DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DEEP LEARNING PARA LA DETECCIÓN DE COVID POR LA TOS CON AUMENTO DE DATOS

Trabajo de Fin de Grado

presentado en

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

por

David Marchan del Pino

En cumplimiento parcial

de los requisitos para el grado en

INGENIERÍA EN TECNOLOGÍAS Y SERVICIOS DE LA TELECOMUNICACIÓN, MENCIÓN DE SISTEMAS AUDIOVISUALES

Director: Dr. Francisco Javier Hernando Pericas

Barcelona, Mayo 2022

Abstract

In recent years, COVID-19 has had a major impact on today's society and has not gone unnoticed in the world of deep learning. Cases can now be studied using image analysis or in the world of audio analysis.

This report includes a study involving the use of Deep Learning as a basis for the detection of COVID-19. For this purpose, different techniques will be used to use a reference system normally used as speaker recognition to achieve the detection of COVID-19, all from audios with patient coughs. Data augmentation techniques and the use of weights to compensate for the unbalanced classes resulting from the use of an unbalanced database will be used for the implementation. The Cross Entropy Loss function will be used for this purpose. In addition, another system was used for the implementation of a practice taught by Jose Adrian Rodriguez Fonollosa with a different system of the model, data extraction and loss function. The use of the latter system will be for the purpose of comparing the results of the two systems.

The report also contains the sections where the results obtained after these experiments are shown, as well as some conclusions and future development proposals.

Resum

En els darrers anys el COVID-19 ha suposat un gran impacte en la societat actual i al món del deep learning tampoc va passar desapercebut. Actualment es poden estudiar els casos mitjançant l'anàlisi d'imatge o el món de l'anàlisi de l'àudio.

En aquesta memòria consta lestudi que involucra lús del Deep Learning com a base per a la detecció del COVID-19. Per això s'utilitzaran diferents tècniques per a partir d'un sistema de referència normalment utilitzat com a reconeixement de l'interlocutor aconseguir la detecció del COVID-19, tot això a partir d'àudios amb tos de pacients. Per a la implementació es van fer servir tècniques d'augment de dades o Data Augmentation i l'ús de pesos per a la compensació de les classes desbalanceades que prové de l'ús d'una base de dades desbalanceada. Per això s'utilitzarà la funció de pèrdua Cross Entropy Loss. Addicionalment es va fer servir un altre sistema que s'utilitza per a la realització d'una pràctica impartida per Jose Adrian Rodriguez Fonollosa amb un sistema diferent del model, extracció de les dades i funció de pèrdua. L'ús d'aquest darrer sistema serà per comparar els resultats dels dos sistemes.

A la memòria consten també els apartats on es mostren els resultats obtinguts després d'aquests experiments així com unes conclusions i futures propostes de desenvolupament.

Resumen

En los últimos años el COVID-19 ha supuesto un gran impacto en la sociedad actual y en el mundo del deep learning tampoco pasó desapercibido. Actualmente se pueden estudiar los casos mediante el análisis de imagen o con el mundo del análisis del audio.

En esta memoria consta el estudio que involucra el uso del deep learning como base para la detección del COVID-19. Para ello se utilizarán distintas técnicas para a partir de un sistema de referencia normalmente utilizado como reconocimiento del interlocutor conseguir la detección del COVID-19, todo ello a partir de audios con tos de pacientes. Para la implementación se usaron técnicas de aumento de datos o Data Augmentation y el uso de pesos para la compensación de las clases desbalanceadas que proviene del uso de una base de datos desbalanceada. Para ello se utilizará la función de pérdida Cross Entropy Loss. Adicionalmente se usó otro sistema que se utiliza para la realización de una práctica impartida por Jose Adrian Rodriguez Fonollosa con un sistema distinto del modelo, extracción de los datos y función de pérdida. El uso de este último sistema será con objeto de comparar los resultados de los dos sistemas.

En la memoria constan también los apartados donde se muestran los resultados obtenidos tras dichos experimentos así como unas conclusiones y futuras propuestas de desarrollo.

Agradecimientos

En primer lugar, me gustaría agradecer a las partes directas de este proyecto que son Francisco Javier Hernanando y Marc Sanchez. Francisco Javier por su asesoramiento a lo largo del proyecto y sus múltiples ayudas en los momentos que este se quedaba profundamente enquistado. Marc Sanchez por otra parte fue un compañero que me acompañó en gran parte del proyecto.

En segundo lugar, este proyecto contó también con el asesoramiento de diferentes personas, en especial tenemos a Miquel Àngel India, por su ayuda que nos permitió a Marc y a mí tener una base sólida sobre la cual trabajar y llevar a cabo una tarea de investigación.

Por último agradecer a todas aquellas personas que participaron en el consejo o asesoramiento en diferentes momentos del proyecto: Jose Adrian Rodriguez Fonollosa, Víctor Emilio Hernandéz Leal, Santiago Escuder Folch, Daniel Garriga Artieda y Javier Ferrando Monsonis.

Historial de revisiones y registro de aprobación

Revisión	Fecha	Propósito
0	27/03/2022	Creación
1	13/05/2022	Revisión

Nombre	e-mail
David Marchan del Pino	david.marchan@estudiantat.upc.edu
Francisco Javier Hernando Pericas	javier.hernando@upc.edu

Escrito por:		Revisado y aprobado por:	
Fecha	27/03/2022	Fecha	13/05/2022
Nombre	David Marchan del Pino	Nombre	Francisco Javier Hernando Pericas
Posición	Autor	Posición	Supervisor

Índice

Lista de Figuras

Lista de Tablas

1. Introducción

1.1 Motivación

Durante mi paso por las asignaturas del grado siempre tuve curiosidad por aquello que despertaba en mí una atracción. Normalmente, siempre fueron temas relacionados con la electrónica y la programación. Tras varios años, estos campos por desgracia no ayudaban a satisfacer una parte del empeño que tengo desde mí infancia por inventar o desarrollar una idea que haga un bien a la sociedad si por un casual se da el caso.

En el grado, siempre que se acercaba la fecha o la ocasión para tener que decidir un tema para el trabajo de fin de grado, era del grupo en el cual por desgracia nunca sabia por que campo elegiría. Esto fue así hasta que cuando cursaba la especialidad de audiovisuales me encontré con el mundo de la inteligencia artificial y sus grandes posibilidades.

El procesado de señal, como lo conocemos, abre la ventana al deep learning para que gracias a los conocimientos de imagen y voz podamos desarrollar nuevas estrategias para encontrar soluciones a problemas del día a día. Si bien es cierto que la tos no tiene características del contenido como podría tener una frase, siempre podrá albergar características de algún tipo de patología. En primera instancia, los estudios más comunes sirven para reconocimiento del interlocutor o de transcripción de audio a texto. No obstante, siempre se pueden obtener las características para otros fines.

Es evidente que en nuestros días la inteligencia artificial juega un papel crucial para el desarrollo de nuevas ideas e investigaciones, así pues buscando un tema que tuviera conexiones con la sociedad, me encontré con el proyecto propuesto por Francisco Javier, el uso del deep learning para la detección de COVID-19 mediante la tos.

1.2. Objetivos

El principal objetivo de este proyecto es usar un sistema basado en el Deep Learning o aprendizaje profundo para poder detectar el COVID-19 mediante audios de la tos, usar diferentes técnicas y comparar los resultados obtenidos para sacar unas conclusiones y propuestas de futuro desarrollo. Los objetivos son:

- Readaptar el sistema proporcionado basado en el reconocimiento interlocutor para que sea capaz de clasificar entre pacientes con sintomatología de ser positivo o negativo a partir de grabaciones de tos. El sistema en su base empleará el análisis espectral de las señales de voz.

- La implementación de diferentes configuraciones en la base de datos para comparar dos sistemas con la misma información.

- Estudiar los resultados obtenidos y elaborar el informe usando la técnica conocida como Data Augmentation que se suele usar para mejorar los resultados obtenidos en una primera instancia.

1.3. Metodología y contexto

El proyecto surge de la propuesta por parte de Francisco Javier Hernando Pericas en la oferta de la ETSETB de trabajos de final de grado. No existe un proyecto previo en el centro que tenga las mismas características, no obstante sí que se usó la misma base del código en la realización de otros estudios basados en el Deep Learning.

En cuanto a la estructura, este proyecto es el típico problema de clasificación. Consta de las partes de extracción de características, clasificación, prueba o test y resultados.

El proyecto partía de la colaboración de Marc, el autor del documento y supervisado por Francisco Javier. Una vez se alcanzó un sistema de referencia que funcionaba, se separan los caminos de los dos estudiantes y se desarrolla un estudio independiente.

El plan inicial era mucho más ambicioso y no se pudo llevar a cabo debido a los contratiempos, tiempo de investigación y desarrollo de una base que funcionase con los datos que se disponían del COVID-19.

Finalmente, la parte en la que se especializa este proyecto es en el uso de la DA para lograr un mejor resultado en la clasificación del covid y a su vez la comparativa con otro sistema utilizado en el centro.

Podemos ver en la siguiente figura (Figura 1.3.1) el Work Breakdown Structure que se planteó y que se terminó por seguir constituido por los tres WP (Work Packages).

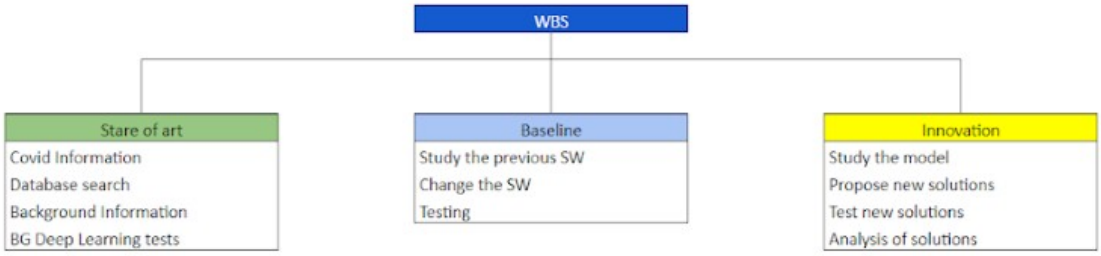

Figura 1 WBS del proyecto.

A continuación se muestra el diagrama de Gantt del proyecto, teniendo en cuenta la prórroga que se solicitó por no conseguir el objetivo de poder tener el desarrollo del proyecto en la fecha inicialmente propuesta.

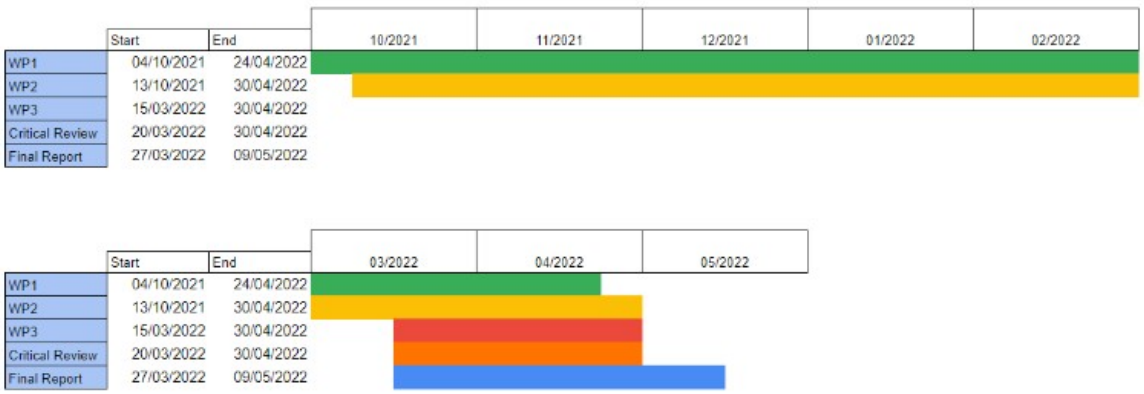

Figura 2 Gantt.

1.4. Estructura de la Memoria

La memoria de este proyecto consta de cinco apartados distintos, entre ellos. En primera instancia encontramos la introducción a la memoria. En segundo lugar, encontramos un resumen de la investigación que hay relacionada con el proyecto en el caso del COVID-19. Concretamente, se mencionan los siguientes temas:

- Deep Learning
- Deep Learning aplicado al COVID-19
- Data Augmentation en Deep Learning

En tercer lugar, nos encontraremos con la información que hace referencia a cómo se desarrolló el proyecto. Podremos ver en sus distintos apartados con que base de datos se trabajó, el sistema del cual se parte adaptado al COVID-19, el sistema con el que haremos la comparación y la explicación de la mejora con aumento de datos. En el cuarto punto, se expondrán los resultados obtenidos de los experimentos realizados. En la quinta parte, podremos encontrar la propuesta económica obligatoria en la memoria. Finalmente, en último lugar, podremos encontrar una conclusión y propuestas a futuro en el caso de que alguien retome el proyecto o que se use este para la inspiración de otro.

2. State of the art

2.1. Deep Learning

La IA (Artificial Intelligence o Inteligencia Artificial) cada vez se utiliza más en el desarrollo de soluciones del día a día. Concretamente, el subconjunto de técnicas de aprendizaje automático, el aprendizaje profundo o deep learning es el escogido para la realización de este estudio. Usualmente, se usa el aprendizaje profundo para asignar características de entrada a una salida. Este proceso de se produce dentro de múltiples capas conectadas que a su vez contienen múltiples neuronas,.Cada neurona es una unidad de procesamiento matemático que, combinada con todas las demás neuronas, está diseñada para aprender la relación entre las características de entrada y la salida. Algunas características del deep learning son:

- Normalmente, se necesita una gran cantidad de información o datos para realizar un entrenamiento que tenga sentido y así poder obtener unas predicciones fiables.

- El hecho de la cantidad de datos a procesar implica una gran capacidad de computación, por ello lo más normal en el momento de ejecutar los cálculos es el uso de una GPU potente para no efectuar entrenamientos de una duración excesiva.

- El sistema implementado tiene como objetivo extraer las características de los datos entregados a un alto nivel y a partir de ellas crea unas nuevas.

- El aprendizaje se lleva a cabo de extremo a extremo del sistema, cosa que no sucede en tecnologías como el machine learning que los hace con aprendizajes con pasos más pequeños.

- El tiempo de ejecución del entrenamiento va en concordancia con el número de capas por donde pasa la información, a más capas más tarda el sistema en aprender, pero mayor extracción de los datos se consigue.

- En la salida del sistema no hay un formato estándar. El formato puede ser implementado de tal manera que devuelva desde un valor numérico o cualquier otro tipo de formato que aporte información como podría ser un audio dictando el resultado.

