de técnicas de aprendizaje automático.

Para que el lector pueda tener una idea general de lo que se explica en el capítulo, se llevará un orden claro y lógico sobre el diseño de la solución. Lo primero que se explicará, será cuál es la base de datos sobre la que se trabaja, así como su estructura inicial y el procesado que se ha hecho de la misma para obtener una estructura que interese para poder realizar un análisis completo del problema.

A continuación, con la base de datos ya dispuesta en la forma adecuada para trabajar sobre ella, se explicará cómo se han extraído los diferentes parámetros acústicos y cómo se han organizado los mismos para trabajar sobre ellos. Estos parámetros se han organizado de tal manera que se puedan hacer varias pruebas con varios subgrupos según interese.

Por último, con ayuda de técnicas de aprendizaje automático, se han entrenado varios modelos acústicos para cada uno de los subgrupos y se han realizado las pruebas pertinentes, en este caso, con las herramientas software Matlab y Excel. Con las pruebas ya realizadas se han comparado los resultados para obtener conclusiones.

El proceso por lo tanto es el siguiente:

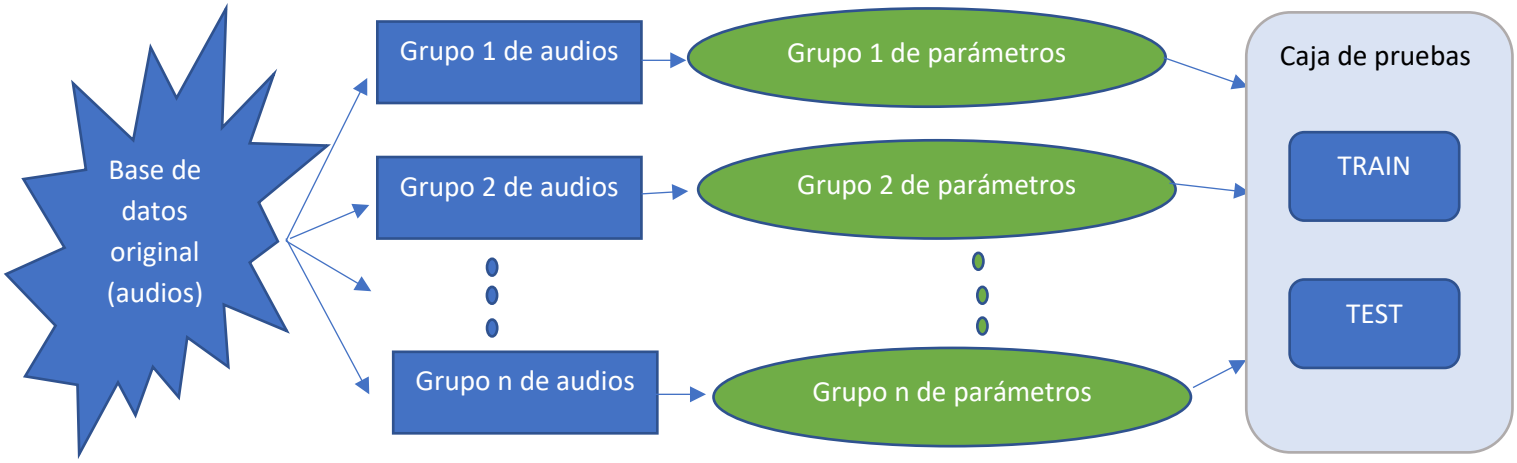

Figura 1: Diseño de la solución

Usando los distintos grupos de parámetros, algunos como parte de un entrenamiento para el aprendizaje máquina y otros como test, y a continuación analizando los resultados, es como se consigue llegar a conclusiones significativas.

3.2. Base de datos

En este apartado se explicará la base de datos utilizada y sus principales características, así como las modificaciones que se han realizado sobre la misma para adecuarla a la experimentación posterior.

La base de datos se llama "*Speech databases of typical children with SLI*" ***[1]*** . Esta base de datos es pública y abierta y se puede acceder a ella a través del enlace que se encuentra en las referencias de este mismo documento. Es una base de datos con ficheros de texto y ficheros de audio en formato WAV que corresponden a audios de voz de los respectivos pacientes y sujetos de control. Por matizar, los niños cuyas voces se pueden escuchar en estos audios son checos, con lo que los audios están grabados en este idioma.

3.2.1. Base de datos original

La distribución de la base de datos original es la siguiente: Dos carpetas, "*Healthy*" y "*Patients*", las cuales inmediatamente se renombraron a "control" y "pacientes".

Dentro de la carpeta de "*Healthy*" se encuentran 44 carpetas que se llaman H26, H27, H28… H69. Dentro de la carpeta "*Patients*" hay 54 carpetas que se llaman P8, P9…P61. Dentro de cada una de estas carpetas (H26, H56, P13, P56…) hay otras 7 carpetas en las cuales se encuentran ficheros de voz y ficheros de texto a modo de etiqueta. Los ficheros modo etiqueta de texto no se han utilizado porque contienen información relevante para otro tipo de estudios, pero que no tienen utilidad para el desarrollo del sistema implementado en este trabajo. Por otra parte, se eliminaron audios que había repetidos.

A continuación, se explica el contenido de cada una de estas 7 subcarpetas correspondientes a cada persona, ya sea control o paciente:

- 01SAMOHL: 5 audios del control o paciente, correspondientes a las 5 vocales a, e, i, o, u.
- 02SOUHL: 10 audios del control o paciente, correspondientes a las consonantes b, d, g, h, k, l, m, r, t, ch.
- 03_1SL: 9 audios correspondientes a los monosílabos: be, ber, krk, la, nos, pe, pro, prst, vla.
- 04_2SL: 5 audios correspondientes a los bisílabos kolo, papir, pivo, sokol, trdlo.
- 05_3SL: 4 audios correspondientes a 4 palabras de 3 sílabas: dedecek, pohadka, pokemon y kvetina.
- 06_4SL: 3 audios correspondientes a 3 palabras de 4 sílabas: motovidlo, televize y popelnice.
- 07_VSL: 2 audios correspondientes a 2 palabras de 5 sílabas: ruznobarevny y materidouska.

En total hay 38 audios por persona, ya sea paciente o control. Para contribuir a la comprensión del lector, a continuación, se añade una tabla que explica los audios que hay por persona y la traducción de las palabras al inglés. Dicha tabla ha sido extraída de un estudio que trabaja con la misma base de datos ***[5][2]***:

Task code	Description	# Patterns	Language	Utterances
[T1]	Vowels	5	Czech	*„a – e – i – o – u“*
			English	*„a – e – i – o – u“*
[T2]	Consonants	10	Czech	*„m - b - t - d - r - l - k - g - h – ch“*
			English	*„m - b - t - d - r - l - k - g - h – ch“*
[T3]	Syllables	9	Czech	*„pe - la - vla - pro - bě - nos - ber - krk – prst“*
			English	*„pe - la - vla - for - bě - nose - take - neck – finger“*
[T4]	Two-syllable words	5	Czech	*„kolo - pivo - sokol - papír – trdlo“*
			English	*„wheel - beer - falcon - paper – boob“*
[T5]	Three-syllable words	4	Czech	*„dědeček - pohádka - pokémon – květina“*
			English	*„grandfather - fairy tale - Pokemon – flower“*
[T6]	Four-syllable words	3	Czech	*„motovidlo - televize – popelnice“*
			English	*„niddy noddy - television – dustbin“*
[T7]	Five-syllable words	2	Czech	*„různobarevný – mateřídouška“*
			English	*„varicoloured – thyme“*

Tabla 1: Contenido por locutor de la base de datos.

En cuanto a la nomenclatura de los audios, después de cambiar algunas excepciones para que todos sigan el mismo patrón, es la siguiente:

- Audios controles: 4[identificador control mayúsculas] [número de tipo de audio (T1 – T7)] [identificador del contenido de audio en mayúsculas]
- Audios pacientes: 4[identificador paciente minúsculas] [número de tipo de audio (T1 – T7)] [identificador del contenido de audio en minúsculas]

De este modo, por ejemplo, el audio 4JR1A.wav representa al control con identificador JR, diciendo una vocal, por pertenecer al grupo T1 y más en concreto la vocal a.

Otro ejemplo sería el siguiente: El audio 4pme7mat.wav representa al paciente con identificador de persona pme, diciendo una palabra de 5 sílabas (por pertenecer al grupo 7), y más en concreto la palabra materidouska, cuyo identificador, como se verá más adelante en la tabla 8, es mat.

A continuación, se detallarán en forma de tabla, los identificadores de contenido de audio de los ficheros de la base de datos. Hay 7 tablas, una para cada grupo de palabras (vocales, consonantes, palabras monosílabas, palabras bisílabas, palabras trisílabas, palabras de 4 sílabas y palabras de 5 silabas). La primera fila de estas tablas indica lo que realmente se dice en el audio y la segunda fila indica el correspondiente identificador:

- Vocales:

a	e	i	o	u
a	e	i	o	u

Tabla 2: identificadores vocales.

- Consonantes:

m	b	t	d	r	l	k	g	h	ch
m	b	t	d	r	l	k	g	h	X

Tabla 3: Identificadores consonantes.

- Monosílabos:

pe	la	vla	pro	be	nos	ber	krk	prst
pe	la	vla	pro	be	nos	ber	krk	prst

Tabla 4: Identificadores monosílabos.

- Bisílabos:

kolo	pivo	sokol	papír	trdlo
kolo	pivo	sok	pap	trd

Tabla 5: Identificadores bisílabos.

- Trisílabos:

dedecek	pohádka	pokemon	kvetina
ded	poh	pok	kve

Tabla 6: Identificadores trisílabos.

- Palabras de 4 sílabas:

motovidlo	televize	popelnice
mot	tel	pop

Tabla 7: Identificadores palabras de 4 sílabas.

- Palabras de 5 sílabas:

materidouska	ruznobarevny
mat	ruz

Tabla 8: Identificadores palabras de 5 sílabas.

Para terminar de explicar la distribución de la base de datos, queda por destacar las excepciones de la misma. Y es que, si la mayoría de los sujetos de la base de datos tienen 38 audios, hay algunos que les faltan ciertos audios. A continuación, se citarán las excepciones en forma de tabla, llamando a cada locutor por su identificador y a cada audio por su identificador, los cuales están explicados en las tablas anteriores. Los audios que no están disponibles son:

IDENTIFICADOR DE LOCUTOR	IDENTIFICADOR DE AUDIO
pme	r
	mat
BU	VLA
HB	PIVO
KH	B
HVN	PE
KK	A
	BE
	VLA
KP	T
SS	I
	B
	NOS
TK	BE
VJ	M

Tabla 9: Audios de la base de datos que no están disponibles.

Por lo tanto, juntando todos los audios de pacientes suman un total de 2050 ficheros de audio y juntando todos los audios de los controles suman un total de 1659 ficheros de audio. Por tanto, se dispone para trabajar con un total de 3079 ficheros de audio, tal y como se representa en la figura siguiente:

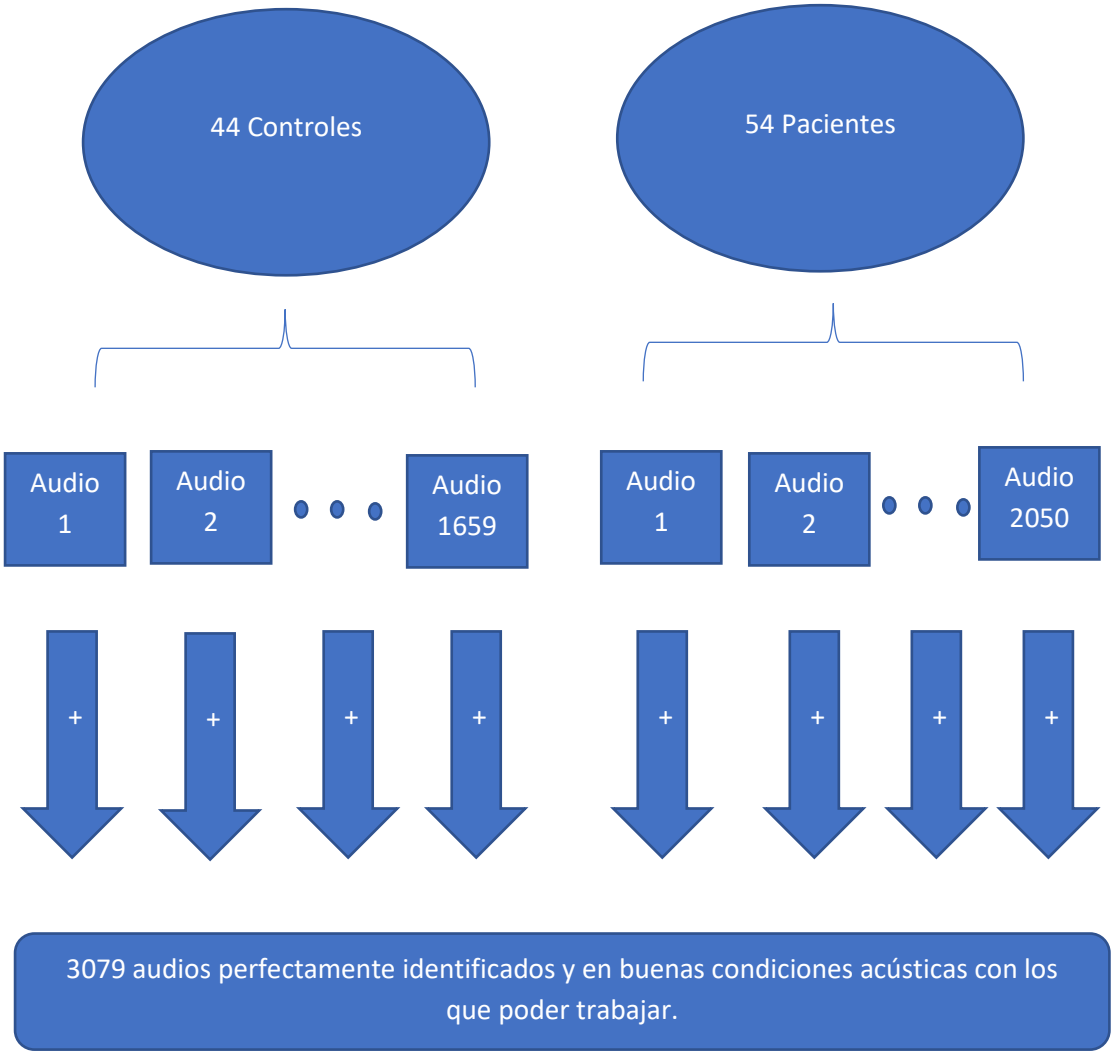

Figura 2: Distribución de audios en base de datos original.

