Universitat Politècnica de Catalunya

Facultat d'Informàtica de Barcelona

P3 TVD: Reconocimiento de comandos de voz

Tractament de la Veu i el Diàleg

Grau en Intel·ligència Artificial
Daniel Álvarez 23857151X
Albert Roca 48106974J

18/12/2025

Contents

1 Introduction

El reconocimiento automático del palabras o frases es una de las áreas fundamentales dentro del procesamiento del lenguaje. Dentro de este ámbito, el reconocimiento de comandos de voz se centra en la identificación de palabras o frases cortas pertenecientes a un vocabulario limitado, lo que permite diseñar sistemas eficientes y robustos para tareas específicas.

En este proyecto se aborda el problema del reconocimiento de comandos de voz utilizando el conjunto de datos proporcionado en la competición que se nos ha proporcionado para esta práctica, de Kaggle. El objetivo principal es desarrollar un sistema capaz de clasificar correctamente grabaciones de audio cortas en función del comando pronunciado. Los audios han sido grabados por distintos hablantes, lo que introduce variabilidad en aspectos como el timbre de voz, la velocidad de pronunciación y las condiciones de grabación.

Para resolver este problema se emplea un enfoque de aprendizaje supervisado basado en técnicas de aprendizaje profundo. Las señales de audio se transforman previamente en representaciones tiempo–frecuencia mediante espectrogramas, que permiten capturar la información acústica relevante del habla. A partir de estas representaciones, se entrena una red neuronal convolucional (CNN) con el objetivo de aprender patrones discriminativos que faciliten la clasificación de los diferentes comandos.

Los objetivos de esta práctica son: (i) analizar y preprocesar señales de voz para su uso en tareas de reconocimiento automático, (ii) aplicar modelos de aprendizaje profundo al reconocimiento de comandos de voz, y (iii) evaluar el rendimiento del sistema propuesto en un entorno de evaluación estandarizado.

2 Tipo Espectograma

En esta sección analizamos el impacto del tipo de espectograma usado como entrada del modelo en el rendimiento del sistema de reconocimiento de comandos de voz. Para ello, empleamos un mismo modelo base de clasificación y evaluamos su comportamiento utilizando diferentes tipos de espectrogramas, con el objetivo de comparar cómo cada representación influye en la capacidad del modelo para extraer información relevante del habla.

Concretamente, probamos un espectrograma básico obtenido a partir de la Transformada de Fourier de tiempo corto (STFT), así como representaciones más elaboradas orientadas al procesamiento del habla, como el espectrograma en escala Mel y los coeficientes cepstrales en frecuencia Mel (MFCC). Mantener fija la arquitectura del modelo nos permite aislar el efecto del tipo de espectrograma y realizar una comparación justa entre las distintas configuraciones.

Table 1: Comparación de resultados del modelo base utilizando distintos tipos de espectrogramas

Tipo de espectrograma	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
Básico (STFT)	0.80	0.7269	1.0729
MFCC	0.79	0.7269	0.9527
Mel	**0.87**	**0.7856**	**0.9319**

A partir de los resultados obtenidos, observamos diferencias significativas en el rendimiento del modelo base en función del tipo de espectrograma utilizado como entrada. El espectrograma básico, obtenido mediante la STFT, ofrece un rendimiento aceptable, aunque presenta una menor capacidad de generalización en comparación con representaciones más específicas para el habla.

El uso de MFCC mejora ligeramente la estabilidad y reduce la pérdida de validación, aunque el valor máximo de precisión en validación es similar al obtenido con el espectrograma básico. Esto sugiere que, si bien los MFCC capturan características relevantes del habla, la información comprimida puede limitar el rendimiento del modelo en esta tarea concreta.

Por otro lado, el espectrograma en escala Mel proporciona los mejores resultados globales, alcanzando la mayor precisión en validación y una menor pérdida. Esta representación ofrece una mejor adaptación a la percepción humana del sonido y conserva una mayor cantidad de información relevante para el reconocimiento de comandos de voz.

En base a estos resultados, decidimos utilizar el espectrograma Mel como representación de entrada en los experimentos posteriores, ya que proporciona el mejor equilibrio entre rendimiento y estabilidad del modelo.