La naturaleza de los datos disponibles para entrenar una DNN (Deep Neural Net) es un punto importante. El aprendizaje supervisado puede producirse si los datos utilizados para entrenar la DNN están etiquetados, lo que significa que la salida (por ejemplo, el ser positivo en COVID-19 o no se reduce a un problema de clasificación binaria) es conocida.

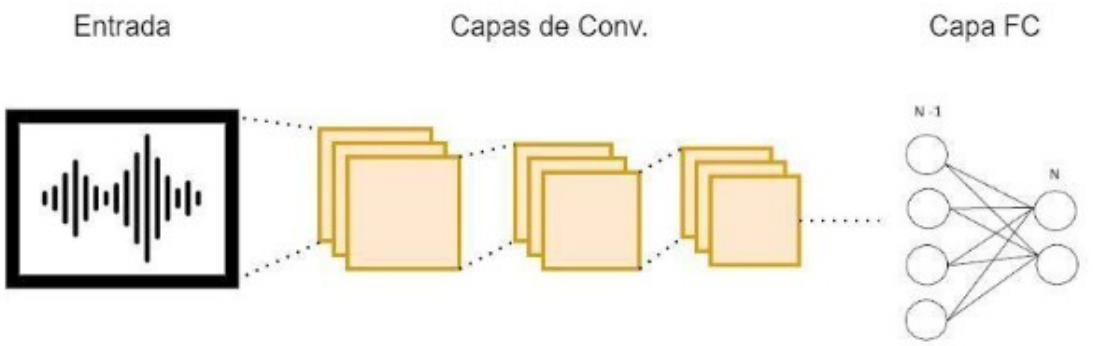

Figura 3 Esquema de alto nivel de una CNN.

Los algoritmos de DL son estructuras matemáticas complejas con varias capas de procesamiento que pueden separar las características de los datos, o representaciones, en varias capas de abstracción. En el aprendizaje supervisado, una DNN pasa secuencialmente los datos de características de entrada de las neuronas de una capa a las neuronas de la siguiente capa durante un proceso que se repite muchas veces, a menudo miles, y dicho ciclo o iteración se conoce como época. Cada neurona acepta entradas ponderadas de otras múltiples neuronas. Si se supera el umbral de activación, la neurona generará una salida que se combina con un valor de peso antes de pasar a las múltiples neuronas de la siguiente capa.

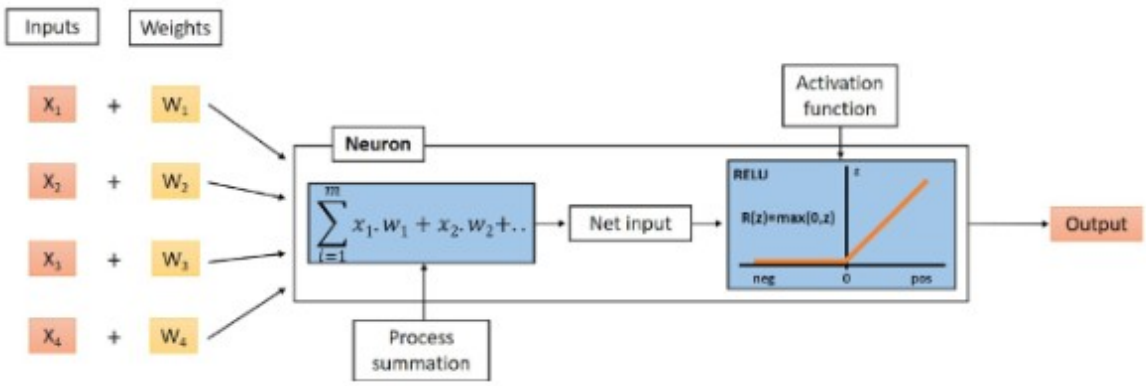

Figura 4 Suma y activación dentro de una sola neurona computacional[1].

Para lograr un entrenamiento tras el esquema anterior se tiene que computar una variable que nos permita medir cuánto estamos mejorando respecto a la mejor marca anteriormente registrada. Para ello existe la función de pérdida, esta depende de la naturaleza del modelo, pero es esencialmente una herramienta matemática para evaluar el rendimiento de un modelo en unos datos determinados, con un valor bajo indica un mejor rendimiento que un valor más alto. Durante las múltiples épocas de aprendizaje, el modelo pretende minimizar el error y encontrar la combinación de los valores de los pesos que genere el menor valor de error. Esta actualización repetida de los pesos en función del tamaño del error es lo que se denomina "aprendizaje".

Se pueden utilizar conjuntos de datos muy grandes para encontrar correlaciones, no obstante no siempre se puede disponer de ellos y hay que recurrir a diferentes técnicas para ensanchar el volumen de datos disponible.

En la última capa de toda aplicación que use el aprendizaje profundo encontramos la capa FC o fully connected. Estas son las capas encargadas de realizar la entrega a la salida la tipología de los resultados, dando por ejemplo en el análisis de imágenes e identificar si a la entrada un animal es un gato, un perro o una jirafa en el caso de clasificación de imagenes.

2.2. Deep Learning aplicado al COVID-19

En un primer contacto con el proyecto se basó el estudio en el artículo que podemos encontrar en [2], se recoge la conclusión:

“Creemos que se trata de resultados prometedores que apoyan la idea de que existen alteraciones en los patrones respiratorios, causadas por COVID-19, que pueden detectarse a partir de muestras de tos o de habla.”

Esta conclusión nos hace pensar en la viabilidad del proyecto y en el estudio que se puede realizar para intentar superar las marcas conseguidas hasta el momento. Es obvio que el uso de técnicas conocidas para la extracción de datos en el habla serían de gran utilidad, se demuestra cuando una gran parte de los proyectos tomados en la dirección del estudio del COVID-19 utilizan los MFCCs para la extracción de características.

No obstante, añaden, por otra parte, en el mismo artículo la siguiente conclusión:

“Dado que una buena generalización es una capacidad esencial para que estos sistemas sean útiles para cualquier escenario del mundo real, nuestro trabajo futuro se centrará en evaluar la generalización, trabajando con conjuntos de datos más amplios y de datos, para el entrenamiento del sistema, y con datos más curados, con etiquetas vinculadas al resultado de la PCR estándar, para la evaluación del sistema.”

Esta conclusión es reveladora debido a que en la búsqueda de diferentes documentos que se dedicasen a la detección del COVID-19 con sistemas basados en deep learning se pueden encontrar sistemas o soluciones de un carácter híbrido, juntando técnicas de procesado de imágenes con radiografías de los pulmones. El objetivo final es apoyar los resultados con deep learning orientado a la voz exclusivamente con otros sistemas que aportan también información para la detección de la dolencia. El resultado de dichos experimentos siempre son de gran valor académico, ya que no se quedan con solo una fuente de datos, sino que gracias al avance tecnológico de los últimos años en potencia de cálculo con GPUs de cada vez más potencia se pueden volcar más datos a los distintos sistemas utilizados para el estudio de distintas problemáticas o desafíos. Un ejemplo claro lo encontramos en el artículo [3] donde se propone una solución híbrida como se muestra en la imagen extraída de ese mismo documento.

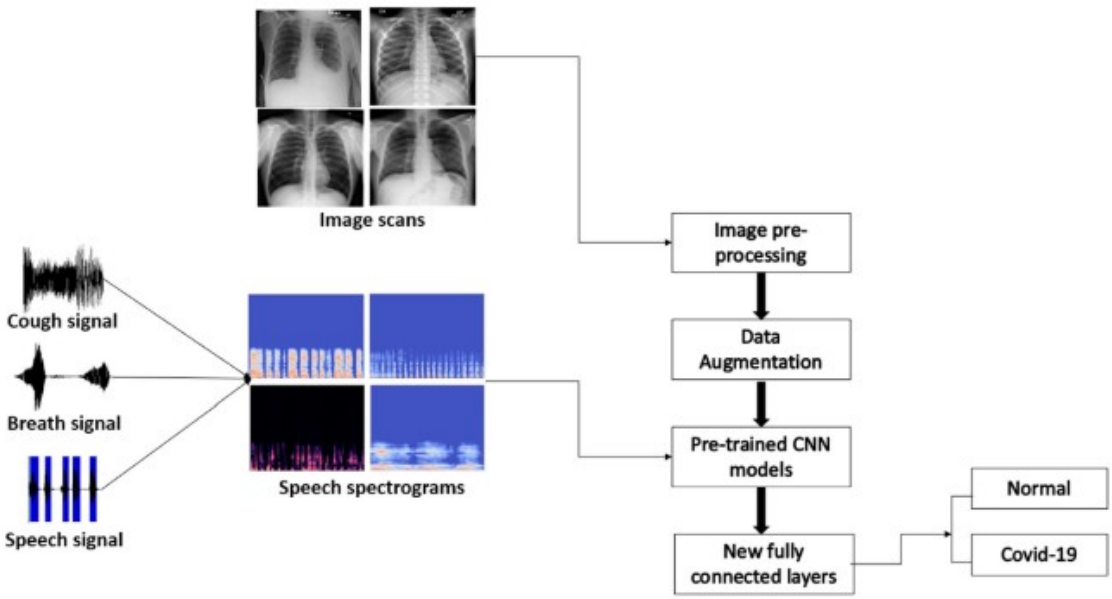

Figura 5 Detección de COVID-19 a partir de radiografías y señales de voz[3].

Extracción de los datos:

En la extracción de las características del audio se observa que lo más común es el uso de las características espectrales y por norma general se suelen usar las relacionadas con la escala Mel, es decir, en otras palabras, los MFCC o coeficientes cepstrales de frecuencia Mel o el espectrograma Log-Mel. Incluso en el paper [3] hacen un estudio del coste o posibles valores del número de filtros Mel que son adecuados o que inducen a mejores resultados.

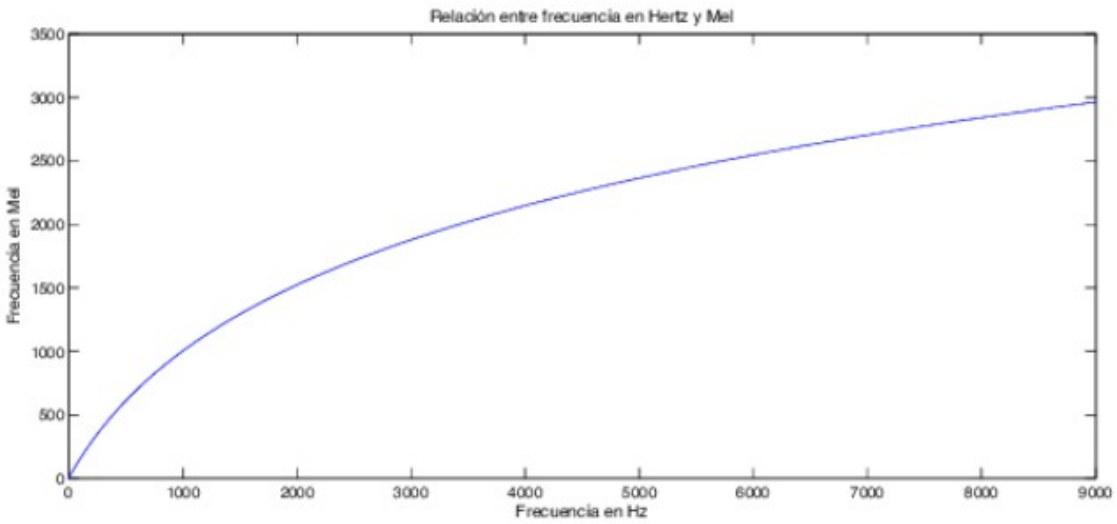

Figura 6 Relación entre frecuencia en Hz (eje x) y en escala Mel (eje y)[4].

Para sintetizar los MFCCs son coeficientes para la representación del habla basados en la percepción auditiva humana. Estos extraen características de las componentes de una señal de audio para que sean adecuadas para la identificación del contenido relevante que aporte información. Hay multitud de otras técnicas para la obtención de las características de los audios, pero los MFCC son a la vez una herramienta muy potente y estandarizada en tareas de reconocimiento del interlocutor.

La escala Mel se construyó basándose en test psicoacústicos utilizando tonos que se podían percibir a una misma distancia de otros. Esta parte de los 1000 Hz cuyo valor se le otorga 1000 mel.

Para el cálculo de la escala completa se usa la siguiente fórmula matemática obteniendo así los valores en las frecuencias siguientes:

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

El uso del Log-Mel-Spectogram se obtiene haciendo un espectrograma de la señal, filtrando los diferentes segmentos de la señal para posteriormente hacer una transformación al dominio frecuencial. El espectrograma resultante pasa por un banco de filtros Mel para obtener sus respuestas frecuenciales equivalentes en la escala Mel. Finalmente, se aplica una función logaritmo a la salida para obtener la señal en el espectrograma Log-Mel.

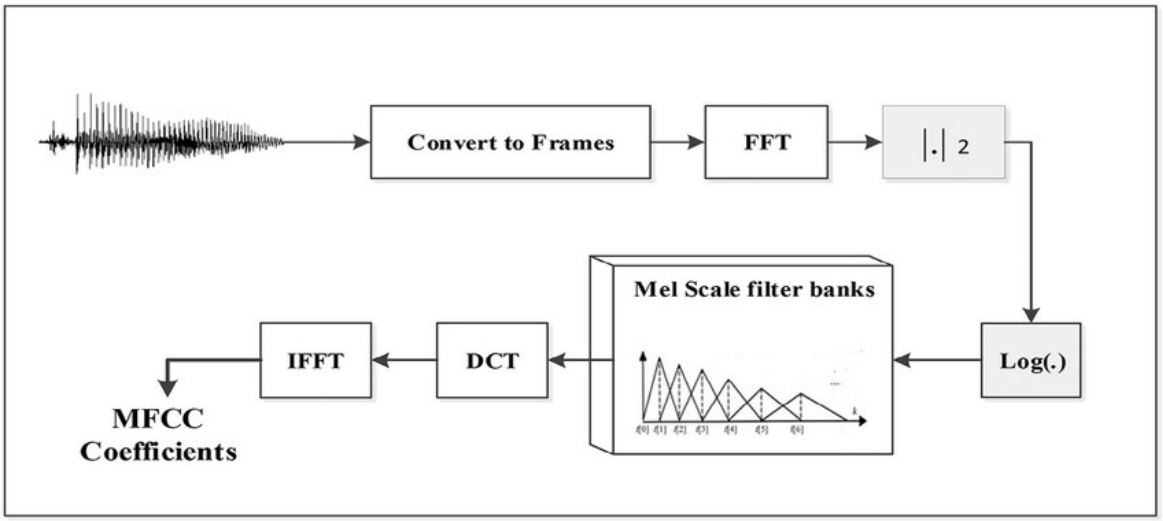

Figura 7 Extracción de los coeficientes cepstrales de frecuencia mel[5].