3.2.2. Adecuación y división de la base de datos en subgrupos.

En el apartado anterior, se explica que la base de datos tiene en total 3079 audios comprobados y listos para trabajar sobre ellos, de los cuales 1659 pertenecen a controles y 2050 a pacientes.

Este apartado se centra en responder a una pregunta, ¿cómo se han organizado estos 3079 audios?

Pues bien, se decidió realizar dos tipos de experimentos, dependientes e independientes de locutor, por lo que se crearon 2 grandes grupos. Estos dos grupos contienen la totalidad de los audios cada uno (3079), pero los audios se distribuyen en subgrupos de manera diferente para el grupo dependiente del locutor que para el grupo independiente de locutor:

- **Dependientes del locutor**: En este grupo, a la hora de realizar las pruebas, se usan algunos audios de cierto paciente o control para entrenar el modelo correspondiente y los audios restantes de ese mismo paciente o control se usan para test.

- **Independiente del locutor**: En este grupo, todos los audios de cada control o paciente, o bien están incluidos en el conjunto de entrenamiento o bien en el conjunto de test. Es decir, una parte de los locutores pertenecen el conjunto de entrenamiento mientras que los restantes pertenecen al conjunto de test.

Dentro de estos dos grandes grupos, como ya se ha explicado, se crean otros dos subgrupos. Un subgrupo será destinado a entrenar el modelo acústico (grupo de entrenamiento o *train*) y el otro subgrupo será destinado a la clasificación en sí (grupo de prueba o test).

Con esta idea, se dispusieron por un lado todos los audios de pacientes en orden alfabético, y por otro lado todos los audios de controles, también alfabéticamente. De este modo, se procedió a crear los cuatro subgrupos que quedan de la siguiente forma:

- **Subgrupo 1, independiente de locutor train**: En esta carpeta se incluyen todos los audios de los 32 primeros pacientes (alfabéticamente hasta el paciente con identificador pmi) y todos los audios de los 30 primeros controles (alfabéticamente hasta el control con identificador NH). Consta de un total de 2346 audios.
- **Subgrupo 2, independiente del locutor test**: En esta carpeta, se incluyen todos los audios de los 22 últimos pacientes (alfabéticamente desde el paciente con identificador pmo hasta el último) y todos los audios de los 14 últimos controles (alfabéticamente desde el control con identificador PC hasta el último). Consta de un total de 1363 audios.
- **Subgrupo 3, dependiente del locutor train**: En esta carpeta se incluyen audios de 3 vocales, 6 consonantes, 6 monosílabos, 3 bisílabos, 2 trisílabos, 2 palabras de 4 sílabas y una palabra de 5 sílabas por paciente y 3 vocales, 7 consonantes, 7 monosílabos, 3 bisílabos, 3 trisílabos, 2 palabras de 4 sílabas y una palabra de 5 sílabas por control. Se presentan las siguientes excepciones: El paciente pme tendrá una consonante menos, el control BU tendrá un monosílabo menos, el control HB tendrá un bisílabo menos, el control KH tendrá una consonante menos, el control HVN tendrá un monosílabo menos, el control KK tendrá una vocal menos y dos monosílabos menos, el control KP tendrá una consonante menos, el control SS tendrá una vocal menos, una consonante menos y un monosílabo menos, el control TK tendrá un monosílabo menos y el control VJ tendrá una consonante menos. Consta de un total de 2372 audios.
- **Subgrupo 4, dependiente del locutor test**: En esta carpeta se incluyen audios de 2 vocales, 4 consonantes, 3 monosílabos, 2 bisílabos, 2 trisílabos, 1 cuatrisílabo y una palabra de 5 sílabas por paciente y 2 vocales, 3 consonantes, 2 monosílabos, 2 bisílabos, 1 trisílabos, 1 palabra de 4 sílabas y una palabra de 5 sílabas por control con la siguiente excepción: El paciente pme no tendrá ninguna palabra de 5 sílabas. Consta de un total de 1337 audios.

Con estos 4 subgrupos, es con los que se trabaja y se realizan los entrenamientos o pruebas (test) pertinentes según corresponda.

3.2.3. Justificación de la elección de la base de datos y su distribución.

En este apartado se explicará el por qué se dan todos los pasos detallados en el punto 3.2. de esta memoria, **Base de datos**, así como en sus subapartados 3.2.1, **Base de datos original**, y **3.2.2. Adecuación y división de la base de datos en subgrupos.** Se seguirá el mismo orden para una mejor comprensión.

Para empezar, se explicará el por qué se escoge la base de datos "*Speech databases of typical children with SLI*" previamente especificada y referenciada, así como el porqué de la modificación inicial a la que fue sometida por parte del alumno. También se detallará cómo se decidió la división de los distintos grupos y subgrupos para su posterior análisis.

La elección de la base de datos "*Speech databases of typical children with SLI*" se debe principalmente a que es una base de datos con dimensión suficiente como para poder trabajar sobre ella con cierta fiabilidad en los resultados, pues son 3709 audios de 44 controles y 54 pacientes. Estos son números suficientes para poder hacer varios tipos de pruebas y hacer comparación entre ellas. Este número de audios también es suficiente para que, si hay algún *outlier*, este no impacte significativamente en nuestro modelo.

La otra gran razón por la que se ha elegido la base de datos es porque esta es abierta y pública, lo cual hace que no haya que solicitar permisos expresos, etc. Es muy difícil encontrar una base de datos de audios de niños para este síndrome en concreto y a la vez es imprescindible para el trabajo que los audios sean de niños, pues es a temprana edad cuando interesa identificar con velocidad y comodidad si existe en la persona este síndrome o no, para poder tomar medidas desde pequeños.

En cuanto a la división de los audios en grupos y subgrupos, la justificación es la siguiente: Se crean dos grandes grupos para pruebas, dependiente del locutor e independiente del locutor. Se hace esta clara separación porque cabe la posibilidad de que el entrenamiento se vea altamente afectado por la dependencia del locutor. Es decir, cabe pensar que, si entrenando un modelo en el cual se encuentran 2 vocales de un paciente, 5 consonantes etc. y para testear, se usan las 3 vocales restantes de este paciente, las 5 consonantes restantes de este paciente etc., el modelo de clasificación futuro identifique personas y no el síndrome específico del lenguaje. De esta forma, nos ha sido posible medir el impacto de la independencia del locutor sobre nuestro sistema.

La siguiente división es más evidente, dentro de los dos grandes grupos, dependiente del locutor e independiente del locutor, hay que hacer para cada grupo una división más, para que podamos tener un grupo para entrenar el modelo y un grupo para testear el modelo. Por ello se crean los subgrupos: dependiente del locutor train, dependiente del locutor test, independiente del locutor train e independiente del locutor test.

La cantidad de audios usados para train y la cantidad de audios usados para test se ha procurado que sean en torno a un 75-80 % de train y el resto de test, siendo el 100% los 3079 audios limpios. En general, para un sistema de clasificación, cuantos más datos se utilicen para entrenarlo, las prestaciones serán mejores. Por otra parte, para testear el funcionamiento del sistema se ha decidido utilizar como mínimo el 20% de los audios para obtener resultados fiables, con objeto de evitar la "fortuna o mala suerte" de acertar o errar en el test si se testea para escasos datos.

3.3. Extracción de parámetros.

En este apartado se explicará qué parámetros acústicos se extraen de la base de datos anteriormente descrita.

La extracción de parámetros o características acústicas tiene como objetivo la obtención de una representación compacta de la señal de voz que sea discriminativa, es decir, que sirva para distinguir los sujetos que sufren síndrome específico del lenguaje (pacientes) de aquellos que no lo padecen (controles).

Los parámetros escogidos para la consecución de los objetivos del planteamiento del problema son los coeficientes mel-cepstrales (*Mel-Frequency Cepstrum Coefficients*, MFCC).

Los coeficientes MFCC se extraen a través de los siguientes procesos:

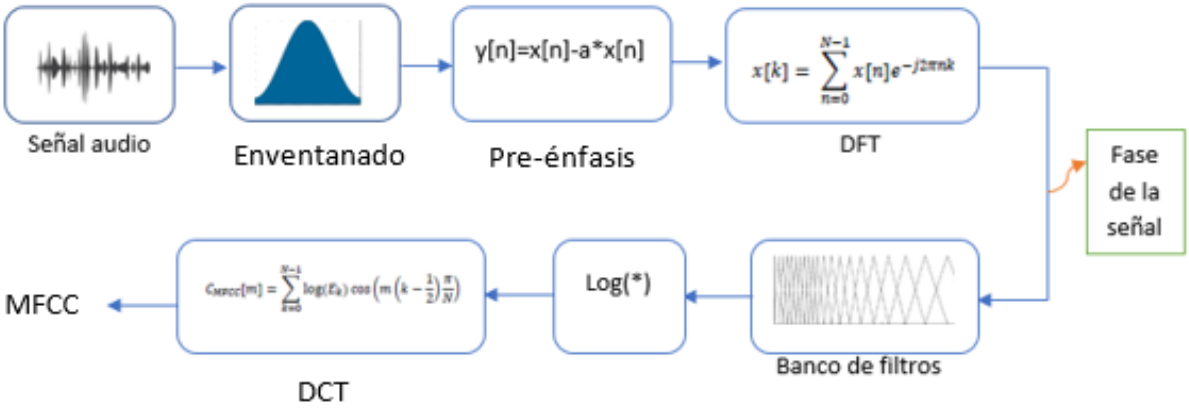

Figura 3: Extracción de los MFCC.

A continuación, se detalla cada uno de los módulos de la **Figura 3** explicando todo el proceso.

En primer lugar, la señal de voz se divide en ventanas con objeto de realizar un análisis frecuencial a corto plazo. En el caso de este trabajo, el enventanado es con ventanas de tipo Hanning. A continuación, la señal pasa por un filtro de preénfasis que enfatiza las altas frecuencias con objeto de compensar el contenido en bajas frecuencias de la señal de voz.

$$y[n] = x[n] - a*x[n-1] \qquad 0{,}95 \leq a \leq 0{,}98$$

Ecuación 1: Preénfasis

En la **Ecuación 1**, x[n] se refiere a la señal que llega al módulo de preénfasis, el parámetro "a" es un escalar y la señal y[n] es la señal que sale del módulo. Esta señal es en vista de la ecuación, para cada muestra n, la resta de x[n] y el valor de la señal justo anterior a k (es decir x[n-1]), pero previamente atenuado por el escalar a.

Una vez llegada aquí, la señal se somete a la transformada discreta de Fourier a través de su ecuación:

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}$$

$$0 \leq k \leq N$$

Ecuación 2: Transformada discreta de Fourier.

La **Ecuación 2** aplicada al problema que se está tratando realiza la transformada discreta de Fourier sobre x[n], siendo x[n] la señal entrante al módulo DFT que se puede ver en la**Figura 3** y N es el número total de muestras de la señal. Por lo tanto, X[k] es el valor resultante a la salida del módulo DFT para cada valor de frecuencia k.

En este punto, ya podemos desechar la fase de la transformada de Fourier y trabajar solo con su módulo. Dicho módulo pasa por un banco de filtros triangulares espaciados según la escala Mel. A continuación, se opera haciendo el logaritmo de las energías de las salidas de cada uno de los filtros. Así llegamos al último paso, la transformada discreta del coseno (*Discrete Cosine Transform*, DCT), la cual decorrela las energías en banda obtenidas en el paso anterior. La DCT a través de su ecuación, transforma los coeficientes espectrales al dominio cepstral, de los que, extrayendo los 12 primeros coeficientes, se obtienen los parámetros MFCC deseados.

$$C_{MFCC} = \sum_{k=0}^{F-1} \log(E_k) \cos\left(m\left(k - \frac{1}{2}\right)\frac{\pi}{N}\right) \qquad m = 1, \ldots\ldots, M$$

Ecuación 3: Coeficientes MFCC

En esta **Ecuación 3**, F es el número de energías en banda (número de filtros en escala mel), E_k es la energía de la señal a la salida del filtro k-ésimo, M es el número de coeficientes MFCC a extraer, y $C_{MFCC}[m]$ es el valor del coeficiente MFCC m-ésimo.