3 Modelos

3.1 Básico

Como punto de partida, implementamos un modelo base sencillo con el objetivo de establecer una referencia común para la comparación de los distintos tipos de espectrogramas descritos en la sección anterior.

La arquitectura del modelo consta de una red neuronal convolucional secuencial. En primer lugar, la entrada corresponde al espectrograma de cada señal de audio, cuyo tamaño se adapta mediante una capa de redimensionado a una resolución fija de 32×32 píxeles. A continuación, se aplica una capa de normalización entrenada sobre el conjunto de entrenamiento, lo que permite estabilizar el proceso de aprendizaje y mejorar la convergencia del modelo.

Después, el modelo incluye una capa convolucional con 32 filtros y función de activación ReLU, encargada de extraer patrones locales relevantes de la representación espectral, seguida de una capa de *max pooling* que reduce la dimensionalidad espacial. La salida de estas capas se aplana mediante una capa *Flatten* y se introduce en una capa completamente conectada de 64 neuronas con activación ReLU. Finalmente, la capa de salida consta de tantas neuronas como clases de comandos, produciendo las predicciones finales del modelo.

El modelo se entrena utilizando el optimizador Adam y la función de pérdida *Sparse Categorical Crossentropy*, adecuada para problemas de clasificación multiclase con etiquetas enteras.

Los resultados obtenidos con este modelo base corresponden a los presentados en la sección de comparación de espectrogramas. En particular, el mejor rendimiento se alcanza cuando se utiliza el espectrograma en escala Mel como entrada, superando tanto al espectrograma básico como a los MFCC en términos de precisión de validación. Por este motivo, en los experimentos posteriores se adopta el espectrograma Mel como representación de entrada por defecto.

3.2 Dropout

Con el objetivo de mejorar la capacidad de generalización del modelo base y reducir el posible sobreajuste, implementamos una extensión directa del modelo CNN básico incorporando una capa de *Dropout*. Esta modificación nos permite evaluar el impacto de la regularización manteniendo constante el resto de la arquitectura.

La estructura del modelo se mantiene idéntica a la del modelo base descrito anteriormente, añadiendo una capa de Dropout con una tasa de 0.3 tras la capa densa de 64 neuronas. De este modo, durante el entrenamiento se desactivan aleatoriamente un 30% de las neuronas de dicha capa, forzando al modelo a aprender representaciones más robustas y menos dependientes de características específicas del conjunto de entrenamiento.

Table 2: Resultados del modelo CNN básico con Dropout utilizando espectrogramas Mel

Modelo	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
CNN Básico (Mel)	0.87	0.7856	0.9319
CNN Básico + Dropout	0.74	0.7744	0.8642

A partir de los resultados obtenidos, observamos que la incorporación de la capa de Dropout reduce ligeramente la precisión de entrenamiento en comparación con el modelo base, lo cual es esperable debido al efecto regularizador introducido durante el aprendizaje. Sin embargo, el modelo con Dropout presenta una pérdida de validación menor y una evolución más estable a lo largo del entrenamiento.

En términos de precisión en validación, el modelo con Dropout alcanza valores similares a los del modelo base, aunque ligeramente inferiores en su máximo.

3.3 RNN BiLSTM

A continuación queremos evaluar si las redes neuronales recurrentes sirven para modelar las dependencias temporales presentes en las señales de voz, implementamos un modelo basado exclusivamente en redes LSTM bidireccionales (BiLSTM). A diferencia de los modelos basados en convoluciones, esta arquitectura se centra en explotar la información secuencial del habla sin emplear capas convolucionales ni mecanismos de atención.

La arquitectura del modelo consta de dos capas LSTM bidireccionales. La primera capa produce una secuencia de salidas que nos permite capturar dependencias temporales tanto pasadas como futuras, mientras que la segunda capa resume esta información en una representación de mayor nivel. A continuación, aplicamos una capa de Dropout seguida de una capa densa de salida con tantas neuronas como clases.