Redes Neuronales:

Es un hecho que se suelen usar redes neuronales entrenadas para el reconocimiento del habla por el hecho de donde provienen los audios. Estas redes son costosas de entrenar y en ocasiones se parte con redes ya pre entrenadas previamente, como son la VGG16 o VGG19, al tener un número grande de capas. Estas suelen utilizarse en tareas como el reconocimiento del interlocutor entre otros campos, como podemos ver en [5]. El uso de estas redes consume muchos recursos si tenemos en cuenta que en las tareas típicas en las que intervienen suelen estar alimentadas con una gran cantidad de datos y en ocasiones los entrenamientos pueden llegar a durar semanas si no se proponen soluciones que reducen la carga computacional al sistema resultante.

La CNN es una de las redes neuronales más populares. Se ha empleado ampliamente en el aprendizaje automático, concretamente en tareas como el reconocimiento de imágenes, como el procesamiento del lenguaje natural (NLP), en el conjunto de datos de clasificación de imágenes y computer vision.

Las CNN se componen esencialmente de tres bloques: las capas convolucionales, las capas de agrupación y las capas totalmente conectadas o fully-connected. Cuando estas capas se encadenan una detrás de otra, se puede decir que la CNN ha sido implementada.

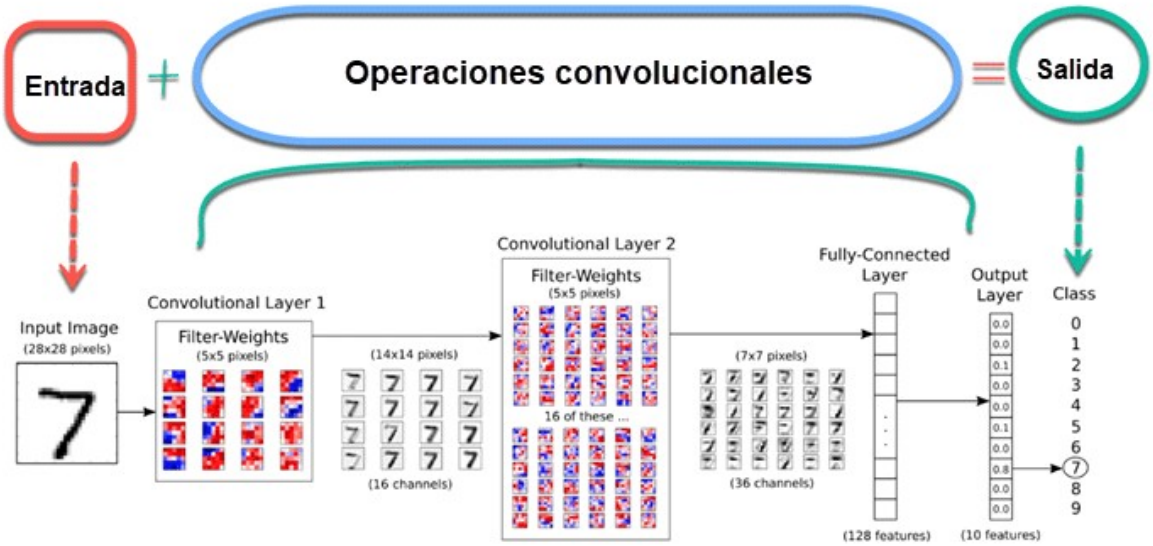

Figura 8 Representación de una CNN[6].

La capa convolucional es responsable de la determinación y el cálculo de la salida de las neuronas, que están conectadas a regiones locales de la entrada, utiliza el producto escalar entre sus pesos y la región que está conectada directamente al volumen de entrada.

La capa de no linealidad, que es la siguiente capa después de la capa convolucional, puede emplearse para ajustar la salida generada. El uso de esta capa consiste en saturar o limitar la salida generada. En esta capa podemos encontrar las funciones denominadas RELU o la unidad lineal rectificada.

Tras la etapa de convolución y las de no linealidad existen las pooling layers, que se encargan de realizar el downsampling de los datos resultantes para poderlos pasar a la siguiente capas y reducir el número de parámetros entregados. Finalmente, las capas fully connected se encargan de calcular las puntuaciones de las clases que serán entregadas para comparar las predicciones con la entrada.

Funciones de pérdida:

El entendimiento de las funciones de las funciones de pérdida es crucial para el entendimiento de la parte de correr el código conocida como entrenamiento o train. Estas son las encargadas de gestionar la optimización de los parámetros de la red, variando así los pesos de las neuronas, generando así para un valor de entrada una salida concreta.

La actualización de los pesos tiene lugar una vez se terminó por recoger los valores a la salida del sistema y a partir de la derivada de la función respecto a los pesos de la última capa, es capaz de encontrar los valores de los pesos

actualizados. Este proceso se aplica en toda la red, en consecuencia proboca la modificación de toda la red al completo, este concepto se conoce como backpropagation. Algunos ejemplos de las funciones de pérdida más usadas son la Softmax Loss o Cross-Entropy Loss o la BCE With Logit Loss.

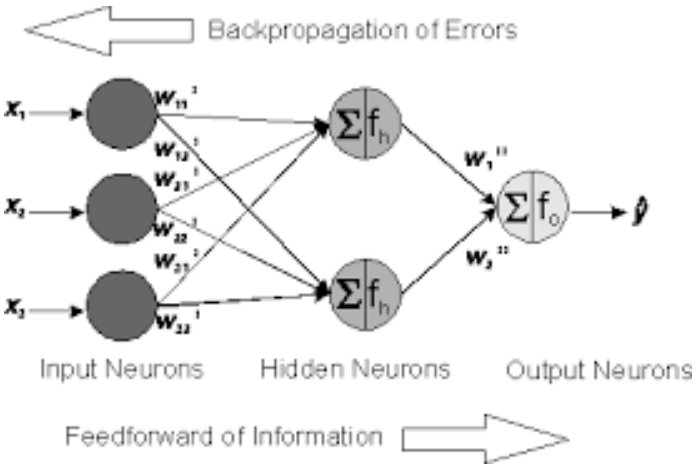

Figura 9 Backpropagation.

2.3 Data Augmentation en Deep Learning

La DA (Data Augmentation) o en castellano conocido como el aumento de datos es un procedimiento por el cual en la construcción de la base de datos sobre la cual se pretende trabajar, se pretende usar para aumentar la base de datos con la particularidad de que las copias son pequeñas variaciones de los datos originales[7].

A priori, el hecho de que una pequeña variación de la información introducida en el sistema puede aportar que este se comporte de manera distinta en el momento del aprendizaje puede resultar un tanto contradictoria. No obstante, por ejemplo, en la detección de coches, si tenemos siempre fotografías de un vehículo mirando hacia el lado derecho, el sistema tan solo aprenderá a detectar los objetivos que circulen en esa dirección, dando como resultado que si a la entrada se le aparece un coche con las mismas características, pero circulando a la izquierda el modelo no será capaz de reconocer que eso es un vehículo con una orientación distinta a la previamente entrenada, generando así una falla en la clasificación o detección del objetivo. Así pues, en el caso del mundo del deep learning es conveniente en este caso introducir un aumento de datos con las imágenes de los vehículos duplicados en diferentes orientaciones para así poder hacer un mejor entrenamiento y conseguir un modelo mucho más robusto. Es evidente que el uso de esta técnica para algunos casos no

tendrá el mismo valor que datos con un origen distinto, como podrían ser en el caso anterior de diferentes modelos de coches.

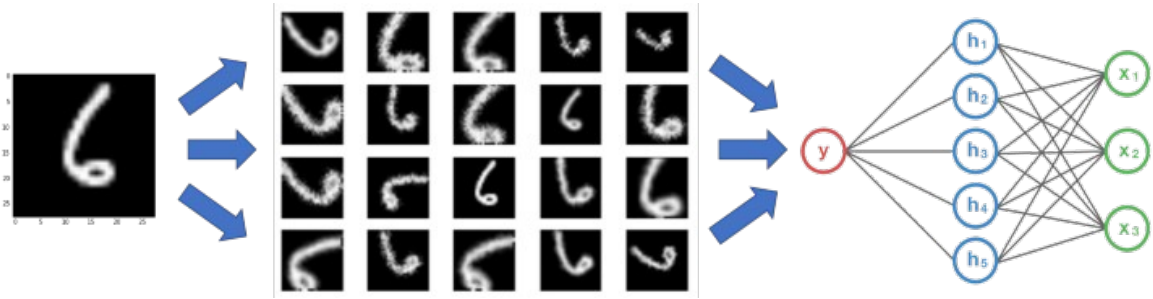

Figura 10 DA aplicado a una imagen.

Comúnmente la DA se usa exclusivamente en los datos de entrenamiento para así incrementar la base de datos y evitar el over-fitting, reducir los desajustes y para hacer el modelo resultante mucho más robusto. Siempre que se logre introducir más información en el sistema, está claro que el consumo de recursos aumentará a costa de que el entrenamiento mejore y el sistema en el momento de validar los datos tenga más información para separar los datos en las clases resultantes.

En el mundo del audio existen multitud de maneras distintas de obtener un aumento de los datos. En última instancia, una vez se consiga una variación del audio, este siempre será válido, mientras que no se pierdan las características que se pretenden detectar. Un ejemplo claro sería que si queremos detectar un sonido puntual en un conjunto de datos formados por audios y así discernir, por ejemplo, el hecho de que una alarma salte, deberemos procurar que los cambios siempre sean de tal manera que la información de los pitidos no sea enmascarada por las transformaciones que les hagamos a los datos. De otra forma, perderíamos toda capacidad para que el sistema en esos datos de entrenamiento pudiese aprender a identificar las alarmas correctamente.

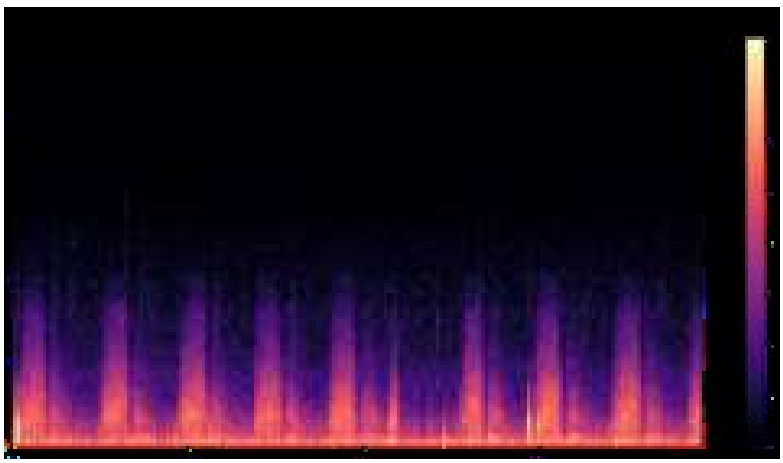

Figura 11 Mel-Spectogram de un audio normal[8].

Algunas de las técnicas más utilizadas para el aumento de datos en el mundo del audio son la introducción de ruido de diferentes procedencias, el desplazamiento temporal y el cambio del pitch.

La primera y más usada es el uso del ruido para generar audios iguales que los de origen, pero con la particularidad de que contienen ruido de mayor o menor amplitud de la señal, para así tener distintos valores de SNR o relación señal ruido, en dB o decibelios.

$$SNR(dB) = 10 \log\left(\frac{Ps}{Pn}\right)$$

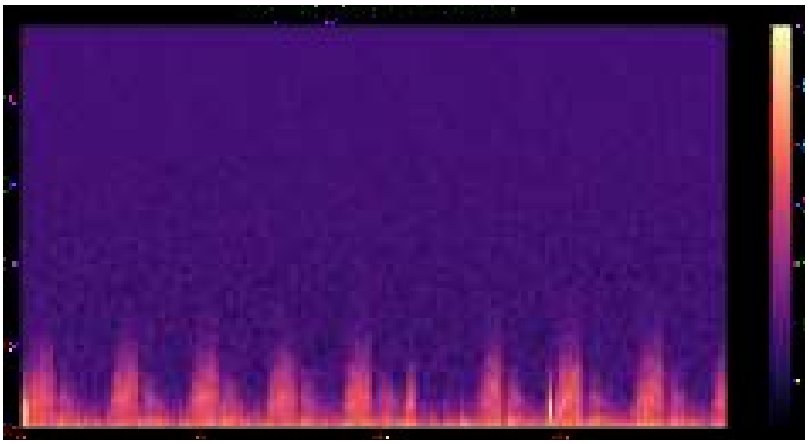

Figura 12 Mel-Spectrogram con ruido añadido en un audio normal[8].

La segunda técnica es conocida como time shifting o desplazamiento en el tiempo, la particularidad en esta opción es que el audio resultante es una traslación en tiempo del audio original. Este proceso se consigue mediante el cambio de la frecuencia de muestreo como indican en [8].

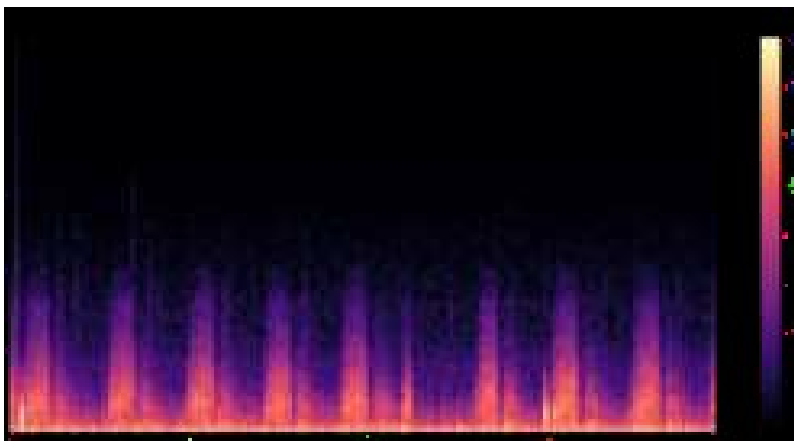

Figura 13 Mel-Spectrogram con desplazamiento temporal en un audio normal[8].

Finalmente, otro recurso para lograr el DA se suele usar el pitch shifting o variación del pitch. El proceso consiste en cambiar el pitch de la señal de audio sin afectar a la velocidad de este. Esta técnica es útil, por ejemplo, en el caso de no tener suficientes candidatos en el reconocimiento del interlocutor para tener en cuenta los márgenes del pitch, aumentar la base de datos de la clase “Hombre” y la clase “Mujer”.