Con estos 12 coeficientes MFCC y un parámetro número 13 que es la log-energía de la señal en cada trama, se realizan la mitad de las pruebas que se explican en este documento. Para la otra mitad, se usan además de estos 13 parámetros, los coeficientes deltas de los mismos, los cuales llamaremos delta-MFCC, que son la derivada de estos 13 coeficientes.

Con estos 13 MFCC y estos 13 delta-MFCC tenemos 26 parámetros con los que trabajar. Para poder realizar las pruebas, se generan ficheros de texto que almacenan los parámetros de cada audio. Estos ficheros de texto son los que se usarán para el entrenamiento o para la clasificación según el rol que le toque jugar a cada audio en la máquina de predicción que se pretende crear.

3.3.1. Justificación de la elección de los parámetros MFCC.

Para el entrenamiento y el posterior test, se han escogido los parámetros MFCC y los delta-MFCC pudiendo haber escogido otros parámetros. ¿Qué favorece la elección de los MFCC?

Está contrastado por otros estudios profesionales y docentes que los parámetros MFCC representan de manera compacta la información del sonido en la voz humana ***[6]***. Está muy extendido el uso de los MFCC para reconocimiento de voz, para reconocimiento de idioma, etc.

Esto se debe principalmente a que, para la extracción de estos coeficientes cepstrales se utiliza la escala de Mel. La escala de Mel se asemeja a la forma de percibir el sonido por parte del oído humano, dando más resolución a las bajas frecuencias que a las altas.

Además, la función que se ha usado para la extracción de los MFCC se ha escogido porque sus bloques, representados en la **Figura 3**, procesan el audio de tal forma, que facilitan que los parámetros sean adecuados para las pruebas. Los parámetros MFCC se extraen siguiendo las acciones que se detallan a continuación:

- El preénfasis compensa una atenuación de 20dB/década característica del habla humana.
- El enventanado, se realiza de tal forma que no se pierda información alguna, pues las ventanas, como se explicará en el apartado de implementación, están solapadas.
- El banco de filtros triangulares es acorde a la escala de frecuencias de Mel. Esta es una de las grandes causas de elegir la función melcepst para extraer los parámetros.
- La decorrelación llevada acaba por la DCT hace que las componentes de los parámetros MFCC sean independientes unas de otras, lo que suele facilitar el proceso de entrenamiento.

En cuanto a los parámetros delta-MFCC, se decide incluirlos en el sistema para que el análisis y el entrenamiento sea más completo, pues los delta-MFCC contienen información sobre la evolución temporal de los MFCC. Se ha comprobado que esta información es importante en otros sistemas basados en voz, como los de reconocimiento automático del habla.

3.4. Sistema de clasificación

En este apartado se detallarán los datos que han sido destinados para train y los datos que han sido destinados para test. Además, se explicará el modelo de entrenamiento al que han sido sometidos los datos de train.

3.4.1. Grupos de train y modelo de entrenamiento

En este apartado se explica qué conjuntos de datos han sido utilizados para el proceso de entrenamiento, para conformar una idea de las pruebas realizadas que se detallarán a posteriori. Dichos conjuntos de datos de entrenamiento han sido:

- Conjunto de parámetros MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**.: independiente del locutor train.

- Conjunto de parámetros MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**.: dependiente del locutor train.
- Conjunto de parámetros MFCC + delta-MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**.: independiente del locutor train.
- Conjunto de parámetros MFCC + delta-MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**.: dependiente del locutor train.
- Conjunto de parámetros MFCC, previa extracción de la log-energía, correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**.: independiente del locutor train.
- Conjunto de parámetros MFCC, previa extracción de la log-energía, correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos.**: dependiente del locutor train.

Para crear un modelo acústico adecuado a las características de nuestros parámetros se ha usado Matlab, más en concreto, con la función fitcsvm***[7]***, la cual se explicará en el apartado de implementación cómo se ha usado y con qué parámetros.

La función fitcsvm entrena un modelo binario a través de una máquina de vectores soporte (*Support Vector Machine*, SVM) usando funciones tipo Kernel.

Las funciones tipo Kernel usadas en el caso de este trabajo han sido las siguientes:

- Función gaussiana.
- Función lineal.
- Función polinómica de grado 2.
- Función polinómica de grado 3.

Estas 4 funciones tipo Kernel se han usado en el entrenamiento de los grupos de train previamente descritos. Cada uno de estos grupos se han entrenado con todas las funciones de Kernel citadas, con lo que el número total de modelos de entrenamiento con los que se ha trabajado ha sido de: 6 x 4 = 24 modelos.

3.4.2. Justificación de la elección de los grupos train y de los modelos de entrenamiento

Cómo se ha visto en el apartado anterior, en total se han creado 24 modelos para la futura clasificación de los audios de los grupos test. Tal diversidad se debe a la búsqueda del modelo más adecuado para la posterior clasificación de un audio. En las pruebas se verá cual es el mejor modelo, es decir, el modelo con más probabilidad de acierto entre paciente o control, así como la dependencia de las condiciones de los audios en qué modelo tenga más probabilidad de acierto.

Para la creación de estos modelos se ha pensado en factores que podrían ser determinantes. Estos factores son los siguientes:

DEPENDENCIA DEL LOCUTOR

Por ello se crean dos grandes grupos de audios:

- dependientes del locutor
- independientes del locutor.

ELECCIÓN DE PARÁMETROS

Y la elección que se ha llevado a cabo ha sido la siguiente:

- Parámetros MFCC
- Parámetros MFCC + delta-MFCC

PARÁMETRO LOG-ENERGÍA

Se ha pensado que el parámetro de la log-energía podría tener una alta importancia. Por ello, también se han creado modelos con la previa extracción de la log-energía en los grupos de train.

FUNCIÓN KERNEL

Así como los 3 factores anteriores tienen más que ver con la creación de distintos grupos de train para entrenar a los distintos modelos o con la extracción de los parámetros acústicos correspondientes, este factor es puramente del modelo de entrenamiento de los datos. En el caso de este trabajo se ha decidido escoger 4 tipos de funciones Kernel para determinar la más apropiada para nuestra tarea:

- Gaussiana
- Lineal
- Polinómica de grado 2.
- Polinómica de grado 3.

3.4.3. Grupos de test

Este apartado trata de explicar el siguiente paso a seguir partiendo de la base de que ya se posee un modelo adecuado para aplicar a los audios de los grupos de test. Dicho paso consiste en aplicar los modelos generados previamente a los ficheros de audio de test. Los grupos de audios que se van a someter a test son los siguientes:

- Conjunto de parámetros MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**: independiente del locutor test.
- Mismo grupo de audios que en el punto anterior, pero añadiendo además los parámetros delta-MFCC.
- Conjunto de parámetros MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**: dependiente del locutor test.

- Mismo grupo de audios que en el punto anterior, pero añadiendo además los parámetros delta-MFCC.
- Conjunto de parámetros MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**: independiente del locutor test, pero sin el parámetro de la log-energía.
- Conjunto de parámetros MFCC correspondientes al grupo de audios, especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**: dependiente del locutor test, pero sin el parámetro de la log-energía.
- Conjunto de parámetros MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**: independiente del locutor test, pero sin el parámetro de la log-energía y previo paso por una función de Matlab la cual elimina silencios, vadsohn ***[8]*** .
- Conjunto de parámetros MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**: dependiente del locutor test, pero sin el parámetro de la log-energía y previo paso por la función vadsohn.
- Conjunto de parámetros MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**: dependiente del locutor test, previo paso por una función que añade ruido llamada v_addnoise ***[8]*** . En este caso, con los parámetros adecuados de la función, se consigue una SNR de -5dB para cada audio, siendo el mismo ruido para todos los casos. El ruido es de habitación y lo podemos encontrar en la base de datos abierta DEMAND ***[9]*** , en el fichero ch1.wav de DLIVING_16k.zip.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 0 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 5 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 10 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 15 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 20 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de -5 dB y usando como ruido el fichero ch1.wav de OOFICE_16K.zip de la misma base de datos, DEMAND. Este ruido corresponde al del interior de una oficina.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 0 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 5 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 10 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 15 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 20 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de -5 dB y usando como ruido el fichero ch1.wav de OMEETING_16K.zip de la misma base de datos, DEMAND. Este ruido corresponde al ruido en el transcurso de una reunión o de una conversación.

- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 0 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 5 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 10 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 15 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 20 dB.
- Conjunto de parámetros MFCC más los parámetros delta-MFCC correspondientes al grupo de audios especificado en el capítulo **3.2.2. Adecuación y división de la base de datos en subgrupos**: dependiente del locutor test, previo paso por una función que añade ruido llamada v_addnoise ***[8]*** . En este caso, con los parámetros adecuados de la función, se consigue una SNR de -5dB para cada audio, siendo el mismo ruido para todos los casos. El ruido es de habitación y lo podemos encontrar en la base de datos abierta DEMAND ***[9]*** , en el fichero ch1.wav de DLIVING_16k.zip.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 0 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 5 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 10 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 15 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 20 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de -5 dB y usando como ruido el fichero ch1.wav de OOFICE_16K.zip de la misma base de datos, DEMAND. Este ruido corresponde al del interior de una oficina.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 0 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 5 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 10 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 15 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 20 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de -5 dB y usando como ruido el fichero ch1.wav de OMEETING_16K.zip de la misma base de datos, DEMAND. Este ruido corresponde al ruido en el transcurso de una reunión o de una conversación.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 0 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 5 dB.
- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 10 dB.
- Parámetros MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 15 dB.

- Parámetros MFCC más los parámetros delta-MFCC del mismo grupo de audios que el punto anterior, pero con una SNR de 20 dB.

Cómo se puede apreciar, hay un gran número de grupos test con los que realizar pruebas con el objetivo de tener una mayor fiabilidad estadística en su posterior análisis, así como diversidad en el análisis.

Una vez citados todos los grupos de test, para una mayor comprensión, se va a realizar un resumen para que se pueda tener una idea global de todos los grupos existentes.

Experimentos dependientes de locutor:

- Condiciones limpias

- Condiciones limpias sin el parámetro de la energía.

- Condiciones limpias sin el parámetro de la energía y con eliminación de silencios.

- Condiciones ruidosas con ruido en el salón de una casa (living) y 6 SNRs distintas (de -5 dB a 20 dB en pasos de 5 dB).

- Condiciones ruidosas con ruido en una sala de reuniones (meeting) y 6 SNRs distintas (de -5 dB a 20 dB en pasos de 5 dB).

- Condiciones ruidosas con ruido en una oficina pequeña (office) y 6 SNRs distintas (de -5 dB a 20 dB en pasos de 5 dB).

Y los mismos puntos que se han detallado para los experimentos dependientes del locutor se repetirán para los experimentos independientes de locutor.

3.4.4. Justificación de la elección de los grupos de test.

Tanto para dependiente del locutor cómo para independiente del locutor se crean varios grupos cómo ya hemos visto en el apartado anterior. A continuación, se explica la razón de crear esos grupos:

AUDIOS LIMPIOS

Las pruebas iniciales se realizan en condiciones óptimas. Para ello, se utilizan los audios limpios, es decir, sin ningún tipo de distorsión o ruido añadido.

AUDIOS LIMPIOS SIN EL PARÁMETRO DE LA LOG-ENERGÍA

Estos grupos test se crean para medir el impacto de la log-energía en el proceso de clasificación.

AUDIOS LIMPIOS SIN EL PARÁMETRO DE LA LOG-ENERGÍA Y SIN SILENCIOS

Estos grupos se crean a modo de comprobación, para descartar o avanzar en el problema si los segmentos de silencio son importantes en la detección del síndrome. En este caso, no se considera el parámetro de log-energía.

CONDICIONES RUIDOSAS

Si se quiere utilizar el sistema en un escenario real y fuera de laboratorios acústicos específicamente preparados, se necesita estar prevenido de posibles condiciones ruidosas y medir cuánto afecta el ruido a las pruebas realizadas. Por ello se ha decidido someter a los audios en condiciones limpias a ciertas condiciones ruidosas que se podrían dar en la vida real cuando unos padres graban a su hijo un audio para enviarlo al sistema de clasificación automática. Las condiciones de ruido que más se pueden asimilar a escenarios reales pueden ser las de una habitación, una sala de reunión o una oficina. Por eso se han escogido estas tres condiciones de ruido. Además, se han creado grupos de test para estas 3 condiciones con distintos valores de relación señal a ruido (*Signal-to-Noise Ratio*, SNR) para poder estimar el mínimo valor de SNR a partir del cual los resultados del sistema son aceptables.

4. Implementación

En este capítulo se describe cómo se ha conseguido implementar el diseño de la solución, como se ha implementado la realización de las pruebas y como se ha implementado la obtención de los resultados finales.

La mayoría de las implementaciones de las distintas fases del problema se han desarrollado con la herramienta de programación Matlab, ya que dicha herramienta posee la capacidad para gestionar matrices de grandes dimensiones, posee funciones capaces de entrenar modelos de clasificación binaria, y existen *toolbox* de Matlab capaces de gestionar, transformar y extraer características de ficheros de voz. Todas estas características de Matlab son necesarias para implementar la solución del problema.

En cuanto a la obtención de los resultados se ha usado la herramienta Excel, la cual ha permitido almacenar los datos que se iban extrayendo de Matlab y usar los mismos para la generación de gráficos y tablas, con los que visualizar los resultados finales.

4.1. Implementación del manejo de la base de datos.

Para ordenar y distribuir la base de datos se usó el explorador de archivos de Windows 10, para almacenar los ficheros de audio en distintas carpetas según las necesidades del problema.

