

INSTITUTO TECNOLÓGICO DE CHIHUAHUA
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

***“MODELO DE APRENDIZAJE PROFUNDO PARA EL
RECONOCIMIENTO DE EMOCIONES POR MEDIO
DEL HABLA EN ESPAÑOL”***

TESIS

PARA OBTENER EL GRADO DE

***MAESTRO EN CIENCIAS EN INGENIERÍA
ELECTRÓNICA***

PRESENTA:

RICARDO STEVE ANG FOSTER

DIRECTOR DE LA TESIS:
DR. JUAN ALBERTO RAMÍREZ QUINTANA

CHIHUAHUA, CHIH., FEBRERO 2026.



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO





Instituto Tecnológico de Chihuahua
Departamento de Estudios de Posgrado e Investigación

Chihuahua, Chih. **17 de marzo de 2026**

**C. RICARDO STEVE ANG FOSTER
PRESENTE**

En cumplimiento con los requerimientos para la obtención del grado de Maestro en Ciencias en Ingeniería Electrónica y a propuesta de su Comité Tutorial, la División de Estudios de Posgrado e Investigación le concede la autorización para imprimir la tesis titulada **"Modelo de aprendizaje profundo para el reconocimiento de emociones por medio del habla en español"** dirigida por el Dr. Juan Alberto Ramírez Quintana con el siguiente contenido de capítulos:

CAPÍTULOS 1: MARCO TEÓRICO

CAPÍTULO 2: ANTECEDENTES

CAPÍTULO 3: CONJUNTOS DE DATOS

CAPÍTULO 4: EXPERIMENTACIÓN CON MODELO POPULARES SER

CAPÍTULO 5: ARQUITECTURA PROPUESTA

CAPÍTULO 6: RESULTADOS Y DISCUSIÓN

CAPÍTULO 7: CONCLUSIONES

ATENTAMENTE

Excelencia en Educación Tecnológica®
"La Técnica por el Engrandecimiento de México"

LEONCIO OCHOA CERVANTES
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

LOC/vgo



2026
año de
**Margarita
Maza**





Instituto Tecnológico de Chihuahua
Departamento de Estudios de Posgrado e Investigación

Chihuahua, Chih., 13/marzo/2026

LEONCIO OCHOA CERVANTES
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
PRESENTE

En cumplimiento con los requerimientos para la obtención del grado de Maestro en Ciencias en Ingeniería Electrónica, le notificamos que el documento de tesis del alumno **C. RICARDO STEVE ANG FOSTER**, titulado **"Modelo de aprendizaje profundo para el reconocimiento de emociones por medio del habla en español"** dirigido por el Dr. Juan Alberto Ramírez Quintana, ha sido aprobado y aceptado para su impresión.

Por lo anterior, proponemos le sea concedida la autorización de impresión correspondiente.

Agradeciendo la atención a la presente, quedamos de usted:

ATENTAMENTE
Excelencia en Educación Tecnológica®
La técnica por el Engrandecimiento de México

Juan Ramírez
DR. JUAN ALBERTO RAMÍREZ QUINTANA
MIEMBRO DEL COMITÉ TUTORIAL

[Firma]
DR. LUIS FRANCISCO CORRAL MARTÍNEZ
MIEMBRO DEL COMITÉ TUTORIAL

Abimael G
DR. ABIMAEEL GUZMÁN PANDO
MIEMBRO DEL COMITÉ TUTORIAL

[Firma]
MTRA. ALMA DELIA CORRAL SÁENZ
MIEMBRO DEL COMITÉ TUTORIAL



2026
año de
Margarita
Maza





CARTA CESIÓN DE DERECHOS

En la ciudad de Chihuahua el día 17 de marzo de 2026, el que suscribe, C. RICARDO STEVE ANG FOSTER con número de control G24060167, de la Maestría en Ciencias en Ingeniería Electrónica, adscrita a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del el Dr. Juan Alberto Ramírez Quintana y cede los derechos del trabajo titulado " Modelo de aprendizaje profundo para el reconocimiento de emociones por medio del habla en español", al Tecnológico Nacional de México y/o Instituto Tecnológico de Chihuahua para su difusión, divulgación, transmisión, reproducción, así como su digitalización con fines académicos y de investigación.

C. RICARDO STEVE ANG FOSTER





DECLARACIÓN DE ORIGINALIDAD

En la ciudad de Chihuahua el día 17 de marzo de 2026, el que suscribe, C. RICARDO STEVE ANG FOSTER de la Maestría en Ciencias en Ingeniería Electrónica, con número de control G24060167, adscrito a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua, manifiesta que es autor intelectual de la tesis titulada " Modelo de aprendizaje profundo para el reconocimiento de emociones por medio del habla en español" bajo la dirección el Dr. Juan Alberto Ramírez Quintana; que el contenido es original y que las fuentes de información consultadas para su fundamentación están debidamente citadas y referenciadas.

C. RICARDO STEVE ANG FOSTER



AGRADECIMIENTOS

Agradezco principalmente a Dios, a mi familia, mis compañeros de laboratorio, a mis amigos, a mi asesor de tesis el Dr. Juan Alberto Ramírez Quintana, al SECIHTI y a todos los que me acompañaron durante esta etapa de mi vida que gracias a ellos pude concluir con este trabajo. Gracias.

RESUMEN

MODELO DE APRENDIZAJE PROFUNDO PARA EL RECONOCIMIENTO DE EMOCIONES POR MEDIO DEL HABLA EN ESPAÑOL

Ing. Ricardo Steve Ang Foster
Maestría en Ciencias en Ingeniería Electrónica
División de Estudios de Posgrado e Investigación del
Instituto Tecnológico de Chihuahua
Chihuahua, Chih. 201
Director de Tesis: Dr. Juan Alberto Ramírez Quintana

El reconocimiento de emociones a partir del habla es un área de investigación relevante dentro de la interacción humano-computadora, ya que la voz contiene información emocional implícita que puede ser aprovechada por sistemas inteligentes. En los últimos años, el aprendizaje profundo ha permitido mejorar significativamente el desempeño de los sistemas de reconocimiento de emociones. Sin embargo, la mayoría de los estudios se han enfocado en idiomas como el inglés o el alemán, existiendo una exploración aún limitada para el idioma español. Por lo tanto, en esta tesis se presenta el diseño y la evaluación de un modelo de aprendizaje profundo para el reconocimiento de emociones a partir del habla en español. Dicho modelo utiliza el espectrograma logarítmico de Mel para la extracción de características y el reconocimiento de emociones mediante una arquitectura híbrida que combina redes neuronales convolucionales profundas y modelos Transformer.

La metodología para la obtención del modelo propuesto consistió en realizar una evaluación comparativa de diversos modelos populares de reconocimiento de emociones, incluyendo arquitecturas convolucionales, recurrentes y basadas en Transformers, utilizando los conjuntos de datos EmoMatchSpanishDB y Spanish MEACorpus 2023. A partir de los resultados experimentales, se identificó que las arquitecturas híbridas en configuración en serie, compuestas por una ResNet-50 y un bloque Transformer/Conformer, presentan un mejor desempeño frente a modelos del estado del arte, logrando resultados robustos en métricas como la exactitud balanceada y la medida-F1.

Los resultados obtenidos confirman que es posible diseñar modelos de aprendizaje profundo eficaces para el reconocimiento de emociones en el habla en español, incluso cuando se dispone de conjuntos de datos de tamaño limitado y con clases desbalanceadas.

CONTENIDO

INTRODUCCIÓN	1
I. MARCO TEÓRICO	3
1.1 Reconocimiento de emociones	3
1.1.1 Reconocimiento de emociones por medio del habla.....	4
1.2 Características del habla	5
1.2.1 Características prosódicas.....	5
1.2.2 Características espectrales.....	5
1.3 Espectrograma logarítmico de Mel.....	6
1.4 Coeficientes Cepstrales de Frecuencia de Mel	7
1.5 Aprendizaje profundo	8
1.6 Redes Neuronales Convolucionales.....	8
1.7 Redes Neuronales Recurrentes	9
1.7.1 Memoria a Corto y Largo Plazo	10
1.7.2 Memoria a Corto y Largo Plazo Bidireccional	11
1.8 Transformers	11
1.9 Métricas de evaluación de desempeño.....	13
1.9.1 Exactitud.....	14
1.9.2 Precisión	14
1.9.3 Sensibilidad	14
1.9.4 Medida-F1	15
1.9.5 Métrica de exactitud balanceada.....	15
II. ANTECEDENTES.....	16
2.1 ¿Cómo desarrollar un sistema de reconocimiento de emociones por medio del habla?..	16
2.2 Conjuntos de datos de habla emocional en español.....	17
2.3 Reconocimiento de emociones por medio del habla en el estado del arte.....	19
2.4 Resumen del estado del arte.....	21
III. CONJUNTOS DE DATOS.....	23