Table 3: Resultados del modelo RNN BiLSTM utilizando espectrogramas Mel

Modelo	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
RNN BiLSTM (Mel)	0.93	0.9058	0.3116

Los resultados obtenidos muestran una mejora significativa respecto a los modelos basados en arquitecturas CNN. El modelo RNN BiLSTM alcanza una precisión de validación notablemente superior, lo que indica que somos capaces de modelar de forma más efectiva las dependencias temporales presentes en las señales de voz.

3.4 RNN BiGRU

Ahora implementamos un modelo basado en GRU bidireccionales (BiGRU) como preprocesamiento, redimensionamos los espectrogramas Mel a una resolución de 96×96 y aplicamos una capa de normalización ajustada sobre el conjunto de entrenamiento. A continuación, reorganizamos el espectrograma en forma de secuencia temporal, de manera que el modelo recibe 96 pasos temporales con 96 características por paso.

La arquitectura contiene dos capas GRU bidireccionales. La primera devuelve una secuencia completa (*return_sequences=True*) para preservar la información temporal en todos los pasos, mientras que la segunda resume esta secuencia en un vector de características.

Table 4: Resultados del modelo RNN BiGRU utilizando espectrogramas Mel

Modelo	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
RNN BiGRU (Mel)	0.94	0.9199	0.2744

Los resultados obtenidos muestran que el modelo BiGRU alcanza un rendimiento muy alto, obteniendo una precisión de validación cercana al 92% y una pérdida de validación baja. Observamos además una convergencia progresiva y estable durante el entrenamiento, lo que sugiere que el modelo aprende representaciones temporales útiles.

Al comparar con el modelo BiLSTM, comprobamos que el BiGRU logra una mejora en precisión de validación y una reducción de la pérdida, lo cual es coherente con el hecho de que las GRU suelen ofrecer un compromiso favorable entre capacidad de modelado y complejidad.

3.5 CNN

Ahora queremos evaluar el rendimiento de una red convolucional (CNN). Este modelo sirve como referencia para comparar el comportamiento de arquitecturas convolucionales frente a modelos recurrentes, híbridos CNN–RNN y modelos basados en atención.

El modelo procesa la información mediante una jerarquía de bloques convolucionales que permiten extraer patrones locales tiempo–frecuencia de manera progresiva. La arquitectura está compuesta por cuatro bloques convolucionales. Cada bloque incluye una capa Conv2D con activación ReLU seguida de normalización por lotes (*Batch Normalization*), y en los tres primeros bloques se aplica además *Max Pooling* para reducir la dimensionalidad. El número de filtros se incrementa progresivamente de 32 a 256, permitiendo al modelo capturar características de mayor nivel a medida que aumenta la profundidad.

Tras la extracción convolucional, utilizamos una capa de *Global Average Pooling* para agregar la información. Finalmente, una capa densa con activación softmax produce las predicciones finales para la clasificación multiclase.

Table 5: Resultados del modelo CNN utilizando espectrogramas Mel

Modelo	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
CNN profunda (Mel)	0.97	0.9343	0.2426

Los resultados obtenidos muestran que la arquitectura CNN profunda alcanza un rendimiento muy competitivo, superando a los modelos recurrentes en cuanto a precisión.

Observamos que, a pesar de no modelar explícitamente dependencias temporales largas, el modelo es capaz de capturar información relevante del habla mediante las convoluciones y el aumento progresivo del número de filtros. La diferencia entre la precisión de entrenamiento y validación es moderada, lo que sugiere un buen equilibrio entre capacidad de aprendizaje y generalización, apoyado por el uso de normalización por lotes y Dropout.