A partir de ahí, para poder transferir los ficheros de audio de una carpeta en concreto al entorno Matlab, se crea una función de inicialización llamada lectura.m, incluida en **9. Anexos**, la cual, lee un fichero de texto (creado previamente en cada carpeta) que contiene 2 columnas: la primera contiene los nombres de los ficheros de audio ordenados alfabéticamente y la segunda contiene etiquetas a modo de número entero. La etiqueta será 0 si el audio de su línea es un paciente y será 1 si el audio de su línea es un control.

Con esto, se crean dos vectores que se almacenarán en Matlab, uno con todos los nombres de los audios de la carpeta y otro con las etiquetas de los mismos.

4.2. Implementación de la extracción de los parámetros MFCC.

Para extraer y almacenar los parámetros MFCC de cada fichero de audio se crea una función Matlab llamada escritura_mfcc.m, incluida en **9. Anexos**, la cual consta de tres fases que se repite en bucle hasta escribir en un fichero de texto distinto para cada audio los parámetros MFCC trama a trama:

1. Lectura y almacenamiento del fichero de audio en Matlab.
2. Extracción y almacenamiento de los parámetros MFCC de cada fichero de audio.
3. Escritura de los parámetros MFCC extraídos en el paso 2 en un fichero de texto con el mismo nombre que el audio original y distinta extensión.

El paso 1 se implementa principalmente mediante la función audioread.m ***[10]*** de Matlab la cual pasándole como parámetro el nombre de un fichero de audio, devuelve un vector con las muestras de audio contenidas en dicho fichero para poder ser interpretado por Matlab y la frecuencia de muestreo. Cabe destacar que entre el paso 1 y el 2, el audio se remuestrea a 16000 Hz para que todos los audios tengan la misma frecuencia de muestreo.

El paso 2 se implementa principalmente mediante la función melcepst.m perteneciente a la *voicebox* de Matlab ***[8]*** . Esta función melcepst.m tiene la siguiente sintaxis: [c] = melcepst (s, fs, w, nc, p, n, inc).

Como se puede observar, la función tiene varias entradas y una salida que se detallan a continuación:

ENTRADAS:

- s: El audio en concreto del que se quieren obtener los parámetros. En este caso el obtenido del paso 1 de la función audioread.m.
- fs: Frecuencia de muestreo en Hz. En este caso la frecuencia de muestreo usada para todo el problema es 16000 Hz.
- w: Tipo de enventanado. En este caso ventana tipo Hanning añadiendo la energía, 'E'.
- nc: Número de coeficientes MFCC. En este caso 12, pues la energía es adicional.
- p: Número de filtros. En este caso 40 (se decide usar este estándar).
- n: Longitud de la ventana de análisis en muestras. En este caso 20 ms multiplicado por la frecuencia de muestreo (fs).
- inc: periodo de trama (indica cada cuántas muestras se calculan los parámetros sobre ventanas de análisis de longitud n). En este caso 10 ms multiplicado por la frecuencia de muestreo, es decir, se decide que inc es la mitad que n.

SALIDAS:

Como salida, se obtiene la matriz c, la cual consta de 13 columnas (cada columna corresponde a un parámetro mfcc, más la log-energía) y tantas filas como muestras tenga el audio dividido del valor "inc" (por tanto, el número de filas corresponde con el número de tramas de audio, "ntramas").

El paso 3 se implementa principalmente mediante la función de Matlab fprintf.m ***[10]***, la cual, aplicada al presente problema, escribe en un fichero de texto lo almacenado por Matlab en la matriz c obtenida en el paso 2.

Por último, cabe destacar, que esta función escritura_mfcc.m que se ha explicado en el presente apartado, según los parámetros que se han deseado guardar en los ficheros de texto, ha sufrido algunas modificaciones. Por ejemplo, cuando se ha deseado obtener los parámetros delta-MFCC además de los MFCC o cuando se ha deseado añadir algún tipo de ruido a los audios de un determinado grupo de train.

Para obtener los parámetros MFCC, la función melcepst se ejecuta de la siguiente manera: melcepst (aud, 16000, 'E', 12, 40, 0'02 x 16000, 0'01 x 16000).

Para obtener los parámetros MFCC más los delta-MFCC, la función melcepst se ejecuta añadiendo en el tipo de enventanado una "d" que indica que se deben añadir los parámetros delta: melcepst (aud, 16000, 'dE', 12, 40, 0'02 x 16000, 0'01 x 16000).

Para añadir ruido se usa la función addnoise.m de Matlab antes de usar melcepst de la siguiente manera: aud = v_addnoise (aud, Fs, snr, 'k', xn, Fs). Siendo Fs nuestra frecuencia de muestreo y con la que se muestreará tanto el audio original como el ruido que se quiera añadir (xn). La snr, dependerá de la snr que se desee para según que prueba y el parámetro 'k' indica que se mantenga la potencia de la señal original.

4.3. Implementación del entrenamiento de los distintos modelos.

Para realizar el entrenamiento de un modelo, se necesita unos datos de entrada con una estructura apropiada. La estructura base elegida ha sido una matriz, en la que cada fila represente un audio en concreto y cada columna represente un parámetro MFCC. Así, por ejemplo, si se desea entrenar al subgrupo 1, independiente del locutor train, el cual contiene 2346 audios, se necesitará obtener una matriz de dimensiones (2346 x 13).

Sin embargo, según lo comentado en el capítulo anterior, hasta este punto de la implementación del problema, se posee una matriz distinta para cada audio. Para combinar los parámetros MFCC a nivel de trama de cada fichero en un único vector acústico y concatenar dichos vectores acústicos en una única matriz con las dimensiones adecuadas, se procede de la siguiente manera:

Se ha creado la función de Matlab matriz.m, situada en **9. Anexos**, la cual, recorre una a una las matrices que contienen los parámetros MFCC de cada audio, realizando la media de cada uno de estos parámetros MFCC. De esta manera, para cada audio, se obtiene una representación vectorial (1x13) de los parámetros MFCC y no matricial (ntramas x 13), que es lo que se tenía hasta este punto.

De esta manera, concatenando estos vectores uno debajo del otro obtenemos la matriz deseada. Esta matriz será nuestra matriz de datos de entrada para entrenar al modelo.

Esta función matriz.m, se ha implementado también con algunas variaciones, para poder obtener matrices de 12 columnas cuando se ha deseado obviar el parámetro log-energía o para poder obtener matrices de 26 parámetros cuando se ha deseado utilizar los parámetros MFCC y los parámetros delta-MFCC.

Una vez que ya se tienen los datos de entrada, para entrenar el modelo se ha usado la función fitcsvm ***[7]***.

La función fitcsvm entrena un modelo binario a través de una máquina de vectores soporte (*Support Vector Machine*, SVM) usando funciones tipo Kernel.

Las funciones tipo Kernel usadas en el caso de este trabajo han sido las siguientes:

- Función gaussiana.
- Función lineal.
- Función polinómica de grado 2.
- Función polinómica de grado 3.

Exceptuando el tipo de función y los datos de entrada, el resto de los parámetros de la función fitcsvm se ha mantenido fijo para todas las pruebas.

- Standardize: true (Indicador para estandarizar los datos del predictor. Escribiendo true, el software centra y escala cada variable predictiva (X o Tbl) según la media ponderada de columna correspondiente y la desviación estándar.)
- OptimizeHyperparameters: auto (Optimiza los parámetros usando *BoxConstraint* y la escala de Kernel)
- ClassName: [0,1] Que representan a pacientes y controles respectivamente.
- HyperparameterOptimizationOptions: struct ('AcquisitionFunctionName', 'expected-improvement-plus') (Con este parámetro se especifica la función para elegir el siguiente punto de evaluación).

4.4. Implementación de la realización de la clasificación.

Una vez entrenado el modelo, como se ha explicado en el apartado anterior, queda realizar la clasificación para un grupo test, lo cual se explicará en este apartado, y extraer y visualizar los resultados que se verá en el siguiente apartado.

La manera con la que se ha implementado esta clasificación es la siguiente: Primero, a través de la función matriz.m previamente explicada, obtenemos la matriz de parámetros de un grupo de test. Una vez obtenida esta matriz, para realizar la clasificación se ha usado la función predict.m de Matlab ***[10]***, la cual, pasándole como datos de entrada un modelo entrenado y un grupo de test sobre el que realizar la clasificación, devuelve un vector de etiquetas tamaño (1xnfich) siendo nfich el número total de casos a clasificar. Cada una de las etiquetas del vector corresponde, en el caso de este problema, a cada uno de los audios del grupo de test expuesto a clasificación, y las etiquetas serán o bien un "0" o bien un "1", es decir, o paciente o control.

4.5. Implementación de la obtención y visualización de los resultados.

En este apartado se explicará cómo se ha implementado la extracción de los datos procedentes de la clasificación, como se han procesado estos datos para la obtención de los resultados correspondientes y cómo se ha implementado la visualización de dichos resultados. Para todo ello se ha usado la herramienta de hojas de cálculo Excel.

Para extraer los datos procedentes de la clasificación simplemente se ha copiado el vector de clasificación y se ha pegado en una columna de una hoja de cálculo. Dos columnas a la izquierda de esta columna, se ha escrito el nombre de los audios correspondientes al grupo test sometido a clasificación, y entre medias de estas dos columnas, es decir, la tercera, se han escrito las etiquetas reales de cada audio. Cómo se muestra en la siguiente figura**:**

1	AUDIOS	Etiqueta real	Predicción	Acierto/error	VP	FP	VN	FN
2	4PC1A.wav	1	1	1	0	0	1	0
3	4PC1E.wav	1	1	1	0	0	1	0
4	4PC1I.wav	1	1	1	0	0	1	0
5	4PC1O.wav	1	1	1	0	0	1	0
6	4PC1U.wav	1	1	1	0	0	1	0

Figura 4

Las columnas 4, 5, 6, 7 y 8 de la **Figura 4** son relaciones entre las columnas "Etiqueta real" y "Predicción" que servirán para obtener los valores necesarios para el cálculo de los resultados finales.

Estas columnas se han implementado mediante las siguientes funciones condicionales (n la memoria están escritas en pseudocódigo y no en código Excel):

- Columna 4 (Acierto/error): Esta columna muestra un 1 si la clasificación ha sido correcta y mostrará un 0 si ha sido errónea.

  Función: SI (Etiqueta real = Predicción; 1; 0)

- Columna 5 (VP): Esta columna muestra los verdaderos positivos, es decir, si se predice de cierto audio que es un paciente (“0”) y realmente es un paciente, mostrará un 1, en cualquier otro caso mostrará un 0.

  Función: SI ((Etiqueta real = 0) && (Acierto/error = 1); 1; 0)

- Columna 6 (FP): Esta columna muestra los falsos positivos, es decir, si se predice de cierto audio que es un paciente (“0”) y realmente es un control (“1”), mostrará un 1, en cualquier otro caso mostrará un 0.

  Función: SI ((Etiqueta real = 1) && (Predicción = 0); 1; 0)

- Columna 7 (VN): Esta columna muestra los verdaderos negativos, es decir, si se predice de cierto audio que es un control (“1”) y realmente es un control (“1”), mostrará un 1, en cualquier otro caso mostrará un 0.

  Función: SI ((Etiqueta real = 1) && (Predicción = 1); 1; 0)

- Columna 8 (FN): Esta columna muestra los falsos negativos, es decir, si se predice de cierto audio que es un control (“1”) y realmente es un paciente (“0”), mostrará un 1, en cualquier otro caso mostrará un 0.

  Función: SI ((Etiqueta real = 0) && (Predicción = 1); 1; 0)

Para la mejor comprensión de cada uno de estos casos de la columna 4 hasta la columna 8, se va a mostrar una figura que englobe los 4 casos posibles: VP, FP, VN, FN.

AUDIOS	Etiqueta real	Predicción	Acierto/error	VP	FP	VN	FN
4ri7ruz.wav	0	0	1	1	0	0	0
4RO1A.wav	1	0	0	0	1	0	0
4SK3KRK.wav	1	1	1	0	0	1	0
4sk3la.wav	0	1	0	0	0	0	1

Figura 5

ACIERTO/ERROR

Cómo se ha puede observar en la **Figura 5**, en el audio 4ri7ruz.wav hay un acierto en la predicción ("1"), pues el sistema predice que ese audio pertenece a un paciente y realmente el audio pertenece a un paciente. Sin embargo, en el audio 4RO1A.wav hay un error en la predicción, pues el sistema predice que el audio pertenece a un paciente cuando realmente pertenece a un control.

VP

Cómo se ha puede observar en la **Figura 5**, en el audio 4ri7ruz.wav hay un verdadero positivo, pues el sistema predice que ese audio pertenece a un paciente y realmente el audio pertenece a un paciente.

FP

Cómo se ha puede observar en la **Figura 5**, en el audio 4RO1A.wav hay un falso positivo, pues el sistema predice que ese audio pertenece a un paciente y realmente el audio pertenece a un control.

VN

Cómo se ha puede observar en la **Figura 5**, en el audio 4SK3KRK.wav hay un verdadero negativo, pues el sistema predice que ese audio pertenece a un control y realmente el audio pertenece a un control.

FN

Cómo se ha puede observar en la **Figura 5**, en el audio 4sk3la.wav hay un falso negativo, pues el sistema predice que ese audio pertenece a un control y realmente el audio pertenece a un paciente.