3.1	EmoMatchSpanishDB.....	23
3.2	Spanish MEACorpus 2023	24
IV.	EXPERIMENTACIÓN CON MODELOS POPULARES SER.....	26
4.1	Metodología	26
4.2	Modelos seleccionados	27
4.2.1	ResNet50	28
4.2.2	EfficientNetB0	29
4.2.3	MobileNetV2	30
4.2.4	InceptionV3	31
4.2.5	VGG16	34
4.2.6	Vision Transformer	35
4.3	Preprocesamiento	36
4.4	Hiperparámetros de entrenamiento para los experimentos	37
4.5	Experimentos con EmoMatchSpanishDB.....	40
4.6	Experimentos con Spanish MEACorpus 2023	42
4.7	Resultados con EmoMatch y MEACorpus	44
4.8	Experimentos con variantes de ResNet.....	47
4.8.1	Resultados en variantes de ResNet.....	50
4.9	Experimentación de modelos híbridos ResNet50-Transformer en serie y paralelo	51
4.9.1	Resultados de la arquitectura ResNet50-Transformer en serie.....	53
4.9.2	Resultados de la arquitectura ResNet50-Transformer en paralelo	54
4.9.3	Análisis de resultados.....	54
4.10	Análisis de distintas arquitecturas Transformer.....	55
4.10.1	Transformer de espectrogramas de audio	56
4.10.2	Performer	57
4.10.3	Conformer	57
V.	ARQUITECTURA PROPUESTA	60
5.1	Entrada del modelo	60
5.2	Regularizaciones	62

5.2.1 Penalización L2	62
5.2.2 Dropout	63
5.3 Módulo ResNet50	64
5.3.1 Bloque inicial (Conv1_x)	64
5.3.2 Bloque residual 1 (Conv2_x)	66
5.3.3 Bloque residual 2 (Conv3_x)	67
5.3.4 Bloque residual 3 (Conv4_x)	69
5.3.5 Bloque residual 4 (Conv5_x)	70
5.4 Módulo Transformer/Conformer	72
5.4.1 Mecanismos de atención	73
5.4.2 Red Totalmente Conectada	76
5.4.3 Arquitectura del bloque Transformer/Conformer	76
5.5 Bloque de clasificación	79
VI. RESULTADOS Y DISCUSIÓN	82
6.1. Resultados con Spanish MEACorpus 2023	83
6.2. Resultados con EmoMatchSpanishDB	87
6.3. Pruebas de significancia estadística	91
6.4. Comparación con modelos del estado del arte	92
VII. CONCLUSIONES	95
GLOSARIO	97
REFERENCIAS	100

LISTA DE FIGURAS

Figura 1.1. Estructura básica de un sistema de reconocimiento de emociones por medio del habla.	5
Figura 1.2. Ejemplo de una red neuronal convolucional.	9
Figura 1.3. Estructura de una red neuronal recurrente básica con bucle.	10
Figura 1.4. Estructura Long Short-Term Memory.	11
Figura 1.5. Estructura básica de un Transformer.	13
Figura 2.1. Principales características de un sistema SER.	16
Figura 3.1. Diagrama de flujo del proceso utilizado para la creación de EmoSpanishDB y EmoMatchSpanishDB.	18
Figura 3.2. Flujo de trabajo recomendado para trabajar con Spanish MEACorpus 2023.	20
Figura 4.1. Simbología utilizada para describir los modelos.	29
Figura 4.2. Arquitectura de ResNet50.	30
Figura 4.3. Arquitectura de EfficientNetB0.	31
Figura 4.4. Arquitectura de los bloques MBConv.	31
Figura 4.5. Arquitectura de MobileNetV2.	32
Figura 4.6. Bloques Bottleneck con Stride = 1 y Stride = 2.	32
Figura 4.7. Arquitectura de InceptionV3.	33
Figura 4.8. Bloque Inception-A.	34
Figura 4.9. Bloque Inception-B.	34
Figura 4.10. Bloque Inception-C.	35
Figura 4.11. Arquitectura de VGG16.	36
Figura 4.12. Arquitectura de Vision Transformer.	37
Figura 4.13. Arquitectura general de experimentos con CNNs para EmoMatchSpanishDB.	41
Figura 4.14. Arquitectura con capas LSTM para EmoMatchSpanishDB.	42
Figura 4.15. Arquitectura con capas BiLSTM para EmoMatchSpanishDB.	42
Figura 4.16. Arquitectura con capas LSTM para Spanish MEACorpus 2023.	44
Figura 4.17. Arquitectura con capas BiLSTM para Spanish MEACorpus 2023.	44
Figura 4.18. Arquitectura ResNet50-B1 con un bloque residual.	49
Figura 4.19. Arquitectura ResNet50-B2 con dos bloques residuales.	49
Figura 4.20. Arquitectura ResNet50-B3 con tres bloques residuales.	50
Figura 4.21. Arquitectura ResNet50-B4 con todos los bloques residuales.	50
Figura 4.22. Modelo híbrido en serie con ResNet50 y bloques Transformer.	53
Figura 4.23. Modelo híbrido en paralelo con ResNet50 y bloques Transformer.	54
Figura 4.24. Arquitectura del Audio Spectrogram Transformer.	57
Figura 4.25. Análisis de las matrices en los mecanismos de atención convencionales y en el mecanismo de atención FAVOR+.	58
Figura 4.26. Arquitectura de un bloque Conformer.	59

Figura 5.1. Arquitectura propuesta ResNet50-Transformer/Conformer.	61
Figura 5.2. Arquitectura básica de un bloque residual.	65
Figura 5.3. Arquitectura del Bloque de entrada.	66
Figura 5.4. Arquitectura del Bloque residual 1.	68
Figura 5.5. Arquitectura del Bloque residual 2.	69
Figura 5.6. Arquitectura del Bloque residual 3.	71
Figura 5.7. Arquitectura del Bloque residual 4.	73
Figura 5.8. Atención de Producto Punto Escalado.	75
Figura 5.9. Atención Multi-Cabeza.	76
Figura 5.10. Bloque Transformer desarrollado.	77
Figura 5.11. Bloque Conformer desarrollado.	80
Figura 5.12. Bloque de clasificación.	81
Figura 6.1. Gráficas de desempeño en el entrenamiento del modelo ResNet50-Conformer con MEACorpus.	85
Figura 6.2. Matriz de confusión de la partición uno en ResNet50-Conformer con MEACorpus.	86
Figura 6.3. Gráficas de desempeño en el entrenamiento del modelo ResNet50-Transformer con MEACorpus.	87
Figura 6.4. Matriz de confusión de la partición uno en ResNet50-Conformer con MEACorpus.	88
Figura 6.5. Gráficas de desempeño en el entrenamiento del modelo ResNet50-Conformer con EmoMatch.	90
Figura 6.6. Matriz de confusión de la partición uno en ResNet50-Conformer con EmoMatch.	90
Figura 6.7. Gráficas de desempeño en el entrenamiento del modelo ResNet50-Transformer con EmoMatch.	92
Figura 6.8. Matriz de confusión de la partición uno en ResNet50-Transformer con EmoMatch.	92

LISTA DE TABLAS

Tabla 1.1. Características prosódicas y espectrales más utilizadas.	6
Tabla 2.1. Información sintetizada de modelos de SER en el estado del arte.	23
Tabla 4.1. Resultados de los experimentos realizados con CNN en EmoMatchSpanishDB.	45
Tabla 4.2. Resultados de los experimentos realizados con RNN y ViT en EmoMatchSpanishDB.	45
Tabla 4.3. Resultados de los experimentos realizados con CNN en Spanish MEACorpus 2023.	46
Tabla 4.4. Resultados de los experimentos realizados con RNN y ViT en Spanish MEACorpus 2023.	46
Tabla 4.5. Resultados con las variantes de ResNet50.	51
Tabla 4.6. Resultados de los experimentos realizados con las variaciones de ResNet50+Transformer en serie.	54
Tabla 4.7. Resultados de los experimentos realizados con las variaciones de ResNet50+Transformer en paralelo.	55
Tabla 6.1. Resultados de validación cruzada en ResNet50-Conformer con MEACorpus.	84
Tabla 6.2. Resultados de validación cruzada en ResNet50-Tranformer con MEACorpus.	86
Tabla 6.3. Resultados de validación cruzada en ResNet50-Conformer con EmoMatch.	88
Tabla 6.4. Resultados de validación cruzada en ResNet50-Transformer con EmoMatch.	91
Tabla 6.5. Resultados de las pruebas de significancia estadística.	93
Tabla 6.6. Comparación con el estado del arte con MEACorpus.	94
Tabla 6.7. Comparación con el estado del arte con EmoMatch.	94

INTRODUCCIÓN

El reconocimiento de emociones ha cobrado un interés creciente en los últimos años debido a su potencial aplicación en áreas como la interacción humano-computadora, sistemas de asistencia virtual, análisis de comportamiento y tecnologías de apoyo en salud. Dentro de esta disciplina, una de las ramas emergentes que ha cobrado especial atención es el reconocimiento de emociones a partir del habla, ya que diversos estudios sugieren que el habla contiene un 38% de la información emocional asociada a un mensaje [1]. La capacidad de identificar el estado emocional de una persona a partir de su voz permite enriquecer la comunicación entre humanos y sistemas inteligentes, dotándolos de una mayor sensibilidad y adaptabilidad al contexto del usuario [2].