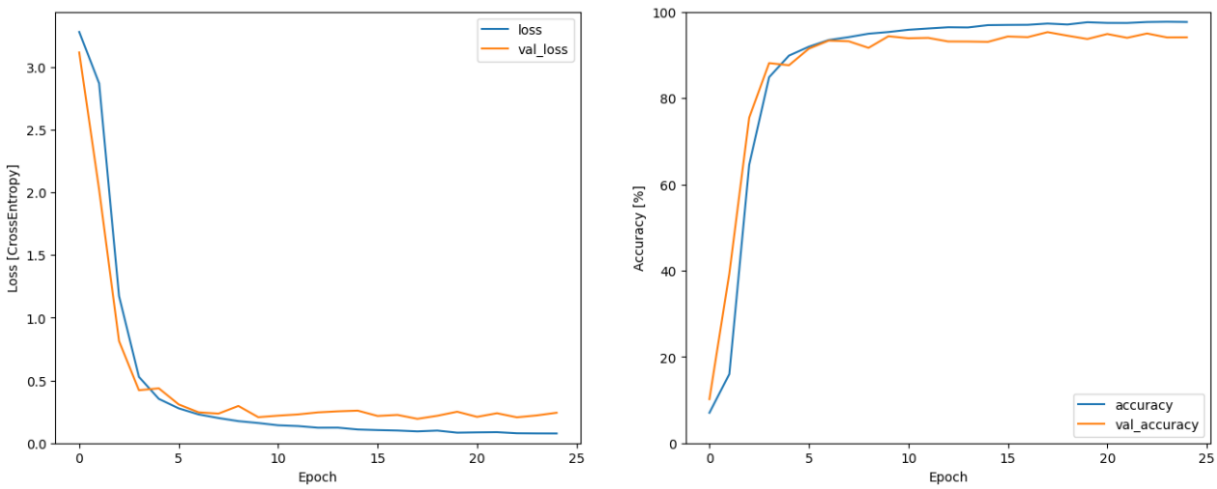

Figure 1: Curvas de loss y accuracy del modelo

3.6 CNN + Bidirectional GRU

Bien, en este apartado vamos a combinar las ventajas de las arquitecturas convolucionales y recurrentes, es por eso que implementamos un modelo híbrido CNN + GRU bidireccional. Este enfoque nos permite aprovechar la capacidad de las convoluciones para extraer patrones espectrales locales, al mismo tiempo que utilizamos una capa recurrente para modelar la dimensión temporal del espectrograma.

La arquitectura CNN consta de tres bloques. Los dos primeros incluyen capas Conv2D con kernels de 5×5, normalización y *Max Pooling* con *padding*. El tercer bloque utiliza una capa convolucional con kernel 3×3 y normalización por lotes, sin aplicar pooling, con el objetivo de preservar una representación rica antes del procesamiento recurrente.

Después de esto, reorganizamos los mapas de características resultantes en forma de secuencia temporal, convirtiendo la representación bidimensional en una secuencia de vectores de características. Sobre esta secuencia aplicamos una capa GRU bidireccional con 128 unidades, lo que nos permite modelar dependencias temporales tanto pasadas como futuras.

Table 6: Resultados del modelo CNN + Bidirectional GRU utilizando espectrogramas Mel

Modelo	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
CNN + BiGRU (Mel)	0.97	0.9530	0.1934

Los resultados nos muestran que el modelo CNN + BiGRU alcanza el mejor rendimiento entre todas las arquitecturas evaluadas. La combinación de convoluciones y una capa recurrente bidireccional permite capturar de forma eficaz todo, lo que se traduce en una mejora significativa de la precisión de validación.

Estos resultados confirman que la combinación de arquitecturas convolucionales y recurrentes constituye una estrategia muy efectiva para el reconocimiento de comandos de voz. Por este motivo, consideramos el modelo CNN + BiGRU como la arquitectura más sólida de las evaluadas y la utilizamos como referencia principal en la comparación final de modelos.

3.7 SpecAugment

Para mejorar la capacidad de generalización del modelo, aplicamos la técnica de aumento de datos *SpecAugment* sobre los espectrogramas Mel del conjunto de entrenamiento. Esta técnica consiste en introducir máscaras aleatorias en las dimensiones de frecuencia y tiempo del espectrograma.

En concreto, utilizamos *frequency masking* para eliminar bandas completas de frecuencia y *time masking* para eliminar intervalos temporales, ambos seleccionados de forma aleatoria. De este modo, forzamos al modelo a no depender de regiones específicas del espectrograma y a aprender representaciones más robustas.

SpecAugment se aplica únicamente durante el entrenamiento, manteniendo el conjunto de validación sin modificaciones. Para evaluar su impacto, entrenamos el modelo CNN + GRU bidireccional descrito anteriormente, manteniendo fija la arquitectura.