Dado que la tarea planteada en este trabajo es binaria, hemos optado por utilizar como medidas de evaluación, la precisión, el *Recall* y el F-score, cuyas fórmulas se incluyen a continuación:

- $Precisión = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos}$ **Ecuación 4**: Precisión
- $Recall = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ positivos}$ **Ecuación 5**: Recall
- $F - score = 2 * \frac{Precisión * Recall}{Precisión + Recall}$ **Ecuación 6**: F-score

Por lo tanto, la precisión se puede definir como el porcentaje de predicciones positivas que fueron correctas, el Recall, como el porcentaje de casos positivos que fueron capturados y el F-score, como un valor único ponderado de la precisión y el Recall.

En cuanto a la visualización de los resultados, estos se han almacenado en tablas de Excel de la siguiente manera:

COLUMA	DESCRIPCIÓN			RESULTADOS			
COLUMNA B	Control o Paciente	"1" Control	"0" Paciente	Total Contoles	527	Total Pacientes	836
COLUMNA C	Predicción	"1" Control	"0" Paciente	Total Controles Predichos	628	Total pacientes Predichos	735
COLUMNA D	Acierto o error en Predicción	"1" Acierto	"0" Error	Total Aciertos	1144	Total Errores	219
COLUMNA E	Verdadero Positivo	1 es un VP	0 No es un VP	Total VP	676	Positivos no detectados	160
COLUMNA F	Falso Positivo	1 es un FP	0 no es un FP	Total FP	59		
COLUMNA G	Verdadero Negativo	1 es un VN	0 no es un VN	Total VN	468	Negativos no detectados	59
COLUMNA H	Falso Negativo	1 es un FN	0 no es un FN	Total FN	160		
PRECISION	P=VP/(VP+FP)			0,919727891			
RECALL	R=VP/(VP+FN)			0,80861244			
FSCORE	FS=2*P*R/(P+R)			0,860598345			

Total Audios	1363

Figura 6

A partir de tablas como la de la **Figura 6** (hay una tabla así para cada prueba), se han graficado los resultados.

5. Pruebas y resultados

5.1. Protocolo experimental

En este apartado, se detallarán las pruebas realizadas. Al ser un gran número de pruebas, estas, se detallarán en una tabla, la cual constará de 3 columnas. La primera especificará qué grupo se ha usado para entrenar el modelo, la segunda, contendrá al grupo test el cuál ha sido sometido a la prueba, y la tercera, cómo se ha entrenado al grupo de la primera columna. A continuación, la incluye la tabla antes mencionada:

<table>
<tr><td rowspan="4">Independiente del locutor train.

Parámetros MFCC</td><td rowspan="3">Independiente del locutor test.

Parámetros MFCC</td><td>Polinómico grado 3</td></tr>
<tr><td>Modelo Lineal</td></tr>
<tr><td>Polinómico grado 2</td></tr>
<tr><td rowspan="3">Independiente del locutor test. Parámetros MFCC. Ruido: Living (-5SNR hasta 20 SNR de 5 en 5)</td><td rowspan="5">Modelo gaussiano</td></tr>
<tr><td rowspan="3">Independiente del locutor test. Parámetros MFCC. Ruido: Meeting (-5 SNR hasta 20 SNR de 5 en 5)</td></tr>
<tr><td rowspan="3">Independiente del locutor test. Parámetros MFCC. Ruido: Office (-5 SNR hasta 20 SNR de 5 en 5)</td></tr>
<tr><td rowspan="8">Independiente del locutor train.

Parámetros MFCC quitando la energía</td><td>Independiente del locutor test. Parámetros MFCC excluyendo la energía</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC excluyendo la energía y con vadsohn</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC excluyendo la energía</td><td rowspan="2">Modelo lineal</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC excluyendo la energía y con vadsohn</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC excluyendo la energía</td><td rowspan="2">Polinómico grado 2</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC excluyendo la energía y con vadsohn</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC excluyendo la energía</td><td rowspan="2">Polinómico grado 3</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC excluyendo la energía y con vadsohn</td></tr>
</table>

<table>
<tr><td rowspan="4">Independiente del locutor train. Parámetros MFCC y delta-MFCC</td><td>Independiente del locutor test.
Parámetros MFCC y delta-MFCC</td><td rowspan="4">Modelo Gaussiano</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC y delta-MFCC. Ruido: Living (-5SNR hasta 20SNR de 5 en 5)</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC + delta-MFCC. Ruido Meeting (de -5SNR hasta 20SNR de 5 en 5)</td></tr>
<tr><td>Independiente del locutor test. Parámetros MFCC y delta-MFCC. Ruido: Office (-5SNR hasta 20SNR de 5 en 5)</td></tr>
<tr><td rowspan="5">Dependiente del locutor train.
Parámetros MFCC</td><td rowspan="2">Dependiente del locutor test.
Parámetros MFCC</td><td>Modelo Lineal</td></tr>
<tr><td>Polinómico grado 2</td></tr>
<tr><td rowspan="3">Dependiente del locutor test. Parámetros MFCC. Ruido: Living (-5 SNR a 20 SNR de 5 en 5)</td><td rowspan="3">Polinómico grado 3</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC. Ruido: Meeting (-5 SNR a 20 SNR de 5 en 5)</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC. Ruido: Office (-5 SNR a 20 SNR de 5 en 5)</td></tr>
<tr><td rowspan="8">Dependiente del locutor train.
Parámetros MFCC quitando la energía</td><td>Dependiente del locutor test. Parámetros MFCC excluyendo la energía</td><td rowspan="2">Modelo Gaussiano</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC excluyendo la energía y con vadsohn</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC excluyendo la energía</td><td rowspan="2">Modelo lineal</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC excluyendo la energía y con vadsohn</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC excluyendo la energía</td><td rowspan="2">Polinómico grado 2</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC excluyendo la energía y con vadsohn</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC excluyendo la energía</td><td rowspan="2">Polinómico grado 3</td></tr>
<tr><td>Dependiente del locutor test. Parámetros MFCC excluyendo la energía y con vadsohn</td></tr>
</table>

Dependiente del locutor train. Parámetros MFCC y delta-MFCC	Dependiente del locutor test. Parámetros MFCC y delta-MFCC	Polinómico grado 3
	Dependiente del locutor test. Parámetros MFCC y delta-MFCC. Ruido: Living (-5SNR a 20SNR de 5 en 5)	
	Dep. del locutor test. Parámetros MFCC y delta-MFCC. Ruido: Meeting (-5SNR a 20SNR de 5 en 5)	
	Dependiente del locutor test. Parámetros MFCC y delta-MFCC. Ruido: Office (-5SNR a 20SNR de 5 en 5)	

Tabla 10: Pruebas

Para todas estas pruebas de la **Tabla 10**, se calcula la Precisión, el Recall y el F-score para poder comparar los distintos modelos entrenados y cómo se comportan estos modelos para distintos datos sometidos a clasificación.

También se han graficado varios datos de distintas pruebas para visualizar mejor los resultados.

5.2. Resultados

En el presente aparatado, se mostrarán los resultados de todas las pruebas realizadas en forma de tablas.

A continuación de todas estas tablas se mostrarán gráficas que muestran el F-score de varias pruebas a modo de columnas.

Cabe destacar las abreviaturas que se usarán para nombrar las pruebas y tablas de resultados:

- SinE: Exclusión de la energía para la prueba.
- VAD: A los audios se les han suprimido los silencios mediante la función vadsohn.
- xSNR_ruido: Se ha introducido el ruido especificado con una SNR de x.
- DMFCC: Parámetros delta-MFCC

Con esto, los resultados son los siguientes:

RESULTADOS DEPENDIENTES DEL LOCUTOR Y FUNCIÓN GAUSSIANA

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC	98,52	98,89	0,9870
SinE		84,24	89,25	0,8667
SinE con VAD		81,30	87,64	0,8435

Tabla 11: Resultados dependientes del locutor y función gaussiana

Cómo se puede observar en la **Tabla 11**, la precisión y el recall y en consecuencia de estos dos, el F-score, disminuyen notablemente, un 14% y un 10% respectivamente, cuando no se tiene en cuenta el parámetro de la log-energía. Además, en el caso de no utilizar la log-energía, eliminar los silencios hace que el F-score disminuya un 2% respecto al caso en que se mantienen los silencios.

RESULTADOS DEPENDIENTES DEL LOCUTOR Y FUNCIÓN POLINÓMICA DE GRADO 2

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC	98,04	98,76	0,9840
SinE		80,84	90,73	0,8550
SinE con VAD		78,34	91,22	0,8429

Tabla 12: Resultados dependientes del locutor y función polinómica de grado 2

Cómo se puede observar en la **Tabla 12,** cuando se entrena el modelo con una función polinómica de segundo grado, la tendencia de los resultados al extraer la energía es similar a la explicada para la **Tabla 11,** la cual entrena el modelo con función gaussiana.

RESULTADOS DEPENDIENTES DEL LOCUTOR Y FUNCIÓN POLINÓMICA DE GRADO 3 (MFCC)

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC	98,04	98,76	0,9840
SinE		80,84	90,73	0,8550
SinE con VAD		78,34	91,22	0,8429
-5SNR_Living		100	16,32	0,2806
0SNR_Living		99,00	49,07	0,6562
5SNR_Living		100	53,65	0,6983
10SNR_Living		99,83	74,04	0,8502
15SNR_Living		99,59	89,74	0,9441
20SNR_Living		99,61	95,80	0,9767
-5SNR_Meeting		100	27,69	0,4337
0SNR_Meeting		99,74	47,59	0,6444
5SNR_ Meeting		99,82	69,22	0,8175
10SNR_Meeting		99,58	87,52	0,9316
15SNR_Meeting		99,48	94,19	0,9676

20SNR_Meeting	MFCC	99,38	98,27	0,9882
-5SNR_Office		100	34,36	0,5115
0SNR_Office		100	53,03	0,6931
5SNR_ Office		99,82	70,21	0,8244
10SNR_Office		99,86	88,63	0,9391
15SNR_Office		99,74	95,67	0,9767
20SNR_Office		99,50	97,78	0,9863

Tabla 13: Resultados dependientes del locutor y función polinómica de grado 3 (MFCC)

En la **Tabla 13,** se puede observar que se sigue cumpliendo para el caso de entrenar el modelo con una función polinómica de grado 3 que, no utilizar la energía y eliminar los silencios de los audios empeora los resultados de precisión recall y F-score; como ya se ha visto también para la **Tabla 11** y la **Tabla 12.**

En esta **Tabla 13**, también se puede observar cómo afecta a la precisión, el recall y el F-score, el añadir a los audios de test distintos ruidos y distintas relaciones de SNR para estos ruidos.

Pues bien, en cuanto a la influencia en el sistema de qué SNR tengan los audios de test, se puede afirmar que, para cualquier tipo de ruido, con SNR = - 5 dB se obtienen porcentajes muy altos de precisión (100%), pero se obtienen valores muy bajos de recall (entre el 16% y el 35%). Es decir, el sistema identifica a todos los pacientes como pacientes, pero a muchos de los controles también les identifica como pacientes.

Según se va aumentando la SNR, la precisión disminuye muy poco el porcentaje, pero el recall va aumentando hasta superar el 95% para cualquier tipo de ruido para una SNR de 20dB. Estos valores de precisión y recall para 20dB hacen que el valor de F-score ya sea más cercano a 1.

En cuanto los tipos de ruido, se puede decir que Living implica que el sistema obtenga peores resultados para SNR bajas (-5dB y 5dB), pero según la SNR va aumentando, los 3 ruidos tienen una relevancia similar para el sistema.

RESULTADOS DEPENDIENTES DEL LOCUTOR Y FUNCIÓN POLINÓMICA DE GRADO 3 (DMFCC)

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC + DMFCC	99,62	99,63	0,9963
-5SNR_Living		98,69	37,33	0,5417
0SNR_Living		97,01	68,11	0,8003
5SNR_Living		93,10	86,77	0,8983
10SNR_Living		89,46	94,44	0,9188
15SNR_Living		86,09	97,90	0,9161

20SNR_Living	MFCC + DMFCC	83,87	99,01	0,9082
-5SNR_Meeting		99,30	52,78	0,6893
0SNR_Meeting		95,52	78,99	0,8647
5SNR_ Meeting		92,23	92,46	0,9235
10SNR_Meeting		88,12	97,16	0,9242
15SNR_Meeting		85,41	99,13	0,9176
20SNR_Meeting		82,97	99,38	0,9044
-5SNR_Office		95,07	78,62	0,8606
0SNR_Office		90,14	92,71	0,9141
5SNR_ Office		86,40	97,40	0,9157
10SNR_Office		83,44	98,39	0,9030
15SNR_Office		81,86	99,26	0,8972
20SNR_Office		81,15	99,51	0,8939

Tabla 14: Resultados dependientes del locutor y función polinómica de grado 3 (DMFCC)

En esta **Tabla 14**, se puede observar que añadir ruido a los audios tiene las siguientes consecuencias en el sistema: La precisión para SNR=-5 dB obtiene porcentajes altos, superiores al 95%, mientras que el recall obtiene resultados muy dispares dependiendo del tipo de ruido. Se obtiene el mejor recall para el ruido de office y el peor recall para el ruido de living.

Según aumenta la SNR, para todos los ruidos, va disminuyendo la precisión hasta situarse en torno al 82% para SNR=20 dB. Y el recall, va aumentando hasta llegar al 99% para los 3 ruidos con una SNR de 20dB.