En el ámbito del aprendizaje automático, el reconocimiento de emociones en el habla se ha abordado tradicionalmente mediante la extracción de características acústicas definidas a priori y el uso de clasificadores tradicional [3]. Sin embargo, el auge del aprendizaje profundo ha permitido el desarrollo de modelos capaces de aprender representaciones más complejas y discriminativas a partir de la señal de audio o de sus representaciones espectrales, logrando mejoras significativas en el desempeño [4] [5] [6]. A pesar de estos avances, el idioma español ha sido menos estudiado en comparación con otros idiomas, pese a ser la cuarta lengua más hablada en el mundo, con más de 500 millones de hablantes nativos. Esta situación motiva la necesidad de desarrollar y evaluar modelos enfocados en el reconocimiento de emociones en el habla en español [7].

El reconocimiento de emociones en el habla en español representa diversos retos tales como la poca disponibilidad de conjuntos de datos y la variabilidad lingüística y cultural presente en los hablantes. Por ello, resulta necesario explorar arquitecturas y representaciones que permitan capturar de manera efectiva la información relevante contenida en la señal de voz, considerando tanto dependencias locales como globales en el dominio espectro-temporal.

En este contexto, el presente trabajo propone el desarrollo de un modelo de aprendizaje profundo basado en arquitecturas híbridas que tienen como entrada espectrogramas logarítmicos de Mel que combinan redes neuronales convolucionales profundas con arquitecturas tipo Transformer. Para validar el modelo propuesto, se utilizaron los conjuntos de datos Spanish MEACorpus 2023 y EmoMatchSpanishDB, ya que empleados en la literatura para el reconocimiento de emociones en el habla en español. Los resultados obtenidos permiten analizar el desempeño del modelo propuesto y su competitividad frente a aproximaciones existentes, incluyendo modelos de mayor complejidad.

La presente tesis se organiza de la siguiente manera: El Capítulo 1 presenta el marco teórico con los conceptos necesarios para el desarrollo de este trabajo. El Capítulo 2 describe los antecedentes y trabajos relacionados y el Capítulo 3 aborda técnicamente los conjuntos de datos utilizados. El Capítulo 4 detalla los experimentos realizados con modelos populares y en el Capítulo 5 se describe la arquitectura propuesta. El Capítulo 6 expone los resultados obtenidos y su comparación con el estado del arte. Finalmente, el Capítulo 7 presenta las conclusiones y el trabajo futuro.

I.MARCO TEÓRICO

Este capítulo define los conceptos necesarios para llevar a cabo el proyecto de investigación y se divide de la siguiente manera: la sección 1.1 corresponde a los elementos conceptuales relacionados con la tarea de reconocimiento emocional y la sección 1.2 describe la información relevante al habla. Las secciones 1.3 y 1.4 definen los espectrogramas empleados en esta investigación. Las secciones 1.5, 1.6, 1.7 y 1.8 presentan los conceptos esenciales del aprendizaje profundo y la sección 1.9 define las métricas de evaluación utilizadas para medir el desempeño de modelos.

1.1 Reconocimiento de emociones

La emoción se puede definir como una respuesta cognitiva, fisiológica y conductual ante estímulos externos, como la interacción con otros seres humanos. Esta respuesta abarca procesos cognitivos, experiencias subjetivas, expresión comunicativa e influencias conductuales. Las emociones son una fuente importante de información que proporciona contexto a las ideas o mensajes que se desean transmitir. Por ello, los seres humanos mantienen continuamente un proceso de reconocimiento de emociones durante la comunicación, al captar señales provenientes de gestos, lenguaje corporal y de la voz [2].

El campo del reconocimiento de emociones busca replicar y simular este proceso de manera artificial mediante diversas técnicas. Estas técnicas tienen aplicaciones relevantes en áreas como:

- La salud con diagnósticos del bienestar emocional.
- El marketing a la hora de analizar reacciones del consumidor.
- Las interacciones máquina-humano con asistentes virtuales.
- La seguridad para detectar emociones críticas en situaciones de riesgo.

Las emociones pueden ser reconocidas a partir de diferentes fuentes de información, que se dividen principalmente en dos categorías: señales físicas y señales fisiológicas. Entre las señales

físicas se encuentran las expresiones faciales, el análisis del texto, la voz, los gestos y las posturas corporales. Por otro lado, las señales fisiológicas incluyen bioseñales como electroencefalogramas (EEG), electrocardiogramas (ECG) y patrones respiratorios (RSP) [2].

En este contexto, las técnicas de inteligencia artificial (IA) se han convertido en herramientas fundamentales para el reconocimiento de emociones, ya que permiten analizar grandes volúmenes de datos mediante el reconocimiento de patrones. Estos algoritmos facilitan la identificación precisa de estados emocionales, abriendo nuevas posibilidades para desarrollar sistemas más inteligentes y empáticos.

1.1.1 Reconocimiento de emociones por medio del habla

El habla es una de las formas más naturales y orgánicas que los seres humanos utilizamos para expresar pensamientos, sentimientos e ideas [8]. Este proceso comienza con la selección de conceptos ubicados en el lexicón mental, los cuales se estructuran lingüísticamente y se organizan mediante procesos como la silabificación. Posteriormente, se accede a las representaciones fonológicas necesarias para producir los sonidos del habla, y se activa el plan motor que coordina los movimientos de los órganos articulatorios (como la lengua, los labios y la mandíbula) [9]. Este proceso lingüístico y motor está estrechamente vinculado con las emociones que experimenta el hablante, influyendo en el tono, la intensidad, el ritmo y otros aspectos del habla que transmiten información emocional adicional al contenido verbal. En resumen, una señal de habla contiene información del léxico, del hablante, de la emoción transmitida y del lenguaje en el que se transmite el mensaje [10]. Partiendo del hecho de que hay información emocional dentro del habla, era consecuente el considerar que existe una rama del reconocimiento de emociones basada en el habla, a esta se le conoce en inglés como *Speech Emotion Recognition* y ya ha sido mencionada en este trabajo como SER.

Un sistema SER se desarrolla con base en los pasos mostrados en la Figura 1.1. Primero se toma un conjunto de datos que cuente con la información del habla, a estos datos se les extraen las características apropiadas. El siguiente paso es un proceso de selección de características

para tomar aquellas que sean más relevantes para el reconocimiento de la emoción y eliminar el ruido. Por último, un algoritmo de clasificación se encargará de determinar la emoción presente en la señal de habla en relación a las características seleccionadas.

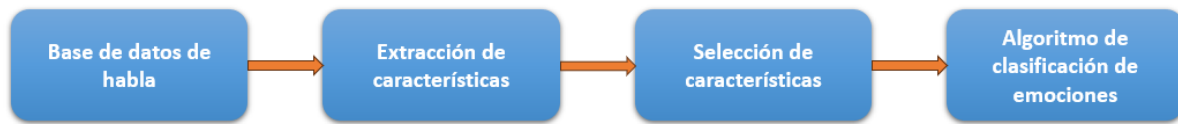


Figura 1.1. Estructura básica de un sistema de reconocimiento de emociones por medio del habla.

1.2 Características del habla

Se mencionaba anteriormente que parte fundamental de un sistema SER es la extracción y selección de características que se encuentran intrínsecas en las señales de habla. Es por esto que resulta importante definir el cómo se categorizan comúnmente las características acústicas emocionales; siendo estas las prosódicas, asociadas más a la información en el dominio del tiempo, y las espectrales, asociadas más a la información en el dominio de la frecuencia [3]. Se pueden ver algunos ejemplos de estas características en la Tabla 1.1.

1.2.1 Características prosódicas

Las características prosódicas corresponden a aquellos rasgos globales de la expresión oral que abarcan más de un sonido individual y no pueden segmentarse en unidades más pequeñas, tales como el acento del hablante, los tonos, la entonación y el ritmo. Muchos expertos han confirmado que características como la energía, la tasa de cruce por cero y la frecuencia del tono describen características prosódicas que expresan de manera significativa las emociones del hablante [11].

1.2.2 Características espectrales

Las características espectrales son propiedades basadas en la frecuencia que se obtienen al transformar las señales de audio del dominio del tiempo al dominio de la frecuencia utilizando la transformada de Fourier [11]. Debido a que el habla es una señal dinámica que está en

constante cambio, se deben establecer ventanas de tiempo para poder obtener las componentes por ventana y ser lo más preciso posible para abstraer la información de frecuencias.

Tabla 1.1. Características prosódicas y espectrales más utilizadas.