Table 7: Resultados del modelo CNN+BiGRU con SpecAugment (Mel)

Configuración	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
CNN+BiGRU + SpecAugment	0.9079	0.9588	0.1464

Los resultados obtenidos con SpecAugment muestran una mejora clara en la capacidad de generalización del modelo, alcanzando una precisión de validación cercana al 96%. La pérdida de validación es baja, lo que indica un aprendizaje estable y efectivo.

Además, observamos que la precisión en entrenamiento es inferior a la de validación, lo cual es coherente con el efecto regularizador introducido por la augmentación. Esto nos confirma que SpecAugment contribuye a reducir el sobreajuste y a mejorar la robustez del modelo.

3.8 SpecAugment + Random Erasing

Para introducir una regularización más agresiva, aplicamos la técnica de *Random Erasing* directamente sobre los espectrogramas Mel. Esta técnica consiste en eliminar aleatoriamente un parche rectangular del espectrograma, cuyo tamaño y posición se seleccionan de forma aleatoria y se aplica con una probabilidad fija.

Random Erasing obliga al modelo a no depender de regiones locales concretas del espectrograma y aumenta su robustez frente a ruido y variabilidad acústica. En este experimento, utilizamos exactamente la misma arquitectura que en el modelo base, aplicando la augmentación únicamente durante el entrenamiento.

Finalmente, evaluamos la combinación de *SpecAugment* y *Random Erasing* como técnica de aumento de datos. En este caso, aplicamos primero SpecAugment para enmascarar bandas temporales y de frecuencia, y posteriormente Random Erasing para eliminar parches rectangulares del espectrograma.

Table 8: Resultados de los modelos CNN+BiGRU con técnicas de aumento de datos (Mel)

Configuración	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
CNN+BiGRU + Random Erasing	0.8561	0.9593	0.1493
CNN+BiGRU + SpecAugment + Random Erasing	0.7542	0.9519	0.1670

Al combinar SpecAugment y Random Erasing obtenemos un rendimiento elevado, con una precisión de validación del 95.19%. Sin embargo, el resultado es ligeramente inferior al obtenido utilizando únicamente Random Erasing, tanto en precisión como en pérdida de validación.

Además, observamos una mayor diferencia entre la precisión de entrenamiento y validación, lo que indica un efecto regularizador más fuerte. Esto sugiere que la combinación de ambas técnicas introduce una augmentación más agresiva, que no aporta una mejora adicional respecto al uso exclusivo de Random Erasing en este caso.

3.9 CNN–BiGRU con Data Augmentation integrado

Aquí hemos implementado una arquitectura híbrida CNN-BiGRU en la que incorporamos técnicas de aumento de datos directamente dentro del modelo. El objetivo es evaluar si aplicar la augmentación de forma integrada durante el entrenamiento mejora la capacidad de generalización frente a la misma arquitectura sin aumento de datos.

Hemos hecho que antes del preprocesamiento, añadimos una capa de *data augmentation* que introduce desplazamientos y escalados aleatorios en los ejes de tiempo y frecuencia, simulando de forma suave técnicas como SpecAugment.

A continuación, el modelo procesa la entrada mediante tres bloques convolucionales para extraer características locales. Después aplicamos una capa GRU bidireccional para modelar dependencias temporales. Finalmente, utilizamos una capa de Dropout para regularización y una capa densa con activación softmax para la clasificación.

La augmentación de datos se aplica únicamente durante el entrenamiento.

Table 9: Resultados del modelo CNN–BiGRU con data augmentation integrado (Mel)

Configuración	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
CNN–BiGRU + Augmentación integrada	0.9497	0.9497	0.1785

Con la augmentación integrada dentro del modelo obtenemos un rendimiento alto, alcanzando un máximo de 94.97% de precisión en validación (mejor época: 17) con una pérdida de validación de 0.1785.

En comparación con las técnicas de augmentación aplicadas directamente al dataset (SpecAugment o Random Erasing), este enfoque no mejora el mejor resultado obtenido, aunque mantiene una generalización sólida y un entrenamiento estable.