Con lo cual, el F-score obtiene su mejor resultado para una SNR de 10 dB para los casos de ruidos de living o meeting y para una SNR de 5 dB para el caso de ruido de office. Estos resultados indican que cómo se puede comprobar comparando la **Tabla 13** y la **Tabla 14**, las cuales solo se diferencian en que se han añadido al sistema los parámetros delta-MFCC, que para SNRs entre -5dB y 5dB para cualquier tipo de ruido, se obtienen mejores resultados introduciendo al sistema los parámetros delta-MFCC al sistema, y que sin embargo para SNRs entre 10dB y 20dB se obtienen mejores resultados sin introducir los delta-MFCC.

RESULTADOS DEPENDIENTES DEL LOCUTOR Y FUNCIÓN LINEAL

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC	98,02	98,15	0,9809
SinE		73,18	89,37	0,8047
SinE con VAD		72,44	89,37	0,8002

Tabla 15: Resultados dependientes del locutor y función lineal

En esta **Tabla 15**, se pueden observar sucesos similares a los comentados en la **Tabla 11** y la **Tabla 12.**

RESULTADOS INDEPENDIENTES DEL LOCUTOR Y FUNCIÓN GAUSSIANA (MFCC)

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC	96,86	99,52	0,9817
SinE		82,14	83,61	0,8287
SinE con VAD		78,16	81,34	0,7972
-5SNR_Living		100	17,34	0,2956
0SNR_Living		100	21,53	0,3543
5SNR_Living		100	50,96	0,6751
10SNR_Living		99,84	75,00	0,8566
15SNR_Living		99,23	92,22	0,9560
20SNR_Living		98,55	97,49	0,9802
-5SNR_Meeting		100	27,51	0,4315
0SNR_Meeting		99,73	44,14	0,6119
5SNR_ Meeting		99,83	72,13	0,8375
10SNR_Meeting		99,34	90,43	0,9468
15SNR_Meeting		98,44	97,97	0,9820
20SNR_Meeting		97,99	98,92	0,9845
-5SNR_Office		100	33,37	0,5004
0SNR_Office		99,56	53,71	0,6977
5SNR_ Office		99,84	73,80	0,8487
10SNR_Office		99,09	91,39	0,9508
15SNR_Office		98,67	97,97	0,9832
20SNR_Office		97,76	99,28	0,9852

Tabla 16: Resultados independientes del locutor y función gaussiana (MFCC)

En esta **Tabla 16,** se observa una evolución de los parámetros precisión, recall y F-score muy similar a lo comentado en la **Tabla 13**. Estas dos tablas se diferencian en la dependencia o independencia del locutor en las pruebas y en la función de Kernel usada para entrenar al sistema.

RESULTADOS INDEPENDIENTES DEL LOCUTOR Y FUNCIÓN LINEAL

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC	96,27	98,68	0,9746
SinE		73,87	81,82	0,7764
SinE con VAD		74,48	80,98	0,7759

Tabla 17: Resultados independientes del locutor y función lineal

En esta **Tabla 17**, se puede observar un impacto negativo de no utilizar la energía del estudio de 0'2 puntos en el F-score y que la eliminación de los silencios no tiene una importancia significativa en los resultados.

RESULTADOS INDEPENDIENTES DEL LOCUTOR Y FUNCIÓN GAUSSIANA (DMFCC)

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC + DMFCC	96,86	99,64	0,9823
-5SNR_Living		99,15	41,63	0,5864
0SNR_Living		95,23	71,65	0,8177
5SNR_Living		90,01	88,40	0,8920
10SNR_Living		86,18	94,02	0,8993
15SNR_Living		82,59	97,61	0,8947
20SNR_Living		80,18	98,21	0,8828
-5SNR_Meeting		98,37	57,66	0,7270
0SNR_Meeting		93,05	80,02	0,8605
5SNR_ Meeting		88,00	91,27	0,8961
10SNR_Meeting		84,64	96,89	0,9035
15SNR_Meeting		81,76	97,61	0,8899
20SNR_Meeting		79,57	98,33	0,8796
-5SNR_Office		92,40	75,60	0,8316
0SNR_Office		86,71	92,11	0,8933
5SNR_ Office		83,16	96,29	0,8925
10SNR_Office		80,39	97,61	0,8817
15SNR_Office		78,47	98,09	0,8719
20SNR_Office		77,05	98,80	0,8658

Tabla 18: Resultados independientes del locutor y función gaussiana (DMFCC)

En la **Tabla 18**, se puede observar como la precisión empeora según aumenta la SNR para cualquier tipo de ruido (exceptuando la condición de audios limpios) y el recall mejora según la SNR aumenta.

También se puede observar, comparando esta tabla con la **Tabla 16,** como la inclusión de los parámetros delta-MFCC mejora los valores de F-score para valores de SNR entre -5dB y 5dB respecto a no incluir estos delta-MFCC.

RESULTADOS INDEPENDIENTES DEL LOCUTOR Y FUNCIÓN POLINÓMICA GRADO 2

Condiciones	Parámetros	Precisión	Recall (%)	F-score
Limpias	MFCC	96,74	99,52	0,9811
SinE		78,99	85,88	0,8229
SinE con VAD		76,96	83,49	0,8009

Tabla 19: Resultados independientes del locutor y función polinómica grado 2

En esta **Tabla 19**, se puede observar un comportamiento de los parámetros precisión recall y F-score según las condiciones de audios limpios, extracción de la energía y extracción de los silencios, similar al observado en la **Tabla 17** Aunque en este caso, la diferencia de las funciones de Kernel usadas en estos modelos, hacen ver una pequeña mejoría en los resultados de la función polinómica de grado 2 respecto a la lineal.

RESULTADOS INDEPENDIENTES DEL LOCUTOR Y FUNCIÓN POLINÓMICA GRADO 3

Condiciones	Parámetros	Precisión (%)	Recall (%)	F-score
Limpias	MFCC	96,30	99,52	0,9788
SinE		80,83	86,24	0,8345
SinE con VAD		80,07	85,05	0,8248

Tabla 20: Resultados independientes del locutor y función polinómica grado 3

En esta **Tabla 20**, se puede observar un comportamiento de los parámetros precisión recall y F-score según las condiciones de audios limpios, extracción de la energía y extracción de los silencios, similar al observado en la **Tabla 17** y en la **Tabla 19**. Aunque en este caso, la diferencia de las funciones de Kernel usadas en estos modelos, hacen ver una pequeña mejoría en los resultados de la función polinómica de grado 2 respecto a la polinómica de grado 3 y una mejoría de la de grado 3 en comparación con la lineal.

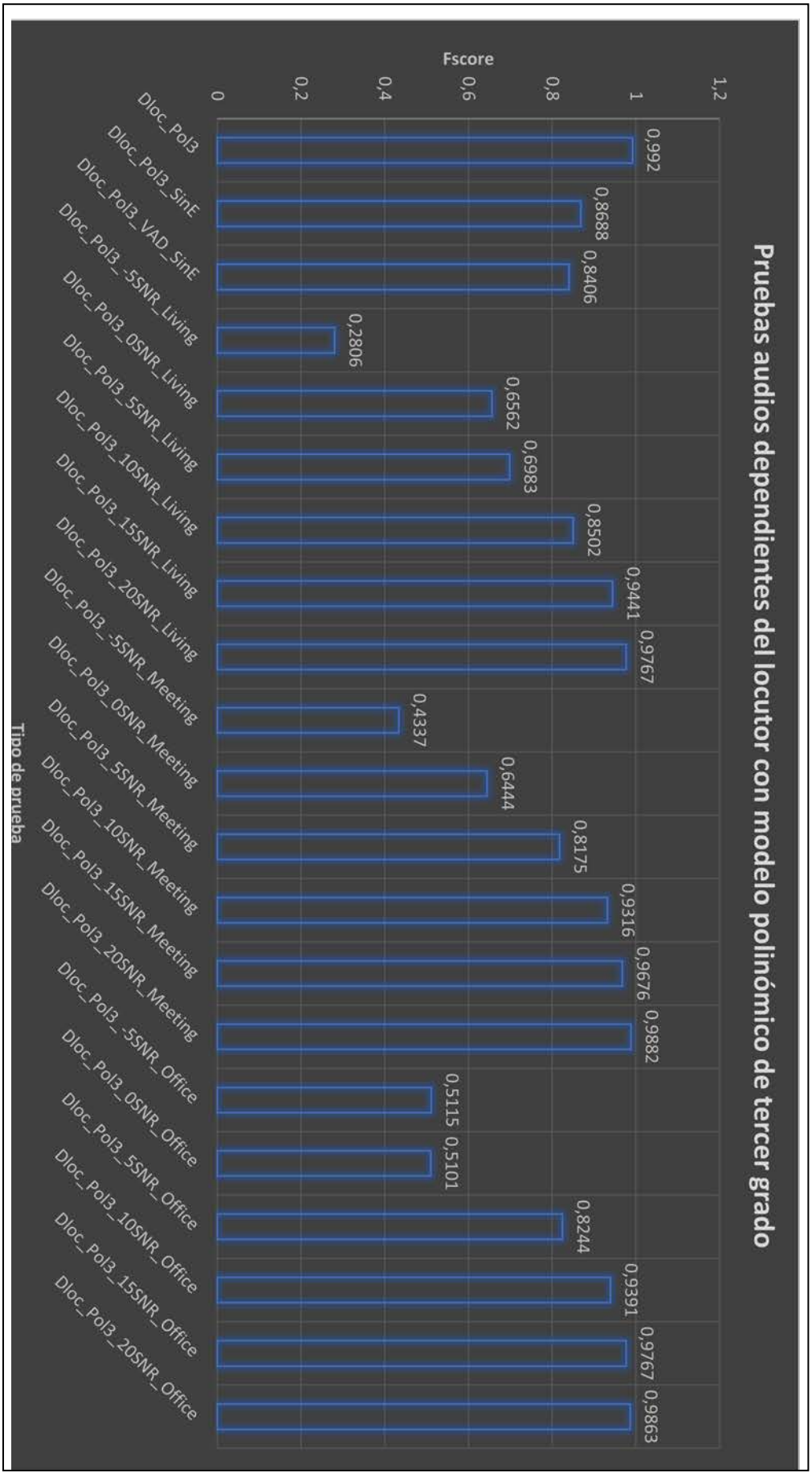

Figura 7: Gráfico pruebas dependientes del locutor con modelo polinómico grado 3

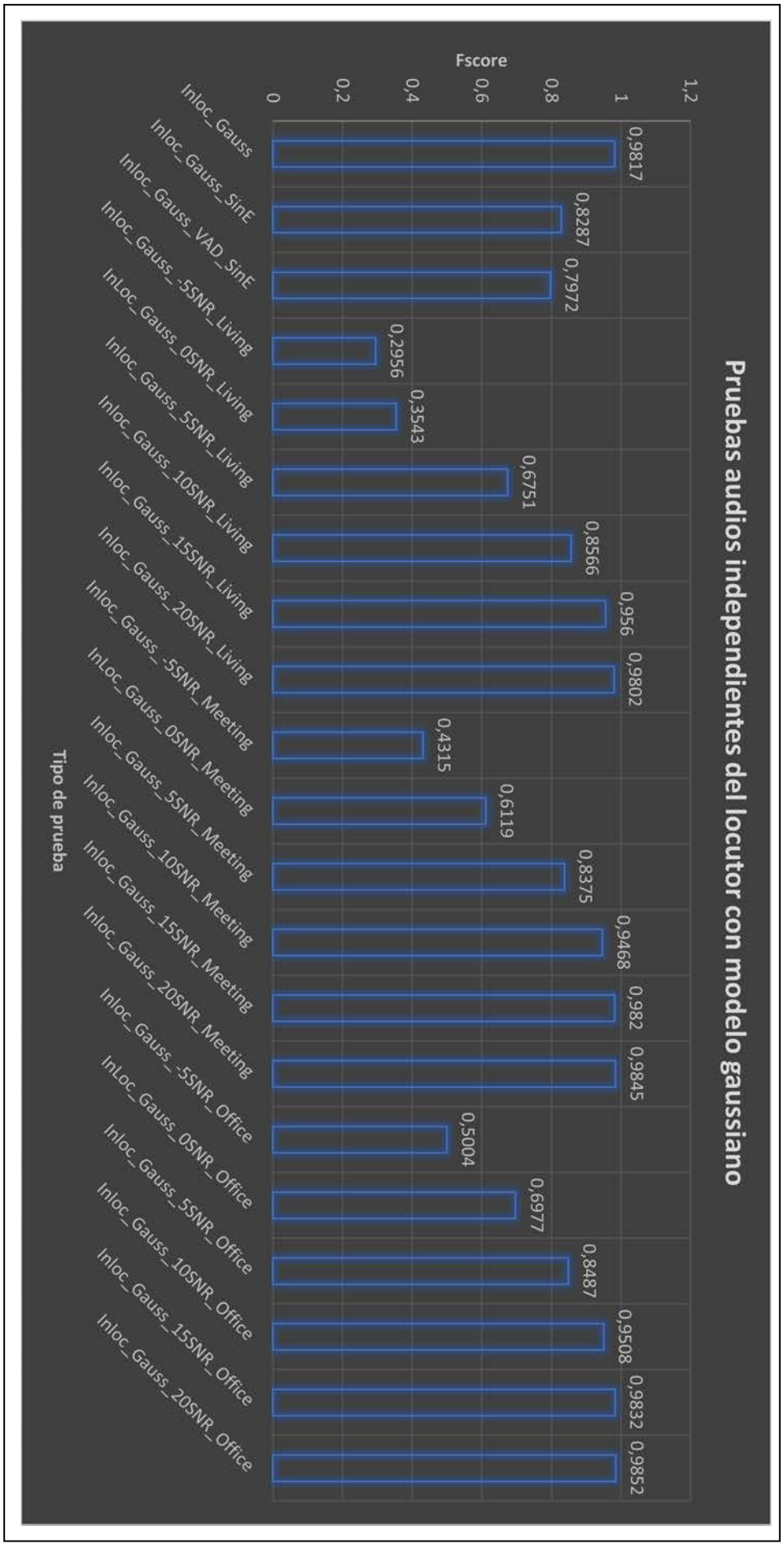

Figura 8: Gráfico pruebas independientes del locutor con modelo gaussiano grado 3

En cuanto a la **Figura 7** y la **Figura 8** cabe comentar que se muestran para observar de manera más gráfica todo lo comentado debajo de cada tabla de este mismo apartado, en concreto, de la **Tabla 13** y la **Tabla 16**.