Características prosódicas	Características espectrales
Frecuencia fundamental (F0)	Coefficientes Cepstrales de Frecuencias Mel (MFCC)
Energía	Coefficientes de Predicción Lineal (LPC)
Frecuencia de tono	Coefficientes Cepstrales de Predicción Lineal (LPCC)
Ritmo	Coefficientes Cepstrales de Frecuencia de tono Gamma (GFCC)
Entonación	Predicción Lineal Perceptual (PLP)
Acento	Espectrogramas Mel
Tasa de cruce por cero	Espectrogramas log-Mel

1.3 Espectrograma logarítmico de Mel

El espectrograma logarítmico de Mel (espectrograma log-Mel) es una representación bidireccional diseñada para aproximarse a la forma en que el oído humano percibe el sonido. Se obtiene a partir de una señal discreta en el dominio del tiempo $x[n]$ mediante una serie de procesos sucesivos. En primer lugar, a la señal se le aplican un filtro de preénfasis, seguido de una segmentación en secciones de tiempo mediante ventanas, generalmente de Hamming. Posteriormente, se emplea la Transformada Rápida de Fourier (FFT) para convertir cada sección de tiempo al dominio de la frecuencia. Este procedimiento puede expresarse matemáticamente como en la ecuación 1.1.

$$X(k, m) = \sum_{n=0}^{N-1} (x[n + mh] - \alpha x[n + mh - 1])w[n]e^{-j\frac{2\pi}{N}kn} \quad (1.1)$$

Donde n es el índice temporal de la señal, N corresponde al tamaño de la ventana, m es el índice de la ventana temporal, h representa el salto entre ventanas, k es el índice de la frecuencia discreta, α es el coeficiente de preénfasis y $w[n]$ denota la ventana de Hamming.

El espectro resultante se pasa por un banco de filtros en escala Mel, el cual permite agrupar la energía espectral de acuerdo con la percepción auditiva humana. Finalmente, se aplica una transformación logarítmica a la energía obtenida en cada banda Mel. Esta conversión logarítmica refleja la percepción humana no lineal de la intensidad del sonido, haciendo que esta representación sea especialmente útil en tareas de reconocimiento de voz y emociones [12]. El cálculo del espectrograma log-Mel se expresa como en la ecuación 1.2.

$$S_{logMel}(m, i) = \log \left(\sum_{k=0}^{N/2} |X(k, m)|^2 H_i(k) + \epsilon \right) \quad (1.2)$$

Donde i representa el índice del filtro Mel, $|X(k, m)|^2$ corresponde a la energía espectral de la ventana m en la frecuencia discreta k , $H_i(k)$ es la respuesta del i -ésimo filtro Mel y ϵ es una constante pequeña positiva utilizada para evitar indeterminaciones numéricas al aplicar el logaritmo.

1.4 Coeficientes cepstrales de frecuencias Mel

Los coeficientes cepstrales de frecuencias Mel (MFCC por sus siglas en inglés) son características que modelan de forma compacta como el sistema auditivo humano percibe el sonido. Su cálculo implica transformar la señal de audio desde el dominio del tiempo al dominio de la frecuencia mediante el procedimiento de análisis espectral descrito en la ecuación 1.1, aplicar un banco de filtros en escala Mel y obtener el logaritmo de la energía espectral resultante. Posteriormente, se aplica una Transformada Discreta del Coseno (DCT) [5]. Esta transformación reduce la redundancia y permite capturar la envolvente espectral, que es útil para distinguir diferentes fonemas y emociones en la voz. Por su efectividad y robustez frente a variaciones de hablante y ruido, los MFCC son una de las características más utilizadas en tareas de reconocimiento del habla y emociones [7]. Matemáticamente, el cálculo de los coeficientes MFCC se expresa como en la ecuación 1.3.

$$MFCC(m, c_e) = \sum_{i=1}^M \log \left(\sum_{k=0}^{N/2} |X(k, m)|^2 H_i(k) + \epsilon \right) \cos \left[\frac{\pi c_e}{M} \left(i - \frac{1}{2} \right) \right] \quad (1.3)$$

donde m es el índice del frame temporal, c_e corresponde al índice del coeficiente cepstral, M es el número total de filtros Mel, $H_i(k)$ representa el i -ésimo filtro en escala Mel, $|X(k, m)|^2$ es la energía espectral y ϵ es una constante positiva pequeña utilizada para evitar indeterminaciones numéricas.

1.5 Aprendizaje profundo

El aprendizaje profundo (*deep learning*, DL) es una rama del aprendizaje automático basada en redes neuronales artificiales (RNAs) que proporcionan un nivel de abstracción profundo respecto a los datos con los cuales se alimenta al algoritmo. Esto se logra a través de un modelo de procesamiento multicapa o estructuras complejas que implican transformaciones no lineales. Entre las RNAs para aprendizaje profundo, se encuentran las redes neuronales profundas, que son sistemas que buscan simular el comportamiento del cerebro humano [3].

En relación con el método de entrenamiento, los algoritmos de aprendizaje profundo pueden clasificarse en diferentes categorías, tales como aprendizaje supervisado, semi-supervisado, no supervisado y por refuerzo [13]. Asimismo, existen diversas arquitecturas de redes neuronales profundas, de las cuales se describen a continuación aquellas más relevantes para el desarrollo de este trabajo de investigación.

1.6 Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son modelos de DL, inspirado en el proceso biológico y el patrón de conectividad que llevan a cabo las neuronas de la corteza visual del cerebro [13]. Existen dos etapas de entrenamiento en la CNN: la *feed-forward*, donde la información fluye desde la entrada hasta la salida, y la *back-propagation*, que ajusta los pesos de la red en función del error observado en la salida en conjunto con el optimizador [14].

Las CNN están compuestas por una secuencia de capas, que incluyen: capas de convolución con activación de Unidad Lineal Rectificada (ReLU), capas de agrupamiento (*pooling*), capa totalmente conectada y, finalmente, una capa con activación Softmax. Esta estructura permite a las CNN aprender características jerárquicas en los datos. La estructura se puede ver en la Figura 1.2.

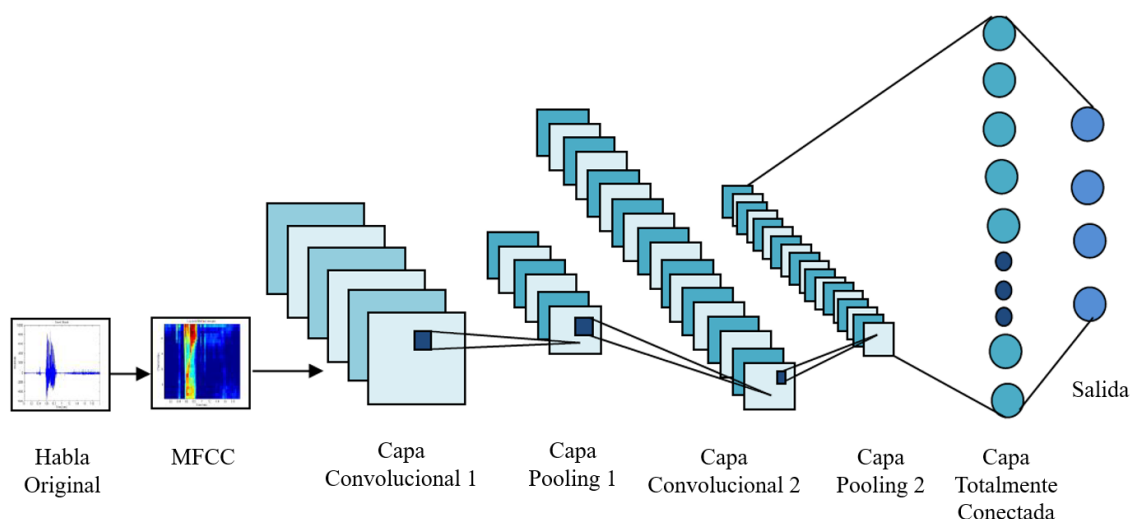


Figura 1.2. Ejemplo de una red neuronal convolucional.

1.7 Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN, por sus siglas en inglés) son un tipo de red neuronal con capacidad de retroalimentación, que se utiliza principalmente para el procesamiento de información secuencial y se caracteriza por la interconexión de entradas y salidas. El término *recurrente* se debe a que la red reutiliza el mismo conjunto de parámetros a lo largo de la secuencia, de manera que el estado interno en cada instante depende tanto de la entrada actual como del estado calculado en pasos temporales anteriores. Esta dependencia de estados previos requiere que las RNN posean una memoria para almacenar el conocimiento acumulado a lo largo de la secuencia [3].

Las RNN presentan una alta sensibilidad a la desaparición del gradiente, que es uno de los principales problemas que afectan su rendimiento [15]. Durante la fase de entrenamiento, los

gradientes pueden disminuir exponencialmente, limitando su capacidad para recordar estímulos iniciales. Aunque esta sensibilidad puede reducirse con el tiempo, llega a comprometer la efectividad del modelo. Para reducir este problema, se desarrolló el modelo Memoria a Corto y Largo Plazo, el cual se abordará a continuación. La Figura 1.3 muestra la estructura de este tipo de red.

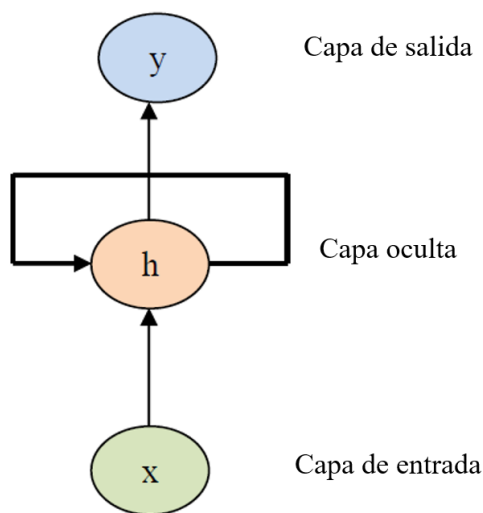


Figura 1.3. Estructura de una red neuronal recurrente básica con bucle.

1.7.1 Memoria a Corto y Largo Plazo

La arquitectura Memoria a Corto y Largo Plazo (LSTM, por sus siglas en inglés), se basa en la arquitectura de las RNN y fue creada específicamente para solucionar el problema de la desaparición del gradiente mediante la inclusión de más interacciones en la red. Esta arquitectura consta de tres puertas: la puerta de olvido, la puerta de entrada y la puerta de salida, junto con un estado de celda. La puerta de entrada se encarga de seleccionar qué información nueva se debe retener, la puerta de olvido determina qué datos de entradas pasadas deben ser descartados y la puerta de salida decide qué parte del estado de celda se debe emitir como salida. Gracias al funcionamiento de estas puertas es que la LSTM puede mantener las dependencias a largo plazo, porque puede olvidar y recordar la información, mientras relaciona conocimiento

del pasado con conocimiento del presente [4]. La estructura de un LSTM se muestra en la Figura 1.4.

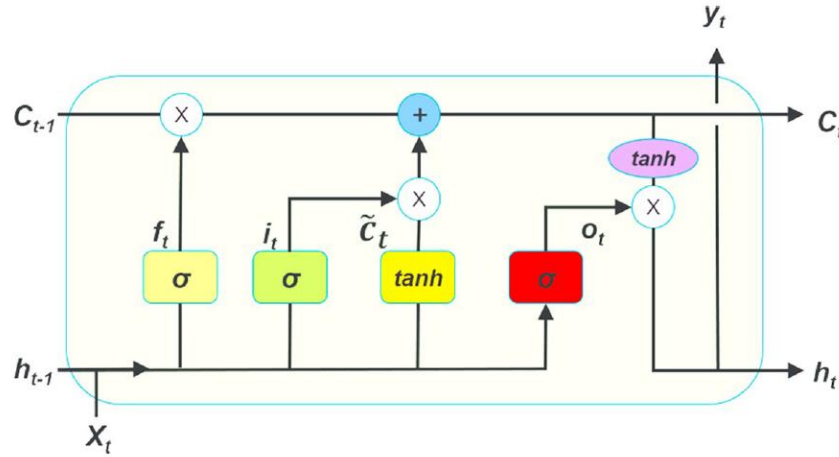


Figura 1.4. Estructura Long Short-Term Memory.

1.7.2 Memoria a Corto y Largo Plazo bidireccional

La Memoria a Corto y Largo Plazo bidireccional (BiLSTM, por sus siglas en inglés), en cuanto a arquitectura, comparte la misma estructura básica que una LSTM, pero con la diferencia clave de que procesa la información en dos direcciones: una hacia adelante (del pasado al futuro) y otra hacia atrás (del futuro al pasado) [6]. Esto se logra mediante la implementación de una técnica de cálculo dual que utiliza dos capas LSTM paralelas, lo que permite generar predicciones considerando tanto el contexto previo como el posterior [16]. Esta arquitectura cuenta con la capacidad de manejar la información de forma bidireccional mejorando significativamente la comprensión de relaciones temporales complejas dentro de una secuencia, permitiendo que la red mantenga presente información de dependencias pasadas y futuras en cada instante de tiempo, lo cual es especialmente útil en tareas el procesamiento automático del habla.

1.8 Transformers

Los Transformers se han consolidado como un modelo de aprendizaje profundo fundamental en tareas relacionadas con el procesamiento del lenguaje natural (NLP, por sus siglas en inglés) y el procesamiento del habla. Como consecuencia, muchos investigadores han explorado su potencial para resolver tareas de SER y han incorporado este modelo en sus estudios [14], [4], [17], [18].

Un Transformer se compone esencialmente de un codificador y un decodificador, ambos basados en mecanismos de atención. Cada bloque del codificador contiene dos subcapas: un mecanismo de auto-atención multi-cabeza (MHSA, por sus siglas en inglés) y una red totalmente conectada hacia delante (FFN, por sus siglas en inglés), la arquitectura habitual suele ser la del perceptrón multicapa.

Por otro lado, los bloques del decodificador añaden un módulo adicional de atención cruzada, que conecta las salidas del MHSA del decodificador con las representaciones generadas por el codificador. Este mecanismo permite que el modelo integre información tanto de la entrada como del contexto generado por el propio proceso de decodificación [19]. La arquitectura se puede observar en la Figura 1.5.

Es importante señalar que los datos de entrada de un Transformer requieren un proceso de tokenización, en el cual la información se divide en unidades más pequeñas (tokens) para facilitar su análisis. Posteriormente, estos tokens son transformados en vectores mediante un módulo de embebido (*embedding*), lo que permite a la información ser interpretada por la red neuronal. Además, es necesario añadir información de posición mediante codificaciones posicionales, ya que la arquitectura del Transformer no posee un mecanismo inherente para capturar el orden secuencial de los datos.

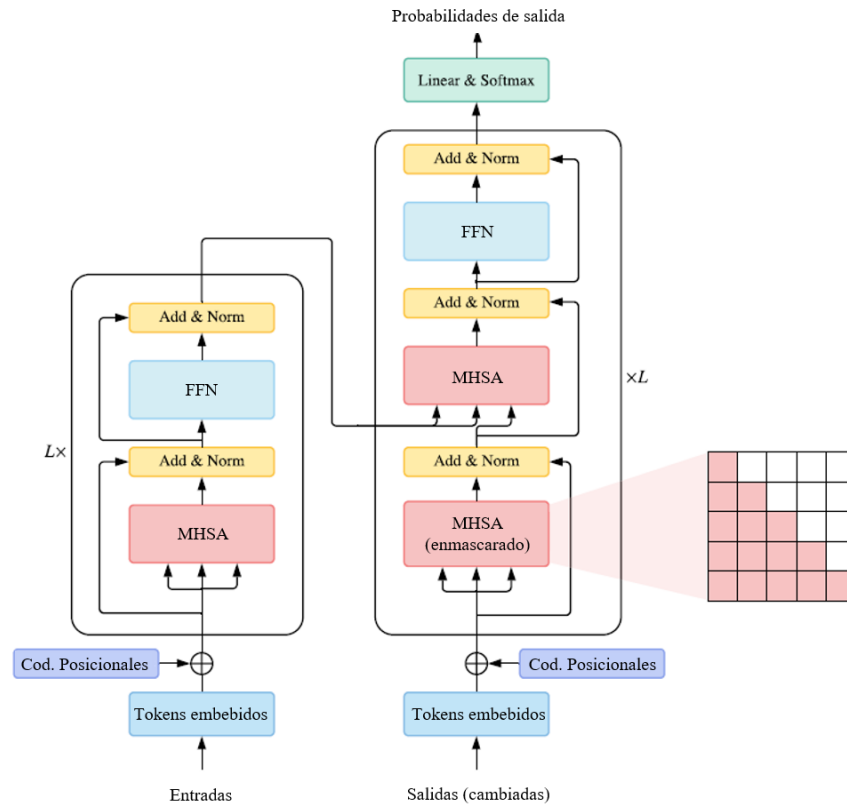


Figura 1.5. Estructura básica de un Transformer.

1.9 Métricas de evaluación de desempeño

Dentro de las tareas de clasificación en aprendizaje automático y aprendizaje profundo se emplean métricas para evaluar el desempeño de los modelos. Estas métricas permiten cuantificar y comparar de manera objetiva los resultados obtenidos, lo cual facilita establecer conclusiones cuantitativas respecto a la capacidad de un modelo para cumplir con la tarea de clasificación.

La mayoría de estas métricas se definen a partir de la matriz de confusión, la cual resume el desempeño del modelo en términos de aciertos y errores en las predicciones [20]. A partir de ella se obtienen los siguientes elementos fundamentales:

- VP (verdaderos positivos): número de instancias correctamente clasificadas como positivas.
- VN (verdaderos negativos): número de instancias correctamente clasificadas como negativas.
- FP (falsos positivos): número de instancias incorrectamente clasificadas como positivas.
- FN (falsos negativos): número de instancias incorrectamente clasificadas como negativas.

1.9.1 Exactitud

La exactitud (conocida como *accuracy*) define la cantidad de predicciones clasificadas correctamente en relación a todas las predicciones.

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \quad (1.4)$$

1.9.2 Precisión

Se define como precisión (*precision*) a la relación que hay entre las predicciones verdaderas positivas y todas las predicciones que se definieron como positivas, este es relacionado con el error Tipo I [20].

$$Precision = \frac{VP}{VP + FP} \quad (1.5)$$

1.9.3 Sensibilidad

Se le conoce como sensibilidad (*recall*) a la relación que hay entre las predicciones verdaderas positivas y todas aquellas predicciones que debieron ser positivas, esta métrica se relaciona al error Tipo II [20].

$$Recall = \frac{VP}{VP + FN} \quad (1.6)$$

1.9.4 Medida-F1

La medida-F1 es una métrica compuesta que representa la media armónica entre la sensibilidad y la precisión. De esta manera, otorga información proporcional entre ambas métricas.

$$\text{Medida} - \text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (1.7)$$

1.9.5 Métrica de exactitud balanceada

La métrica de exactitud balanceada (BAM, por sus siglas en inglés) es una métrica adecuada para bases de datos multiclase con clases desbalanceadas, ya que asigna el mismo peso a cada clase independientemente de su número de muestras. Consiste en obtener un promediado de las sensibilidades de cada clase.

$$\text{BAM} = \frac{\sum_{c=1}^C \frac{VP(c)}{VP(c) + FN(c)}}{|C|} \quad (1.8)$$

Donde c corresponde al índice de la clase y C representa el número total de clases.

II.ANTECEDENTES

Este capítulo presenta los antecedentes considerados para el desarrollo de la tesis. El capítulo se organiza de la siguiente manera: en la sección 2.1 se describen los antecedentes básicos necesarios para entender el desarrollo de un sistema SER, la sección 2.2 aborda conjuntos de datos de habla emocional en español, la sección 2.3 revisa algunos trabajos relevantes en los que se han desarrollado sistemas SER y la sección 2.4 hace un resumen del estado del arte revisado.

2.1 ¿Cómo desarrollar un sistema de reconocimiento de emociones por medio del habla?

Mohammed y Abbas [3] realizaron una recopilación de los aspectos clave que deben considerarse para desarrollar un sistema de reconocimiento de emociones por medio del habla (SER). Estos aspectos se presentan de manera esquemática en la Figura 2.1.

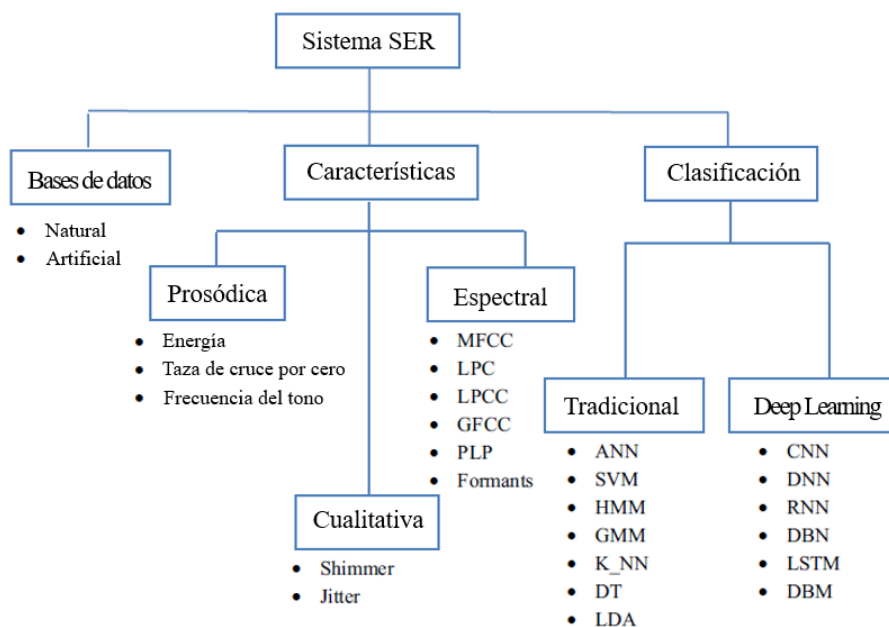


Figura 2.1. Principales características de un sistema SER.

El artículo comienza explicando que los conjuntos de datos utilizados en SER se dividen en dos tipos: artificiales y naturales. Asimismo, entre los conjuntos de datos disponibles en la literatura existen aquellos populares y privados. Por otro lado, hay distintos tipos de características que se pueden extraer de una señal de audio y es de vital importancia el saber seleccionar las características adecuadas para un reconocimiento preciso de las emociones. Estas características se dividen principalmente en tres categorías: prosódicas, cualitativas y espectrales.

Adicionalmente, se revisan los métodos de clasificación más empleados en sistemas SER, diferenciando entre enfoques tradicionales de aprendizaje automático y métodos basados en aprendizaje profundo. Finalmente, se identifican diversos retos asociados al desarrollo de estos sistemas, entre los que sobresalen la limitada disponibilidad de conjuntos de datos de habla emocional natural y la ambigüedad inherente a la expresión emocional, factores que dificultan la definición de características discriminativas. En este contexto, se resalta la importancia de considerar el idioma como un elemento crítico en el diseño de modelos SER, aunque también se mencionan enfoques multilingües como una posible alternativa para enfrentar dichas limitaciones.

2.2 Conjuntos de datos de habla emocional en español

García-Cuesta et al. [1] crearon EmoSpanishDB, un conjunto de datos de habla emocional en español compuesto de 4,200 archivos de audio con voces en español generadas por 50 sujetos no-actores. Se utilizó una técnica de inducción de emociones para que los participantes expresaran las seis emociones básicas definidas por Ekman: ira, disgusto, miedo, alegría, tristeza y sorpresa; además de un tono neutral [21]. Como resultado, obtuvieron un conjunto de 4200 archivos de audio. Posteriormente, se empleó un proceso de etiquetado mediante colaboración abierta (*crowdsourcing*), en el cual un grupo de anotadores humanos asignó la emoción correspondiente a cada archivo de audio a partir de un criterio de consenso. Este

procedimiento permitió generar una versión reducida y depurada del conjunto de datos, denominada EmoMatchSpanishDB (EmoMatch), con 2020 archivos.

Una vez desarrollada EmoMatch, los autores probaron su utilidad mediante diferentes algoritmos de aprendizaje automático (ML), como XGBoost, la Máquina de Vectores de Soporte (SVM) y una red neuronal de propagación hacia adelante (FFNN). Estos modelos de ML obtuvieron valores en exactitud de 0.565, 0.643 y 0.571 respectivamente utilizando EmoMatchSpanishDB; mientras que al utilizar EmoSpanishDB obtuvieron 0.524, 0.563 y 0.541 respectivamente. Por medio de la exactitud obtenida se puede demostrar una mejora considerable al utilizar el conjunto de datos EmoMatchSpanishD. En base a estos resultados se concluye que EmoMatch es un conjunto de datos viable para desarrollar sistemas SER y se espera una mejora considerable de desempeño al utilizar métodos de DL.

Pan et al. [22] desarrollaron Spanish MEACorpus 2023 un nuevo conjunto de datos para el reconocimiento de emociones multimodal en español el cual contiene 13.16 horas de habla repartidas entre 5129 archivos de audio los cuales están etiquetados bajo la distinción de las seis emociones básicas de Ekman [21]. Se menciona como a pesar de que el lenguaje español es uno de los más hablados en el mundo, hay una falta de recursos para desempeñar tareas de SER en este idioma; siendo esta la motivación principal por la cual fue considerado necesario el desarrollar este conjunto de datos. El enfoque utilizado para la creación del conjunto de datos consistió en buscar una ambientación natural del mundo, por lo tanto, se recopilaron audios en español extraídos de videos en YouTube; el conjunto fue validado por medio de técnicas multimodales de aprendizaje profundo fusionando texto y audio. Los modelos que destacan los autores en relación a los resultados obtenidos son W2V-BERT para el audio y MarIA para el texto; en los cuales obtuvieron un valor de medida-F1 de 84.48% y 69.389% respectivamente. Por otro lado, el enfoque multimodal basado en fusión tardía con concatenación mejoró los resultados obteniendo 87.74% en medida-F1. Concluyendo que las características del audio y

del texto se complementan mutuamente para lograr un mejor resultado. En la Figura 2.3 se encuentra el flujo de trabajo recomendado para Spanish MEACorpus 2023.

2.3 Reconocimiento de emociones por medio del habla en el estado del arte

Atmaja y Sasou [23] utilizaron la base de datos EmoFilm para analizar el reconocimiento de emociones por medio del habla (SER) en configuraciones multilingües, interlingüísticas y monolingües. Los autores buscan ofrecer una alternativa a los sistemas tradicionales de SER, que en su mayoría han sido diseñados para configuraciones monolingües. Para ello, emplean EmoFilm [24], una base de datos que contiene 1115 muestras de audio en inglés (EN), italiano (IT) y español (SP), extraídas de 43 películas y 207 hablantes distintos. Es importante destacar que las películas originalmente se encuentran en inglés, mientras que las versiones en italiano y español son versiones dobladas. Esta base de datos clasifica las emociones en cinco categorías: miedo, desprecio, alegría, ira y tristeza.

Atmaja y Sasou utilizaron su trabajo para definir tres aproximaciones distintas para entrenar y validar un sistema SER: multilingüe, interlingüístico y monolingüe. El reconocimiento de emociones multilingüe implica entrenar el sistema con los tres idiomas mezclados y evaluarlo en conjuntos que contienen estos idiomas. El enfoque interlingüístico, por otra parte, consiste en entrenar el sistema en un idioma y evaluarlo con otro diferente. Finalmente, el reconocimiento monolingüe se refiere a entrenar y evaluar el sistema en el mismo idioma. El modelo que se utilizó fue SVM para clasificación bajo los tres esquemas de experimentación anteriormente mencionados. Bajo el esquema multilingüe se logró una exactitud promedio del 74.86% en todos los lenguajes, el interlingüístico obtuvo una exactitud promedio del 59.58% y el enfoque monolingüe consiguió una exactitud promedio del 72.17%. Los autores concluyeron que no hay un beneficio significativo en utilizar un modelo monolingüe para SER, ya que los modelos interlingüísticos alcanzan resultados similares. Además, los modelos multilingües demostraron un mejor rendimiento, aunque los autores atribuyen esta mejora al mayor número de muestras utilizadas durante el entrenamiento en esta configuración.

Cabrera-Rodriguez y García Cuesta [25] desarrollaron un sistema de identificación de emociones basado en redes neuronales multiobjetivo, utilizando el conjunto de datos EmoMatchSpanishDb [18]. El objetivo del sistema es identificar tanto la emoción transmitida como reconocer a la persona que emite el mensaje. El sistema busca separar las características personales del hablante de las características del discurso, promoviendo cierta independencia del emisor para mejorar la generalización del modelo. Se desarrollaron principalmente dos modelos multiobjetivo: un perceptrón multicapa (MLP) y otro que combina capas de red neuronal convolucional con una capa unidireccional de LSTM (CNN-LSTM). Los autores concluyeron que añadir el reconocimiento de hablante a la tarea de reconocimiento de emociones no produjo una mejora significativa en la precisión de los resultados. Sin embargo, este trabajo es relevante para la presente investigación, ya que establece un precedente sobre el uso de EmoMatchSpanishDB.

Andayani et al. [4] proponen un modelo híbrido para el reconocimiento de emociones a partir de archivos de audio. El modelo consiste en una red LSTM que reemplaza la codificación posicional de la arquitectura Transformer, junto con una capa Transformer que actúa como Autoencoder para aprender las dependencias a largo plazo en el habla. Este enfoque aprovecha el mecanismo de auto-atención multi-cabeza de los Transformer para capturar dichas dependencias. El modelo fue evaluado utilizando tres bases de datos ampliamente reconocidas en la literatura: RAVDESS un conjunto de datos de habla emocional en inglés, Emo-DB un conjunto de datos de habla emocional en alemán y un conjunto de datos que combina las dos primeras. Para el entrenamiento, se utilizaron los coeficientes cepstrales de frecuencias Mel, que fueron extraídos de los espectrogramas. Este modelo híbrido superó a otros modelos del estado del arte, alcanzando tasas de reconocimiento de emociones del 75.62%, 85.55% y 72.49% para cada una de las bases de datos, respectivamente.

Una aproximación similar fue seguida por Tang et al. [14], quienes proponen un modelo de CNN-Transformer con mecanismos de atención multidimensional. Esta arquitectura modela

tanto la información local como la global en diferentes niveles de granularidad en el habla, capturando dependencias temporales, espaciales y de canal. El modelo utiliza bloques CNN para capturar la información en el dominio tiempo-frecuencia, seguido de una red basada en mecanismos de atención denominada T-Sa. Adicionalmente, la arquitectura incorpora unidades Shuffle, cuyo propósito es mejorar la fusión entre las dimensiones espacial y de canal mediante la redistribución de características entre grupos de canales. Finalmente, el módulo LCT (*Light Convolutional Transformer*) integra la información local con las dependencias a largo plazo usando convoluciones ligeras y mecanismos de atención por coordenadas en la auto-atención multi-cabeza. Este modelo fue evaluado en las bases de datos IEMOCAP y Emo-DB, correspondientes a los idiomas inglés y alemán, obteniendo mejores resultados que otros trabajos previos en el estado del arte con las mismas bases de datos.

Siguiendo con el esquema de modelos basados en arquitecturas híbridas, se encuentra el modelo desarrollado por Mohammed et al. [17], que consiste en un sistema robusto y ligero que utiliza una red de fusión de doble flujo CNN-Transformer (DS-CTFN), dicho modelo fue diseñado para capturar de manera eficaz las características espaciales (por medio de la CNN) y temporales (por medio del Transformer) presentes en los datos de audio crudos. Por medio de este enfoque, es posible el explorar los MFCC y los espectrogramas Mel en forma paralela para extraer información de distintas perspectivas acústicas. Se utilizaron los conjuntos de datos: RAVDESS, EMO-DB y TESS para entrenar la arquitectura propuesta, las cuales se encuentran en inglés y alemán. Con dichos conjuntos de datos se obtuvieron valores en exactitud de 97.64%, 97.53% y 99.42 respectivamente, obteniendo resultados que superan modelos populares en el estado del arte.

2.4 Resumen del estado del arte

Los elementos utilizados para el desarrollo de modelos de reconocimiento de emociones por medio del habla en el estado del arte se sintetizan en la Tabla 2.1, donde se destacan las características acústicas, las arquitecturas empleadas y los conjuntos de datos utilizados. A

partir de esta síntesis, se observa una clara tendencia al uso de representaciones espectrales como los espectrogramas Mel y los MFCC. Asimismo, en cuanto a las arquitecturas, predomina el desarrollo de modelos híbridos que combinan distintos tipos de redes, así como el uso creciente de arquitecturas basadas en mecanismos de atención y Transformers.

Tabla 2.1. Información sintetizada de modelos de SER en el estado del arte.

Artículo	Características utilizadas	Arquitectura	Conjunto de datos
[23]	Representaciones de wav2vec2.0	SVM	EmoFilm
[25]	eGeMAPS, COMPARE y Espectrogramas Mel	Perceptrón Multicapa y CNN-LSTM	EmoMatchSpanishSB y Emo-DB
[4]	MFCC	LSTM-Transformer	RAVDESS, Emo-DB y language-independent dataset
[14]	MFCC	CNN-Transformer con mecanismos de atención multidimensional	IEMOCAP y Emo-DB
[17]	MFCC Espectrogramas Mel	CNN-Transformer	RAVDESS, TESS y Emo-DB

III. CONJUNTOS DE DATOS

Este capítulo describe de manera detallada y técnica los conjuntos de datos empleados a lo largo del presente trabajo de investigación. En la Sección 3.1 se presenta el conjunto EmoMatchSpanishDB y en la Sección 3.2 se describe el conjunto Spanish MEACorpus 2023.

3.1 EmoMatchSpanishDB

EmoMatchSpanishDB (EmoMatch) es un subconjunto derivado de EmoSpanishDB, el cual se generó a partir de las grabaciones de 50 participantes que expresaron 12 oraciones distintas en siete ocasiones cada una, correspondientes a las seis emociones básicas de Ekman (ira, disgusto, miedo, alegría, sorpresa y tristeza), además de una emoción neutral. La muestra estuvo conformada por 30 hombres y 20 mujeres, con una edad promedio de 33.9 años y una desviación estándar de 10.08, lo que produjo un total de 4200 archivos de audio. Para garantizar la precisión en el etiquetado emocional, se aplicó un proceso de crowdsourcing en el que 194 hablantes nativos de español evaluaron las muestras mediante una prueba de percepción que combinó votación mayoritaria y una prueba binomial con un nivel de significancia de $p < 0.05$. Este procedimiento generó 21,490 tareas de etiquetado y permitió obtener consenso en 3550 audios, los cuales conformaron EmoSpanishDB. De ellos, 2,020 coincidieron con la emoción originalmente inducida, constituyendo así la base de datos EmoMatch. Este proceso descrito para la creación de ambos conjuntos de datos se puede ver gráficamente en la Figura 3.1.

De esta manera, EmoMatch es un conjunto de datos de habla emocional en español que se obtuvo a partir de inducir emociones en los hablantes para que posteriormente, digan ciertas oraciones que transmitan las emociones inducidas. El conjunto de datos cuenta con 2,020 archivos de audio, cada uno con un título que describe el número del sujeto hablante, la emoción inducida y el número de la oración. Por ejemplo: 01_Alegría_05.

Para obtener acceso a la base de datos, es necesario contactar al autor Esteban García-Cuesta y enviarle un acuerdo que está en un documento llamado “EMOMATCHAgreement”, el cual es proporcionado por él mismo. Dicho documento formaliza un acuerdo en el que el solicitante

se compromete a no comercializar la base de datos y a utilizarla exclusivamente con fines académicos. Una vez completado este procedimiento, se otorga acceso a los archivos de audio crudos de EmoMatchSpanishDB, que en realidad contiene 2,005 muestras de audio, de los 2,020 reportadas en el artículo.



Figura 3.2. Diagrama de flujo del proceso utilizado para la creación de EmoSpanishDB y EmoMatchSpanishDB.

3.2 Spanish MEACorpus 2023

Spanish MEACorpus 2023 (MEACorpus) es un corpus multimodal de habla y texto en español diseñado para el análisis de emociones en entornos naturales. A diferencia de otras bases de datos, como EmoMatch (que se obtiene en condiciones controladas), que se obtiene en condiciones controladas, este corpus captura interacciones espontáneas en escenarios reales, lo que proporciona una representación más auténtica y variada de las emociones humanas.

El proceso de construcción de MEACorpus consistió en compilar y anotar una serie de segmentos de audio extraídos de videos de YouTube. Para ello, primero se seleccionaron y depuraron los videos, luego se extrajo el audio y se segmentó automáticamente en función de los silencios. Posteriormente, se realizó la transcripción a texto. La anotación de los audios se llevó a cabo siguiendo la taxonomía de Ekman para las emociones y fue realizada por tres miembros del equipo de investigación debidamente capacitados. Una vez conformada el

conjunto de datos, se dividió en conjuntos independientes de entrenamiento (80%) y validación (20%).

Para la selección de los audios, se priorizaron canales de YouTube que evocaran emociones específicas. Por ejemplo, se utilizaron canales de política para capturar expresiones de disgusto, canales deportivos para registrar el enojo de los jugadores tras una derrota y canales de entretenimiento para obtener muestras de alegría. Es importante señalar que, debido a esta estrategia, cada grabación proviene de un entorno acústico diferente.

El corpus incluye un total de 13:16 horas de grabaciones, con 5129 muestras de audio etiquetadas en seis categorías emocionales: disgusto, enojo, alegría, tristeza, miedo y una emoción neutral. Las clases están desbalanceadas, siendo neutral y disgusto las categorías con mayor cantidad de muestras. La duración promedio de los audios es de 9.24 segundos. En la Figura 3.2 se observa el flujo de trabajo recomendado por los autores para el conjunto de datos.

Para obtener acceso a Spanish MEACorpus 2023, es necesario contactar a los autores del estudio para solicitar el acceso a los archivos de audio crudos.

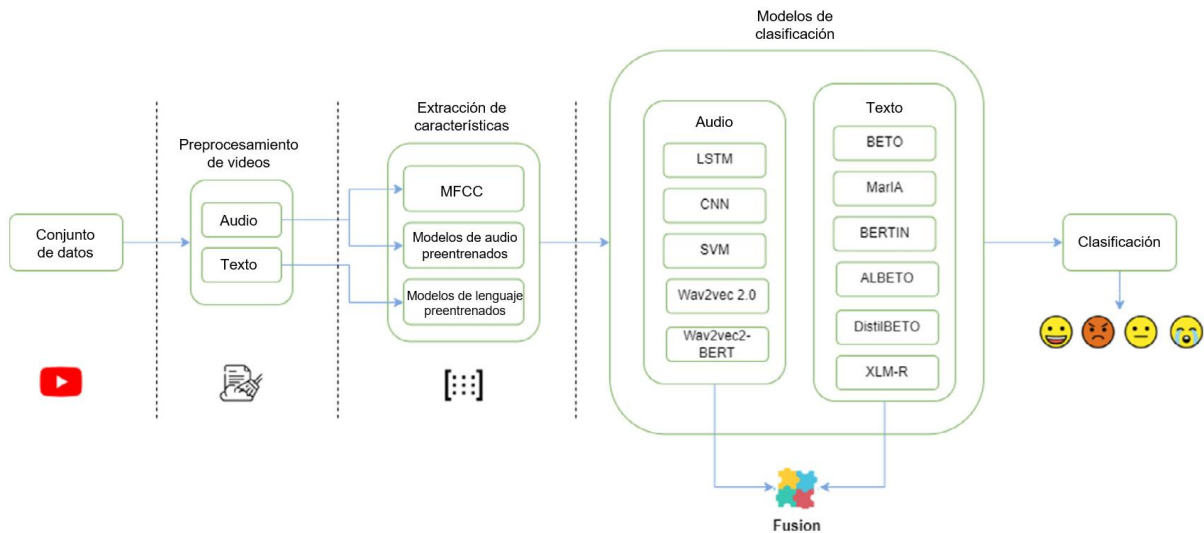


Figura 3.2. Flujo de trabajo recomendado para trabajar con Spanish MEACorpus 2023.

IV. EXPERIMENTACIÓN CON MODELOS POPULARES SER

En este capítulo se presentan diversos experimentos que consistieron en probar diversos modelos de aprendizaje profundo populares en SER para conocer su desempeño en conjuntos de datos de reconocimiento de emociones con habla en español.

El capítulo se organiza de la siguiente manera: en la Sección 4.1 se describe la metodología general empleada en los experimentos. La Sección 4.2 presenta los modelos de DL seleccionados para la tarea de reconocimiento de emociones. La Sección 4.3 detalla el preprocesamiento aplicado a los datos y la Sección 4.4 define los hiperparámetros utilizados durante el entrenamiento. Posteriormente, la Sección 4.5 expone los experimentos realizados con el conjunto de datos EmoMatchSpanishDB y la sección 4.6 describe los experimentos realizados con el conjunto de datos Spanish MEACorpus 2023. La sección 4.7 analiza los resultados obtenidos en ambos conjuntos de datos, la sección 4.8 define la experimentación con las variantes de ResNet, y la sección 4.9 documenta la experimentación con arquitecturas híbridas en configuraciones en serie y en paralelo. Por último, la sección 4.10 describe las arquitecturas alternativas al Transformer consideradas.

4.1 Metodología

Los experimentos realizados en este trabajo tienen como objetivo evaluar el desempeño de distintas arquitecturas de aprendizaje profundo en la tarea de reconocimiento de emociones a partir del habla en español. Para ello, se diseñó un esquema experimental que permitió comparar modelos de diferentes tipos bajo condiciones controladas, utilizando múltiples representaciones acústicas y conjuntos de datos de referencia.

En particular, se entrenaron y evaluaron modelos pertenecientes a tres tipos de arquitecturas: redes neuronales convolucionales, redes neuronales recurrentes y Transformers. En el caso de las CNN se experimentó con las arquitecturas ResNet50, EfficientNetB0, MobileNetV2, VGG16 e InceptionV3. Por parte con las RNN, se utilizaron redes con capas LSTM y BiLSTM.

Finalmente, para los modelos basados en Transformers, se empleó la arquitectura Vision Transformer (ViT).

Previo al entrenamiento de los modelos, se llevó a cabo un proceso experimental para determinar las características acústicas más adecuadas para cada tipo de arquitectura. Se concluyó que los espectrogramas log-Mel eran más apropiados para las CNN y Transformers, debido a la riqueza de información en el dominio frecuencia-tiempo que ofrecen. Por otro lado, para las RNN se optó por utilizar coeficientes cepstrales de frecuencias Mel (MFCC), ya que su naturaleza secuencial y estructura en *quefreny* por ventana de tiempo permite una mejor interpretación por parte de las redes recurrentes.

Se utilizaron las tres bases de datos descritas en el capítulo anterior para el proceso de experimentación, siendo estas: EmoMatchSpanishDB, Spanish MEACorpus 2023 y EmoFilm.

4.2 Modelos seleccionados

Los modelos que fueron seleccionados para los experimentos fueron cinco arquitecturas de CNN populares, un modelo LSTM, una red BiLSTM y un modelo Vision Transformer.

Para fines explicativos, se definió una simbología gráfica para describir las arquitecturas de los modelos empleados en los experimentos, la cual se presenta en la Figura 4.1. Dicha simbología consiste en el uso de bloques rectangulares para representar las capas y bloques principales que conforman cada arquitectura, mientras que los colores se utilizan únicamente con fines de diferenciación visual entre los distintos tipos de capas. Adicionalmente, los círculos representan funciones de activación u operaciones específicas, tales como funciones no lineales, regularización u operaciones aritméticas. Esta notación permite una visualización clara y consistente de las arquitecturas evaluadas a lo largo de este capítulo.