3.10 Conformer (CNN + Self-Attention)

Aquí introducemos el modelo Conformer, que combina convolucional con mecanismos de atención para capturar dependencias temporales de largo alcance. El objetivo de este experimento es evaluar el rendimiento de un modelo basado en self-attention y compararlo con arquitecturas convolucionales, recurrentes e híbridas.

Después del preprocesamiento que hemos ido usando, utilizamos un front-end convolucional compuesto por varias capas Conv2D con normalización y reducción mediante *max pooling*.

Después de esto añadimos una codificación posicional entrenable. Sobre esta secuencia aplicamos dos bloques Conformer simplificados que incluyen capas *feed-forward*, multi-head attention y conexiones residuales con normalización. Finalmente, agregamos la información temporal mediante *Global Average Pooling*, aplicamos Dropout y utilizamos una capa densa con activación softmax para la clasificación.

Table 10: Resultados del modelo Conformer (CNN + Self-Attention) (Mel)

Modelo	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
Conformer (CNN + Attention)	0.9693	0.9261	0.2814

El Conformer alcanza una precisión de validación del 92.61%, con una pérdida de validación de 0.2814. El entrenamiento llega a un *train accuracy* alto, indicando buena capacidad de ajuste.

Sin embargo, en comparación con el modelo CNN+BiGRU con augmentación (SpecAugment/Random Erasing), el Conformer no mejora el mejor resultado obtenido, por lo que su mayor complejidad no se traduce en un beneficio claro en este experimento.

4 Conclusiones

En este trabajo hemos analizado distintas arquitecturas de deep learning para la tarea de reconocimiento de comandos de voz, comenzando por modelos convolucionales simples y avanzando hacia arquitecturas recurrentes, híbridas y basadas en atención. A lo largo de los experimentos hemos comprobado que el uso de *Mel-espectrogramas* proporciona una representación más adecuada del habla, permitiendo obtener mejores resultados que con espectrogramas básicos.

En cuanto a las arquitecturas, los modelos híbridos **CNN–BiGRU** destacan frente a las CNN puras y a las RNN aisladas, ya que combinan de forma efectiva la extracción de patrones locales. Además, las técnicas de aumento de datos han demostrado ser clave para mejorar la generalización, especialmente **SpecAugment** y **Random Erasing**, reduciendo el sobreajuste y aumentando la robustez del sistema.

Finalmente, aunque arquitecturas más complejas como el **Conformer** ofrecen un buen rendimiento, no superan a los mejores modelos CNN–BiGRU con augmentación. Esto sugiere que, para esta tarea, un equilibrio adecuado entre arquitectura y técnicas de regularización resulta más efectivo que incrementar la complejidad del modelo.

4.1 Resumen de resultados

Table 11: Resumen de resultados de todos los modelos evaluados (Mel)

Modelo	**Train Accuracy**	**Val. Accuracy**	**Val. Loss**
CNN Básico	0.87	0.7856	0.9319
CNN Básico + Dropout	0.74	0.7744	0.8642
RNN BiLSTM	0.93	0.9058	0.3116
RNN BiGRU	0.94	0.9199	0.2744
CNN profunda	0.97	0.9343	0.2426
CNN + BiGRU	0.97	0.9530	0.1934
CNN + BiGRU + SpecAugment	0.9079	0.9588	0.1464
CNN + BiGRU + Random Erasing	0.8561	0.9593	0.1493
CNN + BiGRU + SpecAugment + Random Erasing	0.7542	0.9519	0.1670
CNN–BiGRU + DataAugmentation	0.9497	0.9497	0.1785
Conformer (CNN + Self-Attention)	0.9693	0.9261	0.2814

4.2 Mejor modelo

El mejor modelo obtenido es el **CNN+BiGRU con Random Erasing**, que alcanza la mayor precisión de validación (**95.93%**) junto con una pérdida baja. Este resultado confirma que la combinación de una arquitectura híbrida con una técnica de aumento de datos bien ajustada es la estrategia más efectiva para el reconocimiento de comandos de voz en este conjunto de datos. Hay que tener en cuenta que si hubierámos augmentado el número de epochs es posible que podríamos haber obtenido mayores resultados, pero para todos los experimentos las fijamos a 25.