5.2.1. Discusión de los resultados

En este capítulo se realizará un análisis de los resultados mostrados en el apartado anterior.

Para realizar este análisis, deben quedar contestadas las siguientes preguntas:

- ¿Qué influencia tiene extraer la energía de los parámetros que forman parte del sistema de entrenamiento y clasificación?
- ¿Mejora el sistema la eliminación de los silencios de los audios?
- ¿Qué impacto tiene en los resultados la dependencia o independencia del locutor?
- ¿Mejoran los parámetros delta-MFCC a los resultados obtenidos estudiando únicamente los MFCC?
- ¿Cómo afecta el ruido al sistema?
- ¿Podemos concluir que alguna de las funciones de Kernel se impone a las demás en cuanto a resultados obtenidos?

¿Qué influencia tiene extraer la energía de los parámetros que forman parte del sistema de entrenamiento y clasificación?

Cómo se puede observar en las tablas: **Tabla 11**, **Tabla 12**, **Tabla 13**, **Tabla 15**, **Tabla 20**, **Tabla 16**, **Tabla 17**, **Tabla 19.** Se puede concluir, que la energía es un parámetro de vital importancia para el sistema, pues sin incluir la energía en el sistema disminuye considerablemente el Recall, la Precisión y el F-score para cualquiera de las cuatro funciones de Kernel usadas para entrenar los modelos.

Por tanto, para mejorar el funcionamiento del sistema se debe mantener la energía entre los parámetros acústicos utilizados.

¿Mejora el sistema la eliminación de los silencios de los audios?

Como se puede observar en las tablas: **Tabla 11**, **Tabla 12**, **Tabla 13**, **Tabla 15**, **Tabla 20**, **Tabla 16**, **Tabla 17**, **Tabla 19.** Se puede concluir que para cualquiera de las distintas formas en las que se ha entrenado el modelo, al eliminar los silencios, disminuye ligeramente el F-score del sistema en la clasificación. Esto puede ser debido a que los silencios están relacionados con el ritmo de elocución o las difluencias, por lo que pueden ser útiles para la detección del síndrome del trastorno específico del lenguaje, por lo que no deberían eliminarse.

¿Qué impacto tiene en los resultados la dependencia o independencia del locutor?

Como se puede observar en las tablas: : **Tabla 11**, **Tabla 12**, **Tabla 13**, **Tabla 14**,**Tabla 15**, **Tabla 20**, **Tabla 16**, **Tabla 17**, **Tabla 18**,**Tabla 19.** Se puede concluir que las pruebas con dependencia del locutor mejoran ligeramente la precisión respecto a las pruebas con independencia de locutor.

De cara a unas pruebas reales, existiría independencia de locutor, pues el sistema no se entrenaría con pacientes a los que se les quiere detectar o descartar que tengan SLI. Pese a esto, y para audios limpios, el F-score del sistema es superior al 0'95 por audio, con lo que la probabilidad de acierto en la clasificación es alta.

¿Mejoran los parámetros delta-MFCC a los resultados obtenidos estudiando únicamente los MFCC?

Como se puede observar en las tablas: **Tabla 13**, **Tabla 14**, **Tabla 16** y **Tabla 18.** Se puede concluir que, en condiciones limpias, incluir los parámetros delta-MFCC al sistema, mejora la eficiencia en la clasificación.

¿Cómo afecta el ruido al sistema?

Como se puede observar en las tablas: **Tabla 13**, **Tabla 14**, **Tabla 16** y **Tabla 18.** Se puede concluir que, en condiciones ruidosas, se observa que para una SNR menor de 10dB, o 5dB, según la función de Kernel usada para el entrenamiento, los parámetros delta-MFCC mejoran el sistema respecto a sólo usar los MFCC. Pero, desde estas SNR hasta una SNR de 20 dB, el sistema funciona mejor si solo se usan los parámetros MFCC.

Además, como se puede ver en la **Figura 8**, el parámetro F-score supera el 0'95 a partir de 15 SNR para todo tipo de ruido en las pruebas para independencia de locutor con un modelo entrenado basándose en la función de Kernel gaussiana. Y para las pruebas con dependencia del locutor, como se puede observar en la **Figura 7** hay datos similares.

¿Podemos concluir que alguna de las funciones de Kernel se impone a las demás en cuanto a resultados obtenidos?

Para las pruebas con dependencia del locutor, la función con mejores resultados es la polinómica de grado 3, mientras que para las pruebas con independencia del locutor la función de Kernel que ofrece mejores resultados es el Kernel gaussiano. Ambas afirmaciones se pueden contrastar en las tablas: **Tabla 11**, **Tabla 12**, **Tabla 13**, **Tabla 15**, **Tabla 20**, **Tabla 16**, **Tabla 17**, **Tabla 19**.

6. Líneas futuras.

En este apartado, se proponen ideas para mejorar y ampliar en un futuro el presente trabajo.

Una de las líneas en las que cabe pensar es en el estudio de la capacidad del sistema creado para detectar casos de SLI en otros idiomas que no sea en el que se han hecho las pruebas. Esto es de vital importancia para un futuro, pues es conocido que las características acústicas de los idiomas son muy distintas unas respecto a otras. Esto podría hacer que para distintos lenguajes sea necesario entrenar al modelo de diversas maneras o con audios distintos, incluso con parámetros distintos.

También podrían realizarse pruebas en una variedad mayor de condiciones acústicas y aplicar alguna técnica de mejora de la señal de voz o reducción del ruido.

Otra de las líneas futuras es probar el funcionamiento del sistema desarrollado para la detección de otros síndromes del habla o para otras enfermedades que, sin ser del habla, tengan una repercusión directa en el sistema de producción vocal de los pacientes.

También es interesante para un futuro, profundizar en las peculiaridades de los errores del sistema e intentar buscar un patrón común, así como realizar distintos sistemas para los distintos rangos de edades, el sexo del paciente, pues es muy posible que creando subsistemas para subgrupos con las mismas características la eficiencia de los sistemas aumente.

7. Conclusiones.

En este apartado de la memoria se pretende llegar a conclusiones sobre el trabajo realizado, analizando principalmente cómo han funcionado los parámetros MFCC y delta-MFCC de cara a la consecución de los objetivos del problema, a cómo ha funcionado el sistema de entrenamiento proporcionado por Matlab usado y a cómo afectan las situaciones ruidosas a la clasificación final de los audios.

No se debe perder de vista el objetivo inicial de este trabajo de fin de grado para obtener una conclusión acorde al mismo. Pues bien, una medida que cuantifica la capacidad de clasificación entre niños con SLI y niños sin el SLI es el F-score. Y, en base a los resultados de F-score obtenidos, se puede concluir que el objetivo inicial, al menos para audios limpios y para el caso de esta base de datos en concreto, se ha cumplido.

Con lo cual, con este trabajo también se pone un granito más de arena a la teoría de que los parámetros MFCC son muy útiles y obtienen buenos resultados para problemas con el análisis de la voz o detección de trastornos relacionados con el neurodesarrollo.

También se da como bueno en base a los resultados obtenidos, aunque siempre mejorable, el sistema de entrenamiento usado de soporte de máquina de vectores, SVM.

En cuanto al comportamiento del sistema teniendo como entrada audios en condiciones de ruido, se supera un F-score del 0’95 para SNRs mayores a 15dbB. Esta SNR no es excesivamente alta ni difícil de conseguir, aunque un tratado previo de los audios para aumentar esta SNR siempre sería beneficioso para el problema, pues para los tipos de ruido probados, se confirma la relación directa de que a mayor SNR mayor es el valor de F-score y por lo tanto mejores resultados en la clasificación se obtiene.

La conclusión es que, este trabajo es un buen gran primer paso para explorar más tipos de condiciones en los audios, si funciona bien o hay que meter variaciones para otros idiomas etc. En definitiva, sirve como base para explorar las líneas futuras explicadas en el apartado anterior. Pues el F-score obtenido para la prueba con independencia de locutor entrenando el sistema mediante la función de Kernel Gaussiana es bastante bueno, muy cercano a 1, más en concreto: 0’9817.

Es decir, los resultados son optimistas. También se ha de tener en cuenta, que para diagnóstico de carácter clínico se necesita tener unas garantías de acierto mucho mayores que para otros campos. Esto se debe a que un mal diagnóstico repercute directamente en la vida de la persona. Con lo que para que el sistema pueda ser usado por los profesionales de la medicina sobre los cuales recae la responsabilidad de realizar el diagnóstico y tienen el conocimiento para realizarlo, sería bueno mejorar un poco más los resultados, pese a que los mismos son bastante altos.

8. Presupuesto

En este apartado se detallarán los costes empleados para el desarrollo de todas las fases de este trabajo de fin de grado.

En cuanto a material físico, se ha usado un portátil con coste de 700 euros y un disco duro con un coste de 50 euros. Lo que suma un total de material físico de 750 euros

En cuanto a Software ha sido todo gratuito, pues se ha usado Matlab y el paquete Office, todo proporcionado por la universidad Carlos III.

En cuanto a recursos humanos, contando con que la hora de un ingeniero senior como es la tutora del proyecto, Ascensión, se estima en torno a 25 euros la hora y la hora de un ingeniero Junior como es el caso del alumno se estima en 15 euros la hora.

Contabilizando con que la tutora ha empleado 50 horas y el alumno ha empleado 400 horas, se estima un coste en recursos humanos de 7250 euros.

Sumando todos los recursos utilizados, se completa un presupuesto de 8000 euros.

9. Anexos

```
lectura;

M_media = zeros(dim(1),13);
format = '%f';

for i = 1:dim(1)

    b = Mstrings(i);
    b = replace(b, "WAV", "txt");
    b = replace(b, "wav", "txt");

    file = fopen(b{1}, 'r');

    M_datos = textscan(file, format);
    M_datos = M_datos{1};
    d = size(M_datos);
    M_datos = reshape(M_datos, [13, d(1)/13]);
    M_datos = M_datos';

    for j = 1:13

        M_media(i,j) = mean(M_datos(:,j));
```

Figura 9: lectura.m

```
lectura;

parametros = 12;
filtros = 40;

for i=1:dim(1)
    b = Mstrings{i};
    b = replace(b, "WAV", "txt");
    b = replace(b, "wav", "txt");
    fid = fopen(b, 'w');
    [aud, str, lbl, Fs] = extraerAudioPos(i);
    c = melcepst(aud,Fs,'E',parametros,filtros,0.02*Fs,0.01*Fs);
    matriz = size(c);
    filas = matriz(1);
    columnas = matriz(2);
    for n = 1:filas
        for t = 1:columnas
            fprintf(fid,'%f\t',c(n,t));
        end
        fprintf(fid,'\n');
    end
    fclose(fid);
end
```

Figura 10: escritura_mfcc.m

```
lectura;

M_media = zeros(dim(1),13);
format = '%f';
for i = 1:dim(1)

    b = Mstrings(i);
    b = replace(b, "WAV", "txt");
    b = replace(b, "wav", "txt");
    file = fopen(b{1}, 'r');

    M_datos = textscan(file, format);
    M_datos = M_datos{1};
    d = size(M_datos);
    M_datos = reshape(M_datos, [13, d(1)/13]);
    M_datos = M_datos';

    for j = 1:13

        M_media(i,j) = mean(M_datos(:,j));

    end

    fclose(file);

end
```

Figura 11: matriz.m

10. Referencias

- ***[1]*** Speech databases of typical children with SLI (consultado por última vez: 15/05/2019): https://lindat.mff.cuni.cz/repository/xmlui/handle/11372/LRT-1597
- ***[2]*** Agencia Estatal: Boletín Oficial del Estado (consultado por última vez: 13/06/2019): https://www.boe.es/diario_boe/
- ***[3]*** National Institute on Deafness and other Communication Disorders (consultado por última vez: 01/05/2019): https://www.nidcd.nih.gov/
- ***[4]*** Neuronas en Crecimiento (consultado por última vez: 01/05/2019): https://neuropediatra.org/
- ***[5]*** Classification and Detection of Specific Language Impairments in Children Base don their Speech Skills. By Pavel Grill and Jana Tucková (consultado por última vez: 25/05/2019): https://www.intechopen.com/books/learning-disabilities-an-international-perspective/classification-and-detection-of-specific-language-impairments-in-children-based-on-their-speech-skil
- ***[6]*** Sociedad española de acústica (consultado por última vez: 01/04/2019): http://www.sea-acustica.es/fileadmin/Evora12/227.pdf
- ***[7]*** Máquina de entrenamiento de vectores de soporte (SVM) (consultado por última vez: 10/06/2019): https://es.mathworks.com/
- ***[8]*** voicebox (consultado por última vez: 05/05/2019): http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/mdoc/index.html
- ***[9]*** DEMAND (consultado por última vez: 23/05/2019): Diverse Enviroments Multichannel Acoustic Noise Database: https://zenodo.org/record/1227121#.XOAqTMgzbIV
- ***[10]*** Mathworks (consultado por última vez: 06/06/2019): https://es.mathworks.com/
- ***[11]*** Alba Mínguez (2017). Detección de estrés en señales de voz. Trabajo de fin de grado presentado en el Departamento de Teoría de la Señal y Comunicaciones de la Universidad Carlos III de Madrid.
- ***[12]*** Blanca Valdivielso (2017). Medidas de inteligibilidad para predicción del grado de parkinson. Trabajo de fin de grado de Ingeniería de Sistemas de Comunicaciones de la Universidad Carlos III de Madrid.
- ***[13]*** Irene Navidad (2014). Extracción de características espectrales y prosódicas para reconocimiento de emociones. Trabajo de fin de grado de Ingeniería de Sistemas Audiovisuales de la Universidad Carlos III de Madrid.