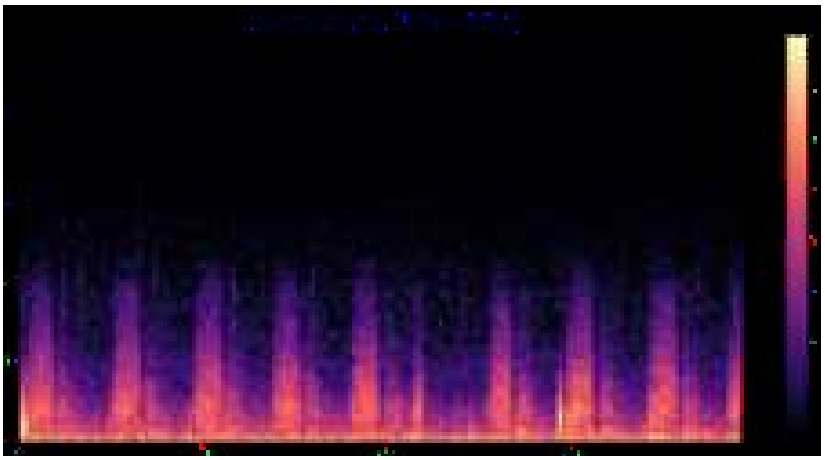

Figura 14 Mel-Spectrogram con variación del pitch en un audio normal[8].

3. Desarrollo del proyecto

3.1 Base de datos CCS

En este proyecto se utilizó la base de datos CCS. La CCS es una parte de la base de datos que se usó en el INTERSPEECH 2021 Computational Paralinguistics ChallengE (ComParE) con fines de investigación [9]. Se disponía de otra base de datos hermana de este concurso llamada CSS, que constaba de audios donde se repetía una misma frase con motivo de la utilización de esta otra para el reconocimiento del COVID-19 en el caso del habla, pero se descartó emplear la CSS con motivo del título del proyecto debido a que está enfocado en el caso particular de la tos.

La CCS en su interior se incluyen archivos .wav de sonidos de tos, así como algunas características anteriormente extraídas, y etiquetas. En cada grabación se incluye una o varias toses. En la fuente se incluyen archivo de metadatos csv para los datos de CCS, que incluye información demográfica como la edad y el sexo, así como otra información como las identificaciones de los usuarios (Uid), el historial médico, los síntomas declarados, el estado de fumador, la hospitalización o no, y los resultados originales de las pruebas de covid.

Dentro de las grabaciones constaban audios con una frecuencia de muestreo inferior a 16 kH. Aunque las grabaciones de audio originales tenían diferentes frecuencias de muestreo, las muestras proporcionadas se volvieron a muestrear a 16 kHz.

Con motivo de la extracción de las etiquetas se tuvo en cuenta el campo de positivos y negativos solamente para generar las etiquetas adaptadas al sistema que se proporcionó para el proyecto, de esta manera no se tuvo que hacer divisiones más pequeñas de los datos debido al tamaño de la base de datos. Con este razonamiento se consigue una clasificación binaria de los datos, refiriéndose a las etiquetas, siendo las dos las resultantes clases, la “Positivo” marcada con un 1 y “Negativo” marcada como 0.

La base de datos se entregó ya separada en sus distintas partes de entrenamiento, validación o desarrollo y los audios que constituyen el test para evaluar los resultados. Hay que tener en cuenta que la base de datos está distribuida de tal manera que el número de audios en el test siempre será mayor que en validación y a su vez esta última siempre será mayor que el tamaño del test.

Fase	Positivos	Negativos
Entrenamiento	71	215
Validación	48	183
Test	39	169
Total	158	567

Tabla 1 Partición de los audios que forman la base de datos CCS.

3.2. DASV adaptado al COVID-19

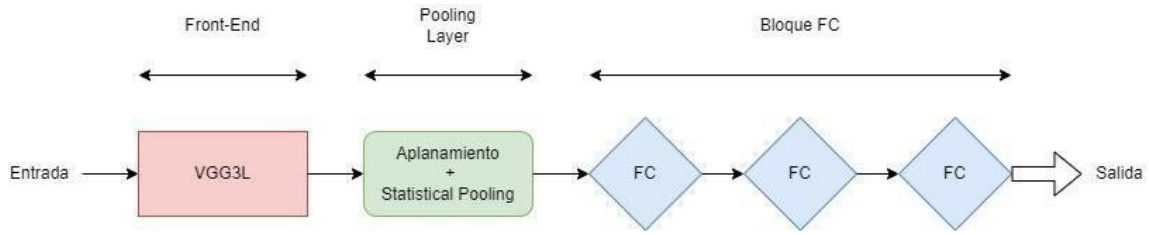

Figura 15 Bloques del modelo del sistema de referencia antes de adaptar.

DASV es una modificación del clasificador utilizado anteriormente en el departamento [10] cuya implementación fue facilitada por Miquel India el autor del proyecto.

El sistema consta con una CNN, a la entrada de esta se le deben pasar los datos de manera que estén formados por matrices bidimensionales. El criterio o la manera en la que se introducen los datos es mediante el uso del espectrograma, más concretamente se le introduce el Mel-Espectrograma generado con la función en el código que se dedica exclusivamente a esa tarea.

Para la generación del Mel-Espectrograma se usó una ventana de 25 ms empleando, concretamente, una ventana de Hamming y un solapamiento de la ventana de 10 ms. Además, se utilizaron 80 filtros mel. Estos parámetros fueron escogidos de tal manera que estuvieran en concordancia con los antiguos proyectos que usaron el mismo sistema y que lo adaptaron a posteriori.

Tras el proceso de la extracción del Mell-Espectrograma obtenemos una salida con N*80 teniendo en cuenta el valor de N es igual a la longitud de los audios.

La siguiente parte del esquema con la que nos encontramos es otro diseño particular de la arquitectura del sistema de referencia. La etapa de convolución tiene dos tipos de etapas de convolución, la VGG3 y la VGG4. Se escogió la VGG3 debido a que tras unas pocas pruebas y por recomendaciones del creador del proyecto original. En la primera etapa de pruebas demostró que VGG3 es la que proporciona mejores resultados.

La VGG3 parte de la base y es una variación de una arquitectura más compleja VGG16. La nomenclatura es intuitiva, puesto que la VGG16 consta con 16 capas en total, no obstante la VGG3 al ser una adaptación y solo utilizar 3 pasó a llamarse VGG3.

La CNN o VGG3 usada consiste en 2 capas de convolución, estas cuentan con un pequeño filtro 3x3 con un desplazamiento de 1 píxel en la imagen de entrada durante las operaciones de convolución, combinado con una operación de MaxPooling sobre una ventana de 2x2 píxeles, con 2 píxeles en la imagen de entrada durante las operaciones de convolución que le pertenecen a esta. Al proceso de desplazar los píxeles se conoce como stride. En la siguiente capa el pooling se merma el tamaño de la imagen entrante en 2 en ambas dimensiones. Todas las capas ocultas están equipadas a su salida con la rectificación no lineal, concretamente ReLu.

Acto seguido el proceso anterior se repite hasta en tres ocasiones. El número de canales se multiplica por 2 en cada capa exterior. Finalmente, el número de canales en cada etapa se incrementa a razón de un multiplicador de 2 iniciando en 128, 256 y finalmente de 512 canales.

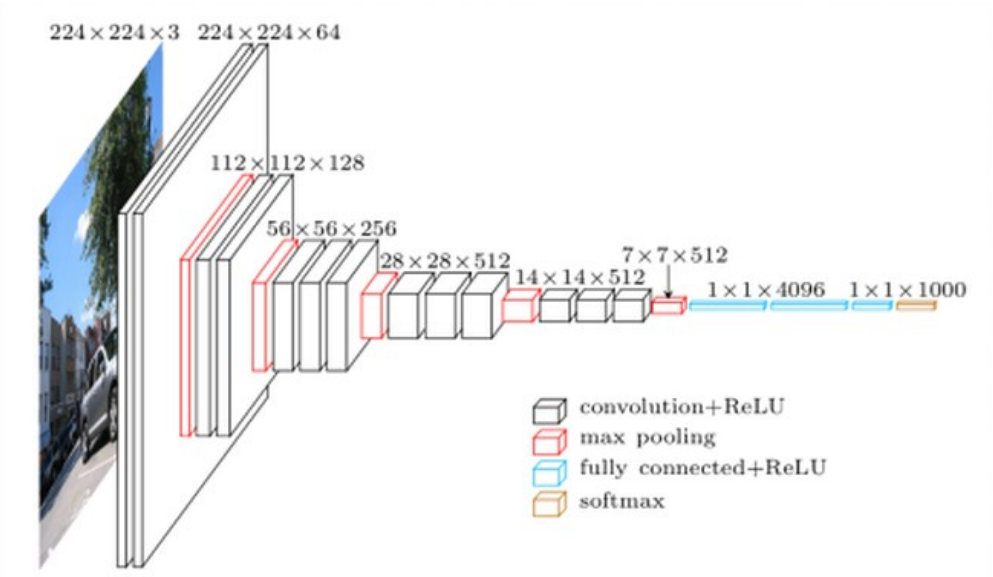

Figura 16 Esquema típico de una VGG16[10].

Antes de avanzar a la etapa donde se procede al fully connected se pasan los datos por una etapa donde se consigue modificar el tensor resultante con la finalidad de que a la salida de este posea unas propiedades tales que le permitan avanzar a la siguiente etapa de manera correcta y redimensionada. Este proceso se conoce como “aplanar” el tensor. Un tensor se define como un objeto matemático o una matriz multidimensional. La definición es válida para el caso concreto de Pythorch.

La etapa de pooling consigue en que tras la etapa anteriormente mencionada y que le precede a esta, la de convolución, aprovecha el tensor entregado para generar matrices que provienen de los distintos canales y generar así una sola matriz resultante. Al conjunto de matrices resultantes se les aplica el cálculo de la media y la varianza en el eje del tiempo aplicado al espectrograma resultante tras la etapa de convolución. El resultado de este procedimiento da lugar a un conjunto de vectores que se agrupan y concatenan entre sí para su posterior traslado a la siguiente etapa del proceso.

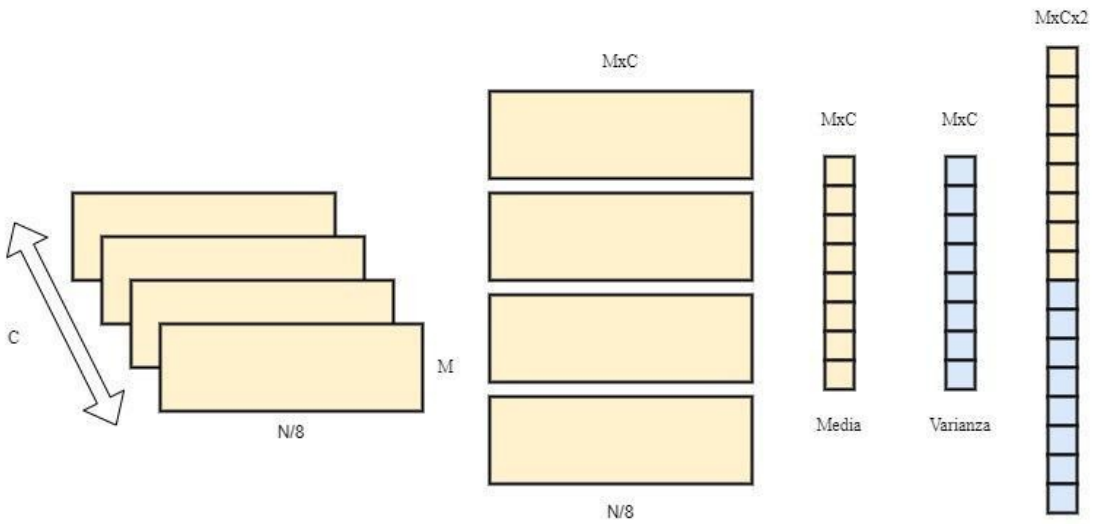

Figura 17 Capa de Statistical Pooling.

En último lugar, nos encontramos la etapa fully connected. Como podemos observar tanto en la Figura 16 como en la Figura 19, la etapa de las fully connected consta de tres etapas. Para el uso de la etapa se decidió no modificar los parámetros que la conforman, siendo así la entrada de dimensión MxCx2 y las capas ocultas de dimensión 400. En el diseño original también tenemos que para cada capa a su salida se aplica la función ReLU.

La principal adaptación llevada a cabo para que el modelo fuera adecuado en el caso de la detección del COVID-19 fue la siguiente. Para que tuviera en cuenta

solamente en la salida las etiquetas de “positivo” y “negativo”, se modifica el clasificador para que adquiriera la capacidad de separar las dos clases, por lo que se modificó la última capa y consiguió una salida de 1x2. Con este cambio el sistema nos entrega la probabilidad de la clase positiva y negativa.

Layer	Dimensiones
Entrada	Nx80
Convolucion+ReLu	128xNx80
Convolucion+ReLu	128xNx80
Max Pooling 2D	256x(N/2)x40
Convolucion+ReLu	256x(N/2)x40
Convolucion+ReLu	256x(N/2)x40
Max Pooling 2D	512x(N/4)x20
Convolucion+ReLu	512x(N/4)x20
Convolucion+ReLu	512x(N/4)x20
Max Pooling 2D	512x(N/8)x20
Aplanamiento	(N/8)x5120
Pooling	10240
FC+ReLu	400
FC+ReLu	400
FC	400
Softmax	1x2

Tabla 2 Dimensiones de la Etapa.

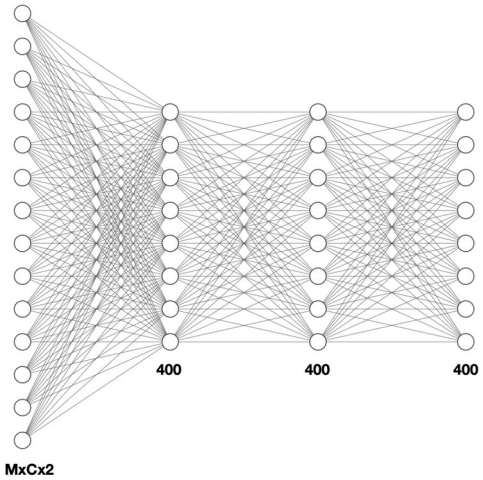

Figura 18 Capa FC.

La CrossEntropyLoss

Este criterio nos entrega la pérdida de entropía cruzada entre la entrada y el target u objetivo. Es útil cuando se entrena un problema de clasificación con C clases, en nuestro caso 2. Tiene la particularidad de que se le puede pasar el argumento opcional weight o peso. Este debe ser un tensor unidimensional que asigne el peso a cada una de las clases. Esto es particularmente útil cuando tienes un conjunto de entrenamiento desequilibrado[13].

Se espera que la entrada contenga puntajes sin procesar y no normalizados para cada clase. La entrada tiene que ser un tensor de tamaño C para una entrada.

$$(minibatch, C) o (minibatch, C, d1, d1, .., dk) \; para \; K \geq 1$$

Los índices de clase en el rango [0, C) donde C es el número de clases.

$$l(x, y) = L = \{l1, \ldots, l_N\}^T, \ln = -w_{yn} \log \frac{\exp(x_{n,yn})}{\sum_{c=1}^{C} \exp(x_{n,c})}$$

Donde “x” es la entrada, “y” el target u objetivo, “*w*” el peso, C el número de clases, N la dimensión del minibatch.

Las probabilidades para cada clase son útiles cuando se requieren etiquetas más allá de una sola clase por minibatch, como etiquetas combinadas, etc. La pérdida en ese caso puede describirse:

$$\ln = -\sum_{c=1}^{C} wc \log \frac{\exp\left(x_{n,c}\right)}{\sum_{i=1}^{C} \exp\left(x_{n,i}\right) y_{n,c}}$$

Como particularidad también fueron implementadas tres soluciones para el pooling la Attention, la Head Attention y la Double Multihead Attention. En los experimentos preliminares, el uso de las dos primeras arrojaba en el entrenamiento y la validación de una gran problemática. Para la decisión del sistema en la fase de entrenamiento como en la de validación, el sistema siempre arrojaba que todos los datos fuesen positivos o negativos, así que se decidió usar la Double Multihead Attention con los hiperparametros que se mencionan en el punto de resultados 4.1, ya que de esta manera la red era capaz de discernir y extraer las características para la distinción de un audio considerado positivo o negativo[11].

Wav2vec2 aplicado al COVID-19

En el transcurso del desarrollo se propuso estudiar la posibilidad de emplear otra implementación para la detección del COVID-19, el sistema que utiliza el Wav2vec2 forma parte de una práctica que se realiza en el Máster que imparte Jose A. R. Fonollosa. Algunas de sus características diferenciales respecto DASV son:

Este sistema consta con un modelo basado en el Wav2Vec2 concretamente "facebook/wav2vec2-base-960h". El modelo base ha sido pre entrenado y ajustado sobre 960 horas, los audios fueron muestreados 16 kHz por lo que los audios que emplearemos serán a la misma frecuencia de muestreo para más información se puede consultar en[12].

Wav2vec se nos presenta con un marco para el aprendizaje autosupervisado de representaciones a partir de datos de audio en bruto. El sistema codifica el audio del habla a través de una red neuronal convolucional multicapa para posteriormente enmascarar tramos de las representaciones del habla resultante. Las representaciones se introducen en una red Transformer o transformadora para construir representaciones contextualizadas.

El modelo construye representaciones de contexto sobre representaciones del habla continuas y la función de atención utilizada, la Self-Attention, captura las dependencias sobre toda la secuencia de representaciones latentes de extremo a extremo. Wav2vec 2.0 bien empleado puede superar a los mejores métodos semi supervisados. Cuando se reduce la cantidad de datos etiquetados a una hora, Wav2vec 2.0 es capaz de tener el mismo resultado o mejor que sistemas con 100 h de grabaciones sin etiquetar. El sistema en concreto de la práctica usa para la validación el cálculo de la AUC o Área Bajo la Curva en cada época y la roc_auc_ci, AUC con un intervalo de confianza del 95%.

En el entrenamiento para calcular la loss o función de pérdida utiliza a diferencia de DASV el criterio llamado BCEWithLogitsLoss. La BCEWithLogitsLoss es una función de pérdida combina una capa sigmoidea y la BCELoss en una sola clase. Esta versión es más estable numéricamente que usar un sigmoide simple seguido de un BCELoss, ya que al combinar las operaciones en una sola capa[14].

A la salida del sistema tenemos una sigmoide que nos entrega el resultado de las predicciones en un archivo csv. El archivo resultante se readapta para que sea procesado por la función de test programado aparte y obtener así un resultado con el cual poder comparar los dos sistemas.

3.4. Data Augmenation aplicada a la base de datos CCS

La elección escogida en el proyecto con tal de mejorar y estudiar los resultados obtenidos fue el uso del aumento de datos o DA (Data Augmentation). Esta técnica explicada en el punto 2.3 está demostrado que partiendo de un sistema de referencia y bien utilizado puede aprovecharse para mejorar el modelo, haciéndolo mucho más robusto, además de prevenir el over-fitting.

Concretamente, la técnica específica elegida fue el empleo del DA con el empleo de ruido para aumentar la base de datos que pertenece a la fase del entrenamiento.

No obstante, en [4] una de las conclusiones obtenidas nos dice que mediante el uso de ruido para incrementar la base de datos del train puede inducir si no se gestiona de manera correcta a la mala interpretación por parte del sistema en tareas como el reconocimiento del interlocutor. Esto es debido en que para ese caso, si se introduce ruido de manera descuidada, el sistema termina por asimilar características más enfocadas en la distinción del ruido que en las propias de la voz.

De esta manera se procedió al empleo de distintos ruidos, dos de ellos utilizados para el reconocimiento de emociones en el habla, en el caso de encontrarse en un entorno de circulación con un automóvil y en el caso particular de este proyecto la utilización de un ruido genérico, concretamente se usó ruido blanco.

El ruido sobre el que se hace más hincapié en el uso de las pruebas de test es el ruido blanco dadas sus características. En el proyecto se usó un ruido blanco generado por el programa de Audacity con las características que tuviera una amplitud de 0.5 y una frecuencia de muestreo de 16 kHz, este último parámetro está en concordancia con el uso de la frecuencia de muestreo empleada en la base de datos CCS.

Tras determinar las características del ruido utilizado también se usaron los audios del interior de un coche y el de tráfico, estos audios los usamos de antiguos experimentos y tienen una frecuencia de muestreo también de 16 kHz.

Para la realización del DA se parte de 286 audios en el subconjunto de la base de datos, la de entrenamiento, y se quiere hacer que crezca como mínimo en un factor de 5.

Se tuvo en cuenta la propuesta de [4] sobre cuáles deberían de ser unos buenos valores de ruido tales que dados su potencia no solaparse en exceso la señal original, teniendo en cuenta este criterio se decidió utilizar para el ruido blanco los rangos $[15,25] \in N$ en dBs. Mientras que para el caso de los otros dos ruidos se propuso un intervalo mayor siendo este $[10,25] \in N$.

El procedimiento para añadir el ruido al audio original se genera de tal manera que en el caso de que la diferencia entre el nivel de potencia de la señal y el del ruido entregado sea distinta a los valores de SNR que le pasamos, se corrige el desajuste gestionando la ganancia aplicada al ruido para que una vez adaptado la señal se puedan sumar las dos señales sin que se enmascaran los audios originales y, por lo tanto, se pierda información.

El resultado tras la realización de este DA en el caso del uso de tráfico e interior de coche resulta de, 9438 audios, partiendo de 286 del subconjunto de entrenamiento, siendo un aumento por un factor de 33. En el uso de ruido blanco se consiguen 3432, siendo un aumento por un factor de 12 respecto a la original.

Dada la partición desigual de la base de datos de train siendo los positivos de 71 frente a los negativos 215, es decir, un 24.8% frente a un 75.1%. Se parte en ese momento con una base de datos desbalanceada igual que en un principio. Para ello se desarrolló un script donde se tenía en cuenta el desbalance. Se llevó a cabo las siguientes consideraciones:

- Se requiere una base de datos en entrenamiento con un 50/50 en cuanto a la relación de positivos enfrente a negativos.
- Se aprovecha la gran mayoría de audios resultantes de audios positivos, ya que tenemos muchos menos.
- Para la elección de los audios negativos se hizo de manera aleatoria y deberá de cumplir siempre la relación 50/50.

Se generó un archivo de etiquetas donde se tienen en cuenta los siguientes requisitos tras el resultado de la salida del script en las siguientes particiones equilibradas.

Fuente	Positivos	Negativos	Total
Referencia	71	215	286
Ruido Blanco	780	780	1560
Ruido Coches	2342	2342	4684

Tabla 3 Aumento final de audios.

Si hacemos los cálculos pertinentes se puede observar que el tamaño multiplicador no es un número entero. Esto es debido a que en el momento de seleccionar el valor aleatorio la función de python utilizada consultaba dos booleanos para comprobar que las listas de positivos y negativos fueran iguales. El problema reside en que en el momento de usar la función aleatoria cada vez le cuesta encontrar más un valor que dentro de la lista total sea un valor positivo que no haya salido anteriormente, con lo que el tiempo de ejecución se alarga cada vez más dependiendo del tamaño de la base de datos a equilibrar. Por ese motivo se seleccionó un valor próximo al número de máximo de positivos sin que comprometa en exceso el perder audios para la implementación del DA. En el caso del ruido blanco se seleccionó el valor de 780 y en el caso del Ruido Coches 2342. El resultado fue la reducción a más de la mitad del tiempo de ejecución del script para posibles usos más adelante en el caso de que alguien retome el proyecto una vez finalizada esta contribución.

4. Resultados

4.1. Resultados de DASV adaptado al COVID-19

Para la realización de los experimentos se utilizaron los siguientes programas y software:

- MobaXterm: Editor de texto para la conexión vía ssh.
- OpenVPN: Programa VPN recomendado por el centro para la conexión con el servidor CALCULA.
- Visual Studio Code: Editor para la implementación de scripts.
- Audacity: Programa utilizado para la creación de ruido blanco y la adaptación de los audios al formato requerido.
- PowerToys: Programa utilizado para la gestión de nombres de archivos y carpetas.
- Windows 10: Sistema operativo sobre el que se trabajó localmente.
- Ubuntu v18: Versión de Ubuntu utilizada en el servidor CALCULA.
- Python v3: Librería de Python requerida para el desarrollo del sistema de referencia.
- byobu: Herramienta integrada en el servidor CALCULA para trabajar en paralelo en el momento de realizar experimentos.

Para su configuración, las pruebas realizadas con el sistema DASV tienen el siguiente procedimiento.

Primeramente, la base de datos sigue la partición comentada en el capítulo 3.1. Conteniendo para la fase de entrenamiento 286 audios, para la de validación 231 y para la fase última de test 208.

Tras organizar los datos se procede a la extracción de sus características para su posterior entrega al sistema. Una vez efectuada la extracción de características se pasan los datos a la fase de entrenamiento donde se procesan y se calcula su precisión y la Fscore.

En cada época se procede a una validación y se guarda la mejor configuración resultante, esta se irá actualizando a medida que se vayan consiguiendo mejores tandas de validación con los parámetros seleccionados. Como el proceso tendrá un límite que cueste superar, se propone el uso de un medidor de épocas donde no se mejora el resultado. Este medidor se situó en 50 épocas dada la rapidez

del sistema en realizar una época. Como resultado de esta práctica se obtiene un resultado donde se asume que el sistema ha convergido hacia una solución válida y que puede pasar a la etapa del test.

Los resultados del entrenamiento de validación se guardan en dos archivos que gestionan los valores de la red que luego pasarán al script que evalúa los resultados. El sistema que evalúa el mejor resultado que consigue en la etapa de validación y nos entrega las predicciones del sistema, los audios que conforman la partición del test son los elegidos para este proceso. Acto seguido entregamos las predicciones por la función que nos calcula la ROC(Característica Operativa del Receptor) donde se extrae el valor de la AUC(Área Debajo de la Curva) [15] que se utiliza para valorar el resultado y poder comparar con otros experimentos con diferentes características.

Algunos parámetros del sistema que se usaron que se pueden destacar son los siguientes:

- Front-end: VGG3
- Optimizador: 'Adam'
- Metrica: Fscore, tiene en cuenta el compromiso entre la precisión y la exhaustividad.
- Learning rate: Hiperparámetro que gestiona la velocidad en la que se realiza el aprendizaje de la propia red. Cuanto más cercano a 0 el sistema converge de manera más lenta o quedarse en un mínimo local, a mayor valor el sistema puede tener un comportamiento errático y nunca alcanzar un valor deseable que converja. El valor escogido fue el 0.0001 típico para casos de aprendizaje profundo.
- Tamaño del batch: 64.
- Weigh-decay:Parámetro asociado al overfitting. Se escogió el valor 0.001.
- Parámetros del DMHA
- Cabezas: 32.
- Máscara de probabilidad: La probabilidad que dada una cabeza tenga una probabilidad de otorgar a la salida un peso de 0. Este parámetro se dejó en un valor de 30%.

Con el fin de comprobar los resultados del sistema de referencia adaptado al COVID-19 se usaron las configuraciones de referencia y referencia con pesos. Pesos hace referencia al uso de la Cross Entropy Loss con una ponderación de los pesos que tiene en cuenta el desbalance. A la función se le pasan la

estadística de la base de datos de entrenamiento, siendo así el porcentaje de positivos y negativos.

Para la interpretación de los resultados se entiende que para la AUC los valores van comprendidos entre 0 y 1. Siendo 0 el peor caso, 0.5 el mismo caso que tirar una moneda equiprobable y 1 el caso perfecto.

Una vez tenidos en consideración los puntos anteriores se obtiene el siguiente resultado de referencias sobre el que calcularemos el porcentaje delta.

$$\Delta\% = \frac{AUCt - AUCb}{AUCb}$$

Siendo AUCt la AUC del experimento y AUCb la AUC del sistema de referencia.

Sistema	AUC / $\Delta\%$
Referencia	0.689 / -

Tabla 4 Resultado AUC del sistema de referencia.

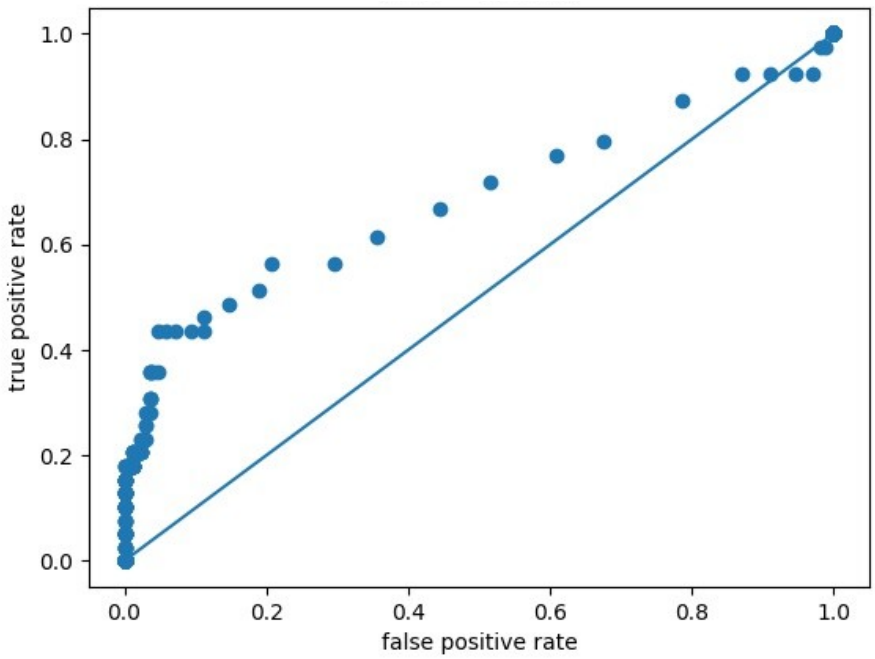

Figura 19 ROC del sistema de referencia.

Para el resultado del test con la configuración de referencia se considera que la AUC de partida pertenece a un test mejorable. Posteriormente, se realizaron pruebas teniendo en cuenta la función Cross Entropy Loss. En lo que respecta al uso de los pesos en la Cross Entropy Loss obtenemos el siguiente resultados, que se comparará con el sistema de referencia a través del porcentaje delta.

Sistema	AUC /Δ%
Referencia + Pesos	0.667 / -3.19%

Tabla 5 Resultado del uso de pesos en el sistema de referencia.

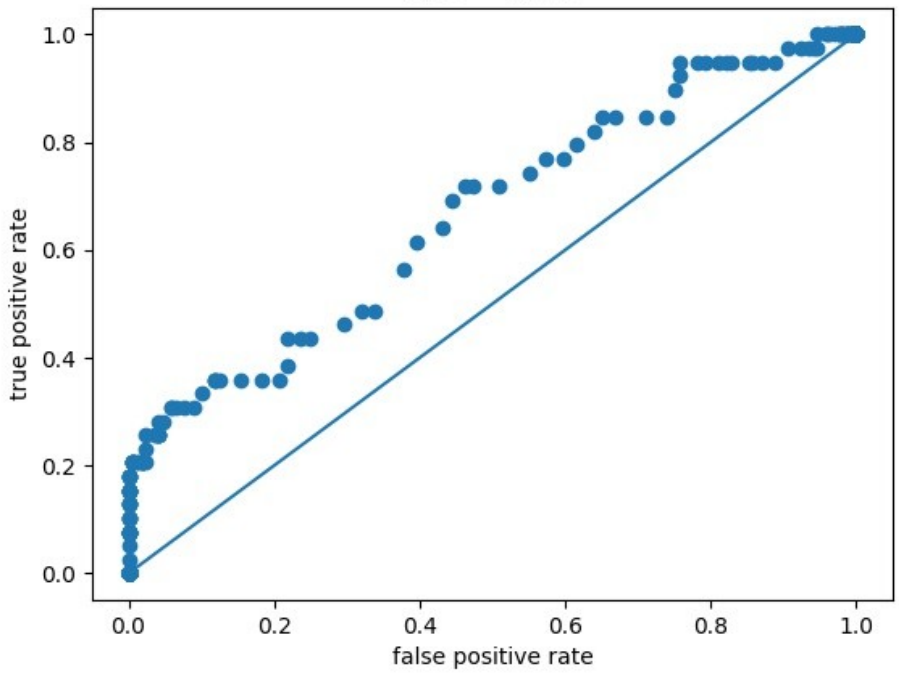

Figura 20 ROC con uso de pesos en el sistema de referencia.

Adicionalmente, se probó el sistema Wav2vec2 obteniendo una AUC de 0.609. Debido a que se considera que el resultado obtenido no era relevante con características de un mal test se optó por proseguir el análisis con el sistema DASV con el fin de lograr un resultado relevante en el proyecto.

4.2. Observaciones y corrección de audios del test

El resultado obtenido con el sistema de referencia y referencia con pesos lejos de mejorar, empeora por muy poco. Se presenta la hipótesis de una problemática en el sistema de test. El hecho del uso de la función de pérdidas debe siempre en el caso que nos ocupa mejorar el resultado debido a que en el momento del entrenamiento se tiene en cuenta el desbalance de las dos clases a entrenar. Pese a que este balance se traslada a las siguientes fases del experimento, debe mejorar y se abre una búsqueda del causante.

Se recurrió al análisis individual de las grabaciones en la partición del test y se observó, no obstante, que algunos audios no estaban bien acondicionados. El mal acondicionamiento de estos audios, concretamente los clasificados como positivos, se tuvo en cuenta para la elaboración de un ajuste para verificar la hipótesis. Se observa una gran cantidad de silencios en los cuales los pacientes no interactúan con la grabación, el ruido excesivo del entorno y la interferencia de otros dispositivos como el audio de un televisor.

Los cambios realizados en los audios del test fueron los siguientes:

- Eliminación de silencios en los que los usuarios no interactúan con la grabación.
- En diferentes grabaciones el usuario hablaba con otra persona del entorno y se borraron dichas conversaciones.
- Si se escuchaba un ruido muy estridente se procedió a la eliminación de dicho sonido mediante un filtro de ruido utilizando el programa Audacity.

Ejemplo test_036:

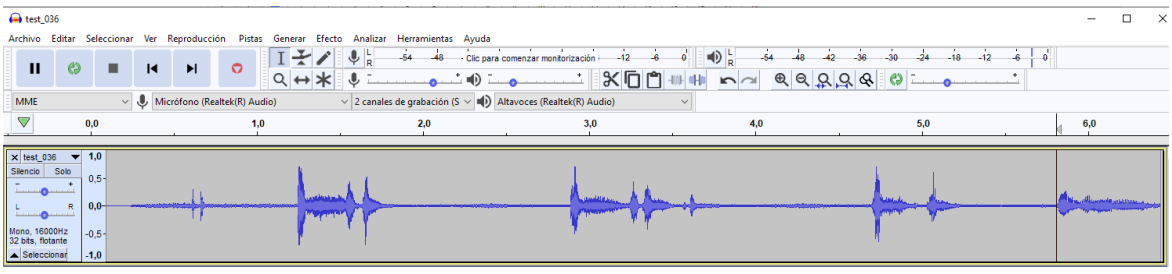

Figura 21 Audio de test_036.

El audio test_036 está etiquetado como positivo, no obstante se observa que en el último tramo de la grabación la forma de onda del audio no se corresponde con las anteriores. Se procede a la escucha del audio y se comprueba que la última

onda forma parte de una risa que graba el paciente. Al instante del tiempo donde se encuentra la marca se procede al corte del audio y su eliminación, dando como resultado un audio más corto, por lo tanto, modificado y no igual al original. Esto provoca que en el momento de hacer el test no se le pasarán los mismos audios o datos al código que evalúa los resultados, arrojando entonces un valor distinto en la predicción. Este proceso se tiene en cuenta en todo momento de la ejecución del test mediante el uso de un archivo que contiene la ubicación de los nuevos audios resultantes.

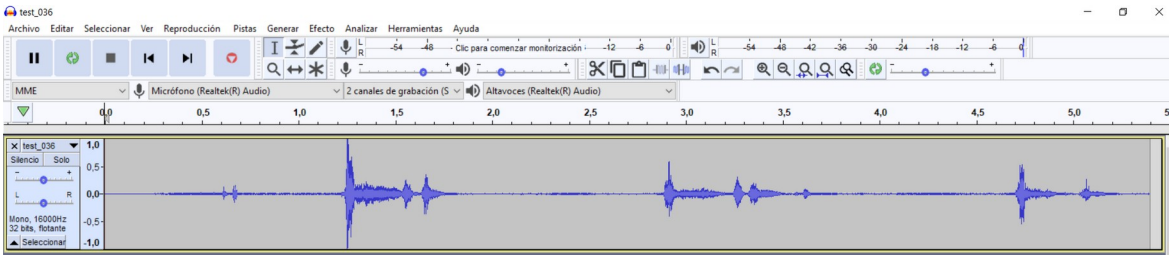

Figura 22 Audio de test_036 corregido.

En el mismo ejemplo también se aplicó el procedimiento de reducción de ruido. Se puede observar que en el tramo del audio desde el inicio hasta el segundo 1.25 existe ruido, por lo que se procede a aplicar un filtro de ruido que nos proporciona Audacity.

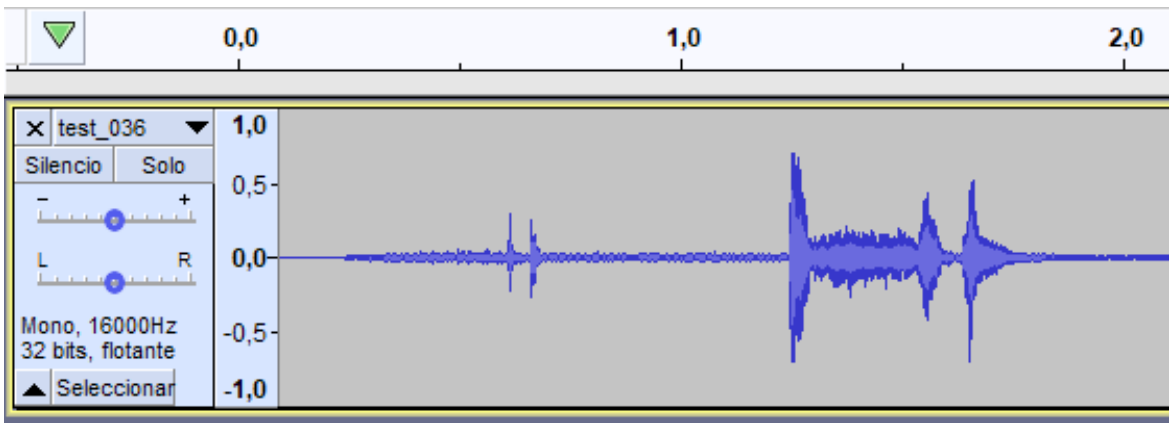

Figura 23 Ruido en audio de test_036.

Tras el filtro en todo el audio se obtiene el siguiente resultado siendo visible la disminución del ruido de fondo.

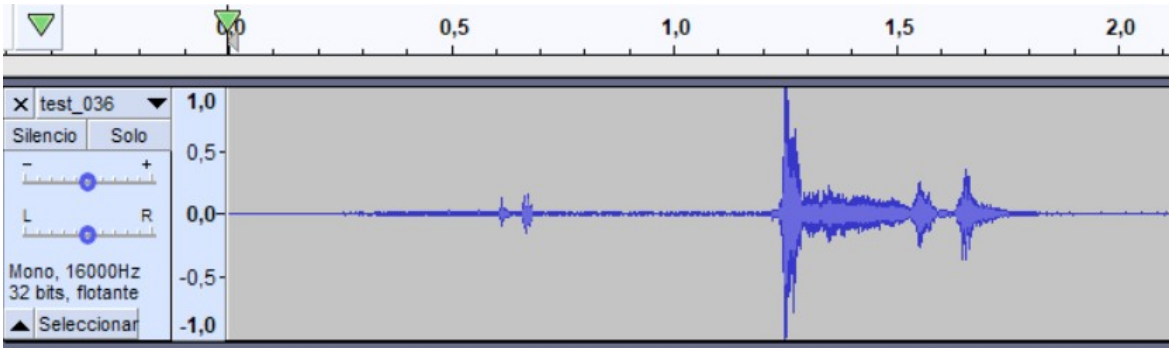

Figura 24 Ruido en audio de test_036 corregido.

El procedimiento anterior mencionado se aplica de la misma manera en 35 audios clasificados como positivos. Previamente y para estudiar la hipótesis se hicieron dos conjuntos de test. El primero es el considerado como referencia sin los audios corregidos y el segundo llamado test corregido.

Teniendo en cuenta el anterior punto se consideró volver a realizar los test con los mismos archivos resultantes de la validación, obteniendo entonces resultados en el test y curvas distintas a las primeramente obtenidas, se aprovecha entonces el hecho de valorar que la fase de test tiene en cuenta las modificaciones que se consideran pertinentes y anteriormente explicadas.

Una vez efectuados los cambios se volvieron a repetir los test y se obtienen los siguientes resultados para referencia y referencia con pesos.

Sistema sin Test Corregido / Δ%		**Sistema con Test Corregido / Δ%**	
Referencia	Referencia + Pesos	Referencia	Referencia + Pesos
0.689 / -	0.667 / -3.19%	0.830 / 20.4%	0.900 / 30.6%

Tabla 6 Resultado con la corrección en el test.

Tras la corrección en la partición de test se observa que a priori el sistema de test tiene una salida más coherente, no obstante se deben mirar las curvas ROC para determinar si existe una mejora notable.

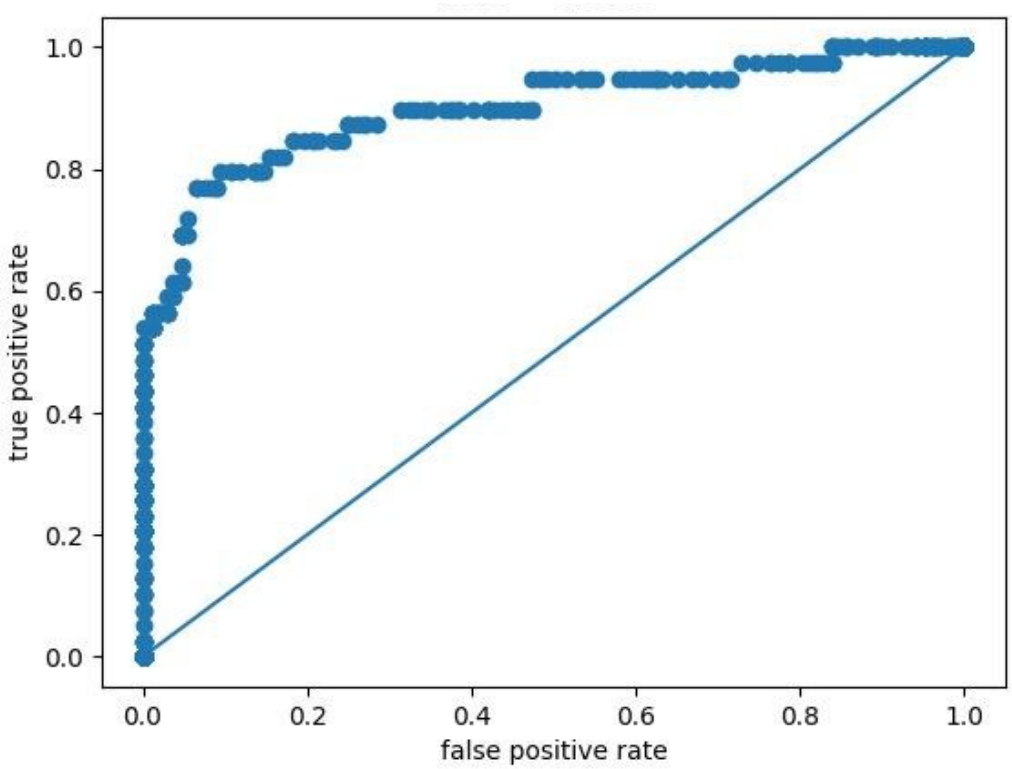

Figura 25 ROC del sistema usando pesos y el test corregido.

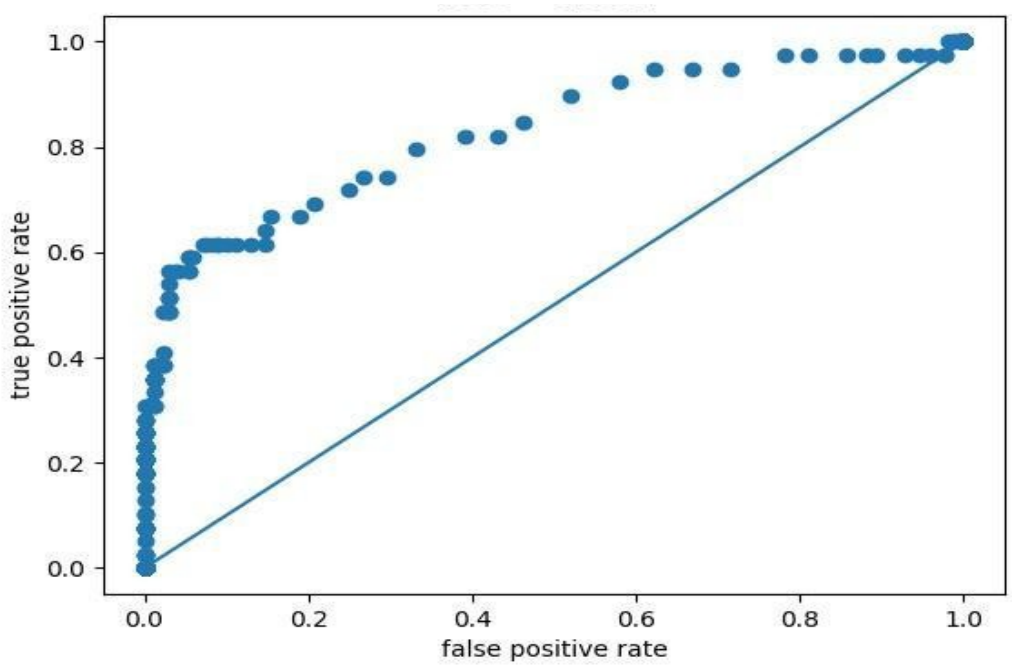

Figura 26 ROC del sistema de referencia con test corregido.

Tras considerar la hipótesis, se demuestra que el test no estaba acondicionado para los test y que los resultados obtenidos con el test corregido arrojan valores esperables y que entran en el marco de un test muy bueno para el caso de uso de pesos.

4.3. Uso de Data Augmentation

Para la realización de los experimentos que se muestran fueron utilizados los mismos programas y software que en el caso del sistema DASV con la variante claro está del uso de los audios resultantes del DA explicado en el punto 3.4.

Con el fin de comprobar los resultados del DASV al COVID-19 usando DA se usaron las configuraciones del sistema de referencia con DA perteneciente a ruido blanco y referencia con DA perteneciente a ruido de coches. Ruido de Coches está formado por ruido procedente de los audios de tráfico y de interior_coche, además nunca se usa más de un ruido en cada audio resultante del DA.

Siguiendo el mismo razonamiento que en el punto 4.2 se tiene en cuenta el factor de los problemas que tiene el conjunto del set de test y se aplica la misma metodología en el momento de realizar los test. El resultado de dichos test se encuentra en la siguiente tabla que refleja la mejora respecto al sistema de referencia sin el trato de ningún tipo de corrección de audios en el test.

Sistema	Sin Test Corregido / Δ%	Con Test Corregido / Δ%
Referencia	0.689 / -	0.830 / 20.4%
Ruido Blanco	0.713 / 3.48%	0.849 / 23.2%
Ruido de Coches	0.612 / -11.1%	0.785 / 13.9%

Tabla 7 Uso de DA con y sin test corregido.

Tras la corrección en la partición de test se observa que a priori el sistema de test tiene una salida más coherente, no obstante se deben mirar las curvas ROC para determinar si existe una mejora notable. A priori también tenemos un resultado malo si tenemos en cuenta que el uso de ruido de coches no mejora el sistema base y, por lo tanto, a partir de la curva ROC se determina si el funcionamiento es el adecuado.

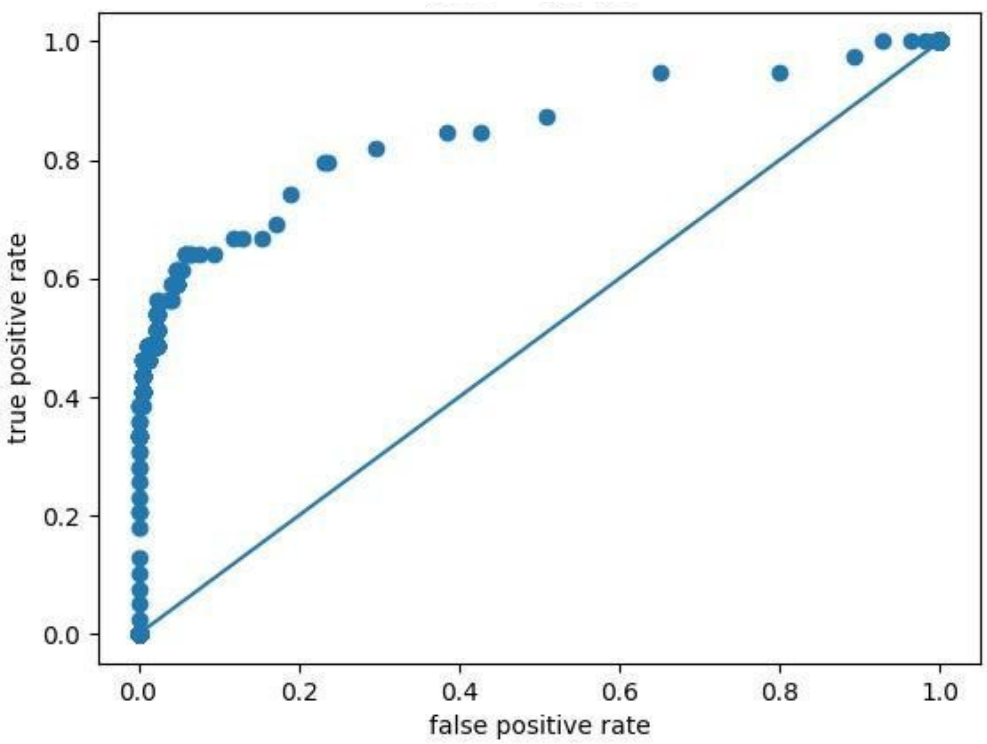

Figura 27: Curva ROC del DA con ruido blanco y test corregido.

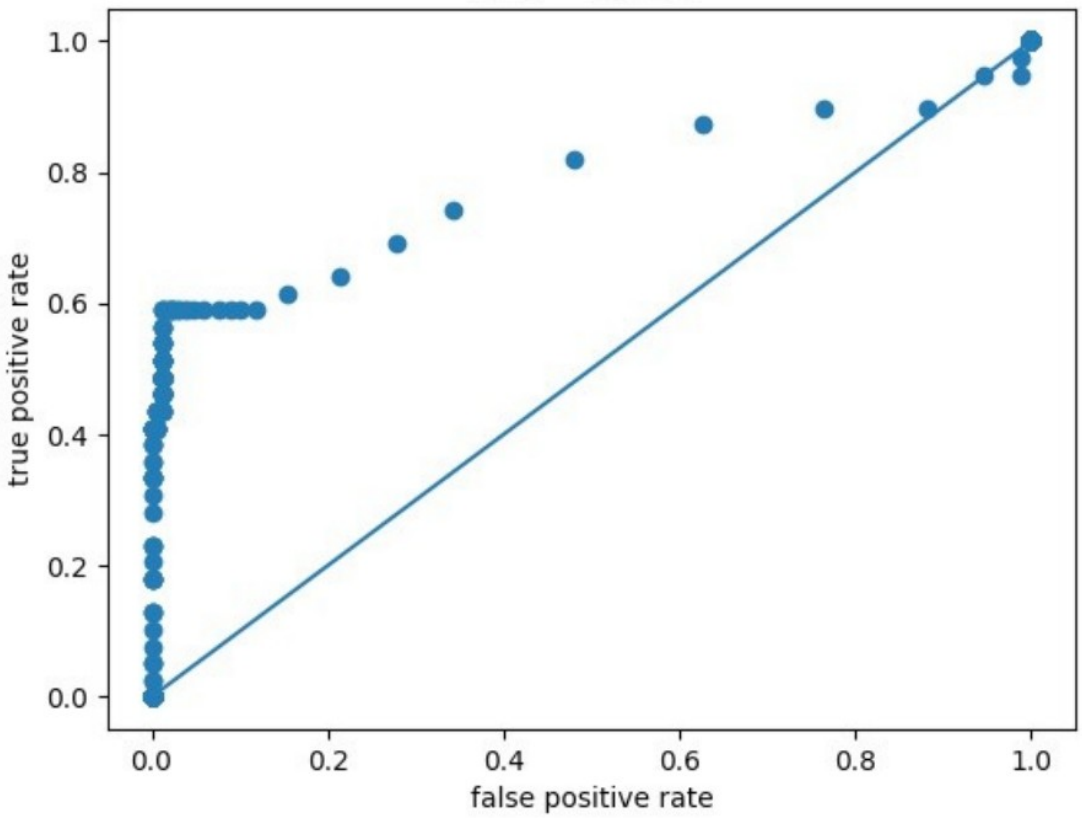

Figura 28: Curva ROC del DA con ruido de coches y test corregido.

Es evidente que para el caso del DA tenemos un impacto que corrobora la teoría de que el uso de DA mejora el modelo y a su vez observamos que los audios corregidos si tienen impacto también en los experimentos con DA, dando a entender y reforzando que la hipótesis inicialmente planteada fue bien planteada y resuelta. Es evidente también que el uso del DA mejora los resultados del sistema de referencia para el caso de ruido blanco y, por lo tanto, una buena manera de intentar solventar la descompensación de una base de datos en el caso de la detección de COVID-19 a partir de audios de tos. No obstante, el sistema que usa el ruido de coches da un peor resultado que el sistema de referencia. Lejos de ser una buena noticia, se observa que la curva ROC para el caso de ruido de coches no arroja una curva que sea con unas características tan buenas si la comparamos con su competidora de ruido blanco. En el apartado de discusión de resultados se analizará el resultado y elaborará una conclusión para dar explicación al suceso.

4.4. Discusión de resultados

En primera instancia se observa que partimos de un sistema que obtiene para la base de datos CCS un resultado en el test con el sistema de referencia con una AUC de 0.689 e incrementándose hasta 0.830, es decir, una diferencia respecto a no usar el test corregido del 20.4%.

Se observa que siempre hay un incremento de la AUC en todas las pruebas realizadas con los audios corregidos, verificando la hipótesis del mal acondicionamiento de la base de datos en test.

Tras los experimentos realizados se observa que si se usa como referencia el sistema de referencia sin la corrección en la partición de audios de test, el mejor resultado con el DA se obtiene del ruido blanco con un incremento del porcentaje delta del 23.2% frente al 13.9% del ruido con coches. Una diferencia cerca del 10% y que es suficientemente grande como para tener en cuenta el uso de un ruido blanco frente al procedente de los que proceden de una grabación de ruidos de coches, es más adecuado para el uso del DA. El uso del ruido de coches también es peor en las mismas condiciones que el sistema de referencia, por lo tanto, no es recomendable su uso en un proyecto con estas condiciones de partida, teniendo en cuenta que el sistema de referencia alcanza una mejora del 20.4% frente al 13.9%.

No obstante, el uso de la Cross Entropy Loss arroja un resultado prometedor con un incremento del 30.6% respecto al sistema de referencia, se esperaba un incremento, pero es ligeramente mejor que el resultado del ruido blanco, dando a entender que el uso de una función de perdidas arroja mejores resultados que el uso del DA en su vertiente que añade ruido, en el caso que nos ocupa ruido blanco.

Si se cambia el valor de referencia por el sistema de referencia con la corrección en la partición de audios de test, es decir, 20.4%, se obtiene otro punto de vista

distinto que permite evaluar los sistemas con a corrección en la partición de audios de test.

La mejora porcentual pasa a ser para el caso de DA con ruido blanco de un 2.80%. No obstante, siempre se puede hablar de una mejora y que el uso del ruido blanco consiguió mejorar el modelo. Para el caso del ruido procedente de coches se observa que se tiene un empeoramiento del 6.25% respecto al sistema de referencia y, por otra parte, un descenso del 9.30% respecto al competidor que usa ruido blanco. En el caso del uso de pesos provenientes del uso de la Cross Entropy Loss se obtiene una mejora del 10.2% siendo una diferencia notable y a tener en cuenta.

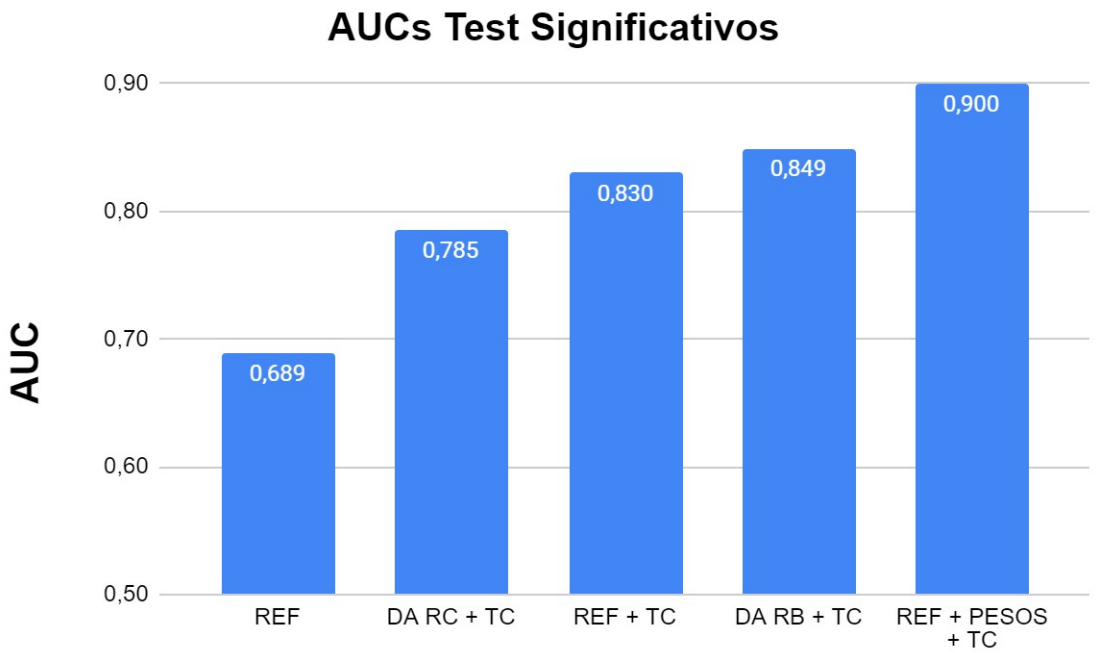

Figura 29 AUCs de los test más significativos.

Tras valorar las AUC se valora también la ROC de los sistemas más significativos. En la ROC existe una interpretación que otorga una amalgama de valores del threshold que pueden ser según el criterio que se use más idóneos que otros. Para poder comparar las ROC se deben de representar en un mismo gráfico para observar que curva otorga unas mayores prestaciones. Así pues, si se observan las curvas se observa que en concordancia con los parámetros de las AUCs también el sistema de los pesos con los test corregidos otorga una curva más cercana a la idónea, seguida por la que pertenece al test con el DA con ruido blanco y seguida de cerca por la de referencia con la partición de test corregida. En la figura también se puede observar que en el caso del DA con ruidos de coche tiene una curva más plana. También se representa y se observa que el sistema de referencia sin modificaciones en la partición del test es el que

otorga peor curva en los sistemas de más interés. Los resultados entonces se valoran de la misma manera y con la misma clasificación que para el caso de las AUCs.

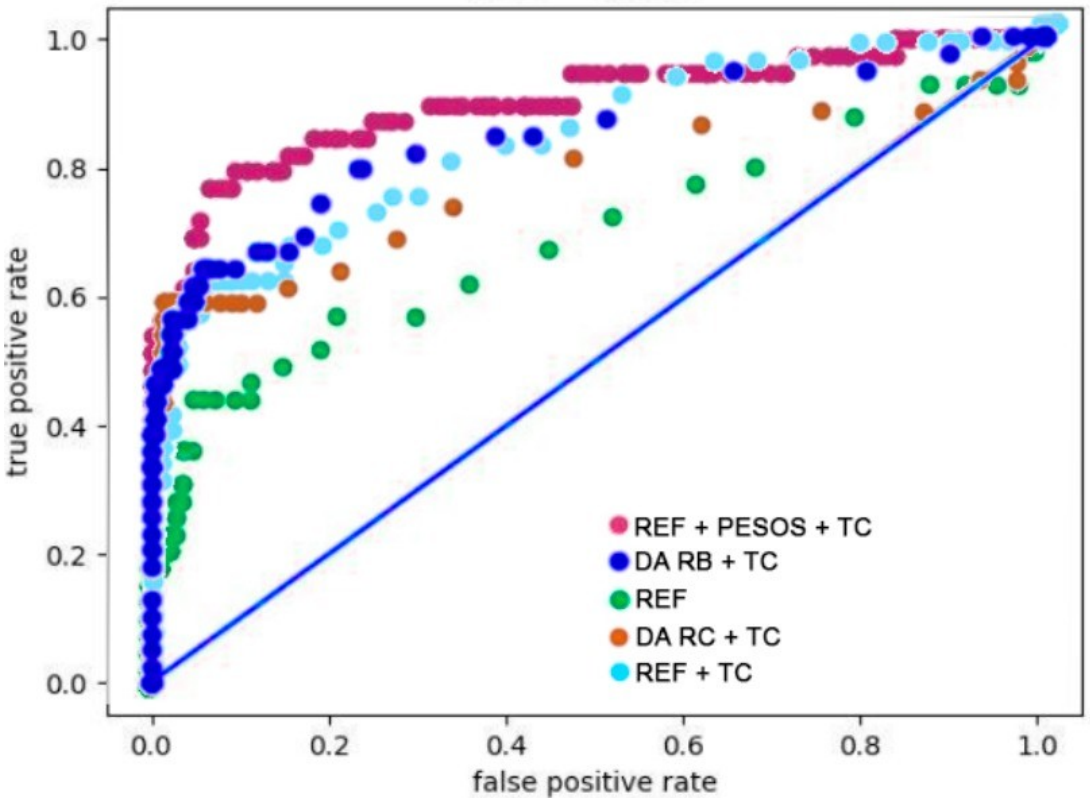

Figura 30 Curvas ROC de los sistemas más representativos.

5. Presupuesto

El proyecto tiene una finalidad clara de investigación y no tiene un propósito final de desarrollar un producto comercial. Como consecuencia, la manera de plantear los costes se tendrán en cuenta las siguientes premisas:

- El proyecto está enmarcado en una asignatura de 18 créditos ECTS, cada crédito equivale a unas 30 h de dedicación por parte del estudiante. Se pidió una extensión del plazo de entrega y, por lo tanto, el recuento de horas será el doble.
- Se considera que el sueldo por hora de un trabajador junior acostumbra a estar sobre los 12 €, mientras que un senior está sobre los 25 €.

El coste por parte de los técnicos sería de:

$$Ct = \left(18\,ETCS * \frac{30\,h}{1\,ETCS} * \frac{12\,€}{1\,h}\right) * 2 + \frac{25\,€}{1\,h} * 30\,h = 13710\,€$$

Los experimentos para ser ejecutados funcionaban sobre la GPU del servidor CALCULA proporcionado por la universidad. Se considera que un servicio en la nube con la misma capacidad de computación suele rondar los 14.76 €/mes, se estuvieron haciendo pruebas durante siete meses. Tras estas suposiciones se considera que el coste por hardware sería de:

$$Ch = 7\,meses * 14.76\,€/mes = 103.32\,€$$

Podemos concluir entonces que el coste total es de:

$$Ctotal = Ct + Ch = 13813.32\,€$$

6. Conclusiones y futuro desarrollo

Para elaborar las conclusiones se tienen en cuenta los objetivos marcados en el punto 1.2. En primer lugar, el conseguir readaptar el sistema entregado para la detección del COVID-19 se puede considerar que el objetivo fue alcanzado gracias a tener un margen de mejora y los resultados obtenidos. El segundo punto se puede discutir sobre si su cumplimiento o no pudo ser alcanzado. Si tenemos en cuenta si se utilizó más de un sistema, podemos concluir que es cierto. No obstante, los resultados arrojados inicialmente provocaron un descarte en el momento de valorar la posible comparación entre los dos sistemas DASV y Wav2vec2. El tercer punto hace referencia al uso del DA y considerando que en los resultados se valora como que se consigue tener una mejora respecto al sistema de referencia, se da por cumplido el requisito.

Si pasamos a la parte de las conclusiones sobre los diferentes experimentos y conclusiones obtenidas. Tenemos que el uso del aumento de los datos gestionado de manera correcta puede, en caso de necesidad construir un modelo mejor y más robusto, poder generar una solución que lejos de estar al mismo nivel que la obtención de nuevos datos puede mediante código solucionar un problema que existe en muchos otros campos que se tratan en el mundo del aprendizaje profundo. Si nos basamos en los resultados observamos una mejora del 2.8% si tenemos en cuenta el uso del test corregido entre el sistema de referencia con el test corregido y el uso de ruido blanco más el test corregido. En el caso de comparar el sistema de referencia la mejora pasa a ser del 23.2% teniendo en cuenta todos los cambios realizados y con el trabajo de investigación que lo sustenta. Se puede concluir que el resultado es remarcable y que debe tenerse en consideración en futuros proyectos que se basen en esta memoria. No obstante, para el sistema que usa el ruido de coches la mejora no está presente, dando a entender que dependiendo de la problemática no se pueden usar todo tipo de ruidos para realizar el DA.

Si nos alejamos de la temática principal del DA se observa que en un principió el uso de la Cross Entropy Loss no era prometedor hasta el uso de los pesos que tienen en cuenta el desbalance en la etapa de entrenamiento. Sin duda mejoran el resultado del uso de ruido blanco con un resultado final del 30.6% de mejora respecto al sistema de referencia, teniendo en cuenta la corrección de la partición de los audios de test.

En el momento de valorar los objetivos inicialmente planteados se tiene que tener en cuenta que el proyecto era muy ambicioso y que por desgracia, como consta en el Critical Review surgieron distintas incidencias. Entre ellas están que el estudio de la base de datos se dilató mucho en el tiempo por el gran desbalance y, por otra parte, es pequeña, por lo que se intentó por mucho tiempo intentar balancear con diferentes funciones de pérdida para calcular la loss y tuviese en cuenta el desbalance, pero en todos los casos los resultados no fueron

satisfactorios menos el resultado para la Cross Entropy Loss. También es cierto que la adaptación del código y el uso de él llevo mucho más tiempo del esperado y, por lo tanto, el proyecto no se pudo desarrollar mucho más.

Es evidente que el proyecto tiene margen de mejora en el proceso de elaborar el proyecto y en la tarea de investigación se vieron diferentes puntos donde se podría indagar y conseguir una mejora del sistema.

Los puntos por los que propongo avanzar pasan primeramente por la construcción de una base de datos más extensa y equilibrada que la original usada en el proyecto, la CCS. Aumentar siempre la base de datos con audios originales bien etiquetados ayuda sin duda a que el sistema sea mucho más robusto.

Por otro lado, se observó que en el empleo de bases de datos mucho más grandes se suele optar por la VGG19 o VGG16 y sería un buen campo de investigación el poder comparar las dos CNN con la VGG3.

Otro punto muy interesante que llama la atención es el uso de sistemas híbridos que por su complejidad no fueron implementados ni propuestos en la solución del proyecto. Estos sistemas son siempre interesantes, ya que para el caso de la detección del covid es muy llamativo el hecho de a través de radiografías poder detectar el COVID-19. Se encontró mucha información al respecto que se consideró descartarla, puesto que no era el tema que seguía el proyecto.

Finalmente, sugerir que si se opta por el DA se tengan en cuenta las diferentes maneras de introducir variaciones en los audios comentadas en el punto 2.3 de la memoria, que pueden ayudar a generar muchos más audios y más diferenciados unos de los otros.

7. Bibliografía

[1]GEORGEVICI, Adrian Iustin a Marius TERBLANCHE. Neural networks and deep learning: a brief introduction. Intensive care medicine. Berlin/Heidelberg: Springer Berlin Heidelberg, 2019, roč. 45, č. 5, s. 712–714. ISSN 0342-4642. DOI: 10.1007/s00134-019-05537-w

[2]KAMBLE, Madhu R et al. PANACEA cough sound-based diagnosis of COVID-19 for the DiCOVA 2021 Challenge. . 2021

[3]ALI BOU NASSIF et al. COVID-19 Detection Systems Using Deep-Learning Algorithms Based on Speech and Image Data. Mathematics (Basel). Basel: MDPI AG, 2022, roč. 10, č. 4, s. 564–. ISSN 2227-7390. DOI: 10.3390/math10040564

[4]Ritter, Fabian. (2016). Uso de SVM para un sistema de reconocimiento de género a través de señales de voz. 10.13140/RG.2.2.26557.26085.

[5]Abou-Abbas, Lina & Tadj, Chakib & Gargour, C. & Montazeri, Leila. (2016). Expiratory and Inspiratory Cries Detection Using Different Signals' Decomposition Techniques. Journal of voice : official journal of the Voice Foundation. 30. 10.1016/j.jvoice.2016.05.015.

[5]D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5329–5333, 2018.

[6]G. (2021, 29 agosto). *Clasificación de imágenes de TensorFlow: CNN (Red NeuralConvolucional)*.Guru99.https://guru99.es/convnet-tensorflow-image-classification/

[7]Ramírez Sánchez, José & Bereau, Montalvo & Lara, José. (2019). A Survey of the Effects of Data Augmentation for Automatic Speech Recognition Systems. 10.1007/978-3-030-33904-3_63.

[8]Alva, M. A., Arancibia-Garcia, A., Chávez, F. W., Cieza-Terrones Michael, Herrera-Arana, V., & Ramos-Cosi, S. (2021). Abnormal pulmonary sounds classification algorithm using convolutional networks. *International Journal of Advanced Computer Science and Applications, 12*(6) doi:http://dx.doi.org/10.14569/IJACSA.2021.0120645

[9]Björn Schuller, Anton Batliner, Christian Bergler, Cecilia Mascolo, et al. "The INTERSPEECH 2021Computational Paralinguistics Challenge: COVID-19 Cough, COVID-19 Speech, Escalation & Primates".En:Proc.INTERSPEECH. 5 páginas, Brno, República Checa, 2021.

[10]Kurama, V. (s. f.). A Review of Popular Deep Learning Architectures: AlexNet, VGG16, and GoogleNet. /blog.paperspace.com/. https://blog.paperspace.com/popular-deep-learning-architectures-alexnet-vgg-googlenet/

[11]INDIA MASSANA, Miquel Àngel, Pooyan SAFARI a Francisco Javier HERNANDO PERICÁS. Double multi-head attention for speaker verification. Institute of Electrical and Electronics Engineers (IEEE), 2021. ISBN 978-1-7281-7605-5.

[12]BAEVSKI, Alexei et al. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. . 2020

[13]*CROSSENTROPYLOSS*.(2019).Pytorch.org. https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

[14]*BCEWITHLOGITSLOSS*.(2019).Pythorch.org. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html#bcewithlogitsloss

[15]]*A Gentle Introduction to ROC Curve and AUC in Machine Learning*. (2020). https://sefiks.com/.https://sefiks.com/2020/12/10/a-gentle-introduction-to-roc-curve-and-auc/