INTRODUCTION

This document, with the work that has been carried out behind it, aims to facilitate the processes of detecting the disorder of the Specific Language Impairment (SLI) in children through the development of a system for detecting this syndrome by means of the analysis of the voice of potential patients.

The possibility that through several audios simply recorded with a mobile phone by parents to their children and that these audios were sent to a remote system in which the detection system is located, would save time for doctors and speech therapists who would be more released for other work or other patients. To achieve this purpose, the system developed must have a high probability of success.

In any case, it should be noted that the objective of this type of system is not to replace medical personnel and other specialists, but to develop a diagnostic aid tool for this type of language disorders.

With all this, the approach of the problem is focused on developing a tool that can help in the clinical diagnosis of the specific syndrome of language in children by means of the analysis of their voice.

SOLUTION DESIGN

In this chapter 3 of the report, the design of the solution will be discussed, without going into the implementation details that will be developed in the following chapter.

The design of the solution focuses on the following objective: to design an automatic system for the detection of the specific language syndrome in children through the analysis of their voice and the use of machine learning techniques.

In order for the reader to have a general idea of what is explained in the chapter, a clear and logical order will be maintained regarding the design of the solution. The first thing that will be explained will be which is the database on which you are working, as well as its initial structure and the processing that has been done of it to obtain a structure that is of interest to be able to carry out a complete analysis of the problem.

Next, with the database already arranged in the appropriate way to work on it, it will be explained how the different acoustic parameters have been extracted and how they have been organized to work on them. These parameters have been organized in such a way that several tests can be made with several subgroups according to the type of experiment to be performed.

Finally, with the help of automatic learning techniques, several acoustic models have been trained for each of the subgroups and the relevant tests have been carried out, in this case with the Matlab and Excel software tools. The corresponding results were compared in order to draw conclusions.

DATABASE

For the development of the system, we have used a database with text files and audio files in WAV format that correspond to voice audios of the respective patients and control subjects. For the sake of completeness, the children whose voices can be heard in these audios are Czech, so the audios are recorded in this language.

The distribution of the original database is as follows: Two folders, "Healthy" and "Patients", which were immediately renamed to "control" and "patients".

Within the Healthy folder there are 44 folders called H26, H27, H28... H69. Inside the "Patients" folder there are 54 folders called P8, P9...P61. Inside each of these folders (H26, H56, P13, P56...) there are 7 other folders in which there are voice files and text files as a label. The text label mode files have not been used because they contain relevant information for other types of study, but they are not useful for the development of the system implemented in this work. On the other hand, audios that had been repeated were eliminated. In total there are 38 audios per person, either patient or control.

In order to finish explaining the distribution of the database, it is necessary to highlight the exceptions to it. If the majority of the subjects in the database have 38 audios, there are some that lack certain audios.

Therefore, putting together all the patient audios add up to a total of 2050 audio files and putting together all the control audios add up to a total of 1659 audio files. Therefore, a total of 3079 audio files are available to work with.

Regarding the distribution of the data, it was decided to carry out two types of experiments, dependent and independent of the speaker, so 2 large groups were created:

- Speaker dependent: In this group, at the time of performing the tests, some audios of a certain patient or control are used to train the corresponding model and the remaining audios of that same patient or control are used for testing.
- Speaker independent: In this group, all the audios of each control or patient are either included in the training set or in the test set. In other words, part of the speakers belong to the training set while the rest belong to the test set.

Within these two large groups, as already explained, two other subgroups are created. One subgroup will be assigned to train the acoustic model (train) and the other subgroup will be assigned to the prediction or classification itself (test). We proceeded to create the four remaining subgroups as follows:

- Subgroup 1, independent of speaker train: In this folder are included all the audios of the first 32 patients and all the audios of the first 30 controls. It consists of a total of 2346 audios.
- Subgroup 2, independent of the speaker test: In this folder, all the audios of the last 22 patients and all the audios of the last 14 controls are included. It consists of a total of 1363 audios.
- Subgroup 3, dependent on the train announcer: This folder includes audios with 3 vowels, 6 consonants, 6 monosyllables, 3 bisyllables, 2 trisyllables, 2 words of 4 syllables and a word of 5 syllables per patient and 3 vowels, 7 consonants, 7 monosyllables, 3 bisyllables, 3 trisyllables, 2 words of 4 syllables and a word of 5 syllables per control. With the existing exceptions, this subgroup consists of a total of 2372 audios.
- Subgroup 4, dependent on the test speaker: This folder includes 2 vowel audios, 4 consonants, 3 monosyllables, 2 bisyllables, 2 trisyllables, 1 quadrisyllable and a 5 syllable word per patient and 2 vowels, 3 consonants, 2 monosyllables, 2 bisyllables, 1 trisyllables, 1 word of 4 syllables and a 5 syllable word per control. With the existing exceptions, this subgroup consists of a total of 1337 audios.

With these 4 subgroups, it is with which the relevant training or tests are carried out as appropriate.

FEATURE EXTRACTION

This section will explain which acoustic parameters are extracted from the database described above.

The purpose of extracting acoustic parameters or characteristics is to obtain a compact representation of the voice signal that is discriminative, that is to say, that serves to distinguish subjects suffering from specific language syndrome (patients) from those who do not (controls).

The parameters chosen to achieve the objectives of the problem are the Mel-Frequency Cepstrum Coefficients (MFCC).

It is contrasted by other professional and educational studies that the MFCC parameters represent in a compacts way the sound information in the human voice. MFCCs are widely used for speech recognition, language recognition, etc.

This is mainly due to the fact that the Mel scale is used for the extraction of these cepstrales coefficients. The Mel scale resembles the way the human ear perceives sound, giving more resolution to low frequencies than to high frequencies.

In the case of this work, 12 MFCC coefficients are chosen; with these 12 MFCC coefficients and a parameter number 13 that is the log-energy of the signal in each frame, half of the tests explained in this document are carried out. For the other half, in addition to these 13 parameters, the delta coefficients are used, which we will call delta-MFCC, which are derived from these 13 coefficients.

With these 13 MFCC and 13 delta-MFCC we have 26 parameters to work with. In order to perform the tests, text files are generated that store the parameters of each audio. These text files are the ones that will be used for the training or for the classification according to the role played by each audio in the prediction machine to be created.

As for the delta-MFCC parameters, it is decided to include them in the system so that the analysis and training is more complete, since the delta-MFCC contain information on the temporal evolution of the MFCC. This information has been found to be important in other voice-based systems, such as automatic speech recognition.

CLASSIFICATION SYSTEM

In this section we will detail the data that have been destined for train and the data that have been destined for test. In addition, the training model to which the train data have been submitted will be explained.

The data used for the training process has been:

- Set of MFCC parameters corresponding to the audios group: independent of the train speaker.
- MFCC parameter set corresponding to the audios group dependent on the speaker train.
- Parameter set MFCC + delta-MFCC corresponding to the audios group independent of the speaker train.

- Parameter set MFCC + delta-MFCC corresponding to the audios group dependent on the speaker train.
- Set of MFCC parameters, after extraction of the log-energy, corresponding to the group of independent of the speaker train.
- Set of parameters MFCC, previous extraction of the log-energy, corresponding to the group of audios dependent on the speaker train.

To create an acoustic model adapted to the characteristics of our parameters Matlab has been used, more specifically, with the function fitcsvm. The fitcsvm function trains a binary model through a Support Vector Machine (SVM) using Kernel type functions.

The Kernel functions used in the case of this work have been the following:

- Gaussian function.
- Linear function.
- Polynomial function of grade 2.
- Polynomial function of grade 3.

These 4 Kernel type functions have been used in the training of the previously described train groups. Each one of these groups has been trained with all the Kernel functions mentioned, so that the total number of training models worked with has been: 6 x 4 = 24 models.

On the other hand, the groups of audios to be tested are the following:

- Experiments dependent on speaker:
- Clean conditions
- Clean conditions without energy parameter.
- Clean conditions without energy parameter and with elimination of silences.
- Noisy conditions with noise in the living room of a house and 6 different SNRs (from -5 dB to 20 dB in steps of 5 dB).
- Noisy conditions with noise in a meeting room and 6 different SNRs (from -5 dB to 20 dB in 5 dB steps).
- Noisy conditions with noise in a small office and 6 different SNRs (from -5 dB to 20 dB in 5 dB steps).

And the same points that have been detailed for the speaker dependent experiments will be repeated for the independent speaker experiments.

IMPLEMENTATION

This chapter describes how the solution design has been implemented, how the testing has been implemented, and how the final results have been implemented.

DATABASE

To sort and distribute the database, the Windows 10 file explorer was used to store the audio files in different folders.

From there, in order to transfer the audio files from a specific folder to the Matlab environment, an initialization function called read.m is created, which reads a text file (previously created in each folder) that contains 2 columns: the first contains the names of the audio files in alphabetical order and the second contains integer labels. The label will be 0 if the audio of your line is a patient and it will be 1 if the audio of your line is a control.

With this, two vectors are created that will be stored in Matlab, one with all the names of the audios in the folder and the other with their labels.

MFCC EXTRACTION

In order to extract and store the MFCC parameters of each audio file, a Matlab function called escritura.m is created which consists of three phases that are repeated in a loop until the MFCC parameters are written frame by frame in a different text file for each audio:

1. Reading and storing the audio file in Matlab.

2. Extraction and storage of the MFCC parameters of each audio file.

3. Writing the MFCC parameters extracted in step 2 in a text file with the same name as the original audio and different extension.

Step 1 is implemented mainly through Matlab's audioread.m function which, passing the name of an audio file as a parameter, returns a vector with the audio samples contained in that file to be interpreted by Matlab and the sampling frequency.

Step 2 is implemented mainly through the function melcepst.m which, as output obtains an array consisting of 13 columns (each column corresponds to a parameter mfcc, plus the log-energy) and as many rows as the number of frames of the audio has.

Step 3 is implemented mainly through the Matlab function fprintf.m.

TRAINING

To train a model, you need input data with an appropriate structure. The basic structure chosen has been a matrix, in which each row represents a specific audio and each column represents an MFCC parameter. So, for example, if you want to train subgroup 1, independent of the speaker train, which contains 2346 audios, you will need to obtain a matrix of dimensions (2346 x 13).

The Matlab matrix.m function has been created, which, one by one, runs through the matrices that contain the MFCC parameters of each audio, averaging each of these MFCC parameters. In this way, for each audio, we obtain a vectorial representation (1x13) of the MFCC and non-matrix parameters (number of frames x 13), which is what we had up to this point.

Once you have the input data, the fitcsvm function has been used to train the model. The fitcsvm function trains a binary model through a Support Vector Machine (SVM) using Kernel type functions.

CLASSIFICATION

Matla's predict.m function has been used, which, passing as input data a trained model and a test group on which to perform the classification, returns a size label vector (1 x number of cases to classify). Each one of the vector labels corresponds, in the case of this problem, to each one of the audios of the group of test exposed to classification, and the labels will be either a "0" or a "1", that is to say, or patient or control.

RESULTS

To extract the data from the classification, simply copy the classification vector and paste it into a column in a spreadsheet, "Prediction. Two columns to the left of this column, the name of the audios corresponding to the test group subjected to classification has been written, and between the middle of these two columns, that is to say, the second, the real labels of each audio have been written. Columns 4, 5, 6, 7 and 8 are relations between the columns "Real label" and "Prediction" that will serve to obtain the values necessary for the calculation of the final results. Column 4 marks "1" when there is a hit and "0" when there is a prediction error. Column 5 marks true positives with a "1". Column 6 marks false positives with a "1". Column 7 marks true negatives with a "1". Column 8 marks false negatives with a "1.

Given that the task posed in this work is binary, we have chosen to use as evaluation measures, precision, Recall and F-score, whose formulas need the number of true positives, true negatives, false positives and false negatives previously extracted.

Therefore, accuracy can be defined as the percentage of positive predictions that were correct, the Recall, as the percentage of positive cases that were captured and the F-score, as a single weighted value of accuracy and Recall. These values for each test have been stored in Excel.

CONCLUSIONS

The conclusion is that, this work is a good first step to explore more types of conditions in the audios, whether it works well or you have to put variations for other languages etc. In short, it serves as a basis for exploring the future lines explained in the previous section. The F-score obtained for the case of speaker Independence and clean speech, with MFCC+delta-MFCC and the Gaussian Kernel is quite good, very close to 1, more specifically, 0,9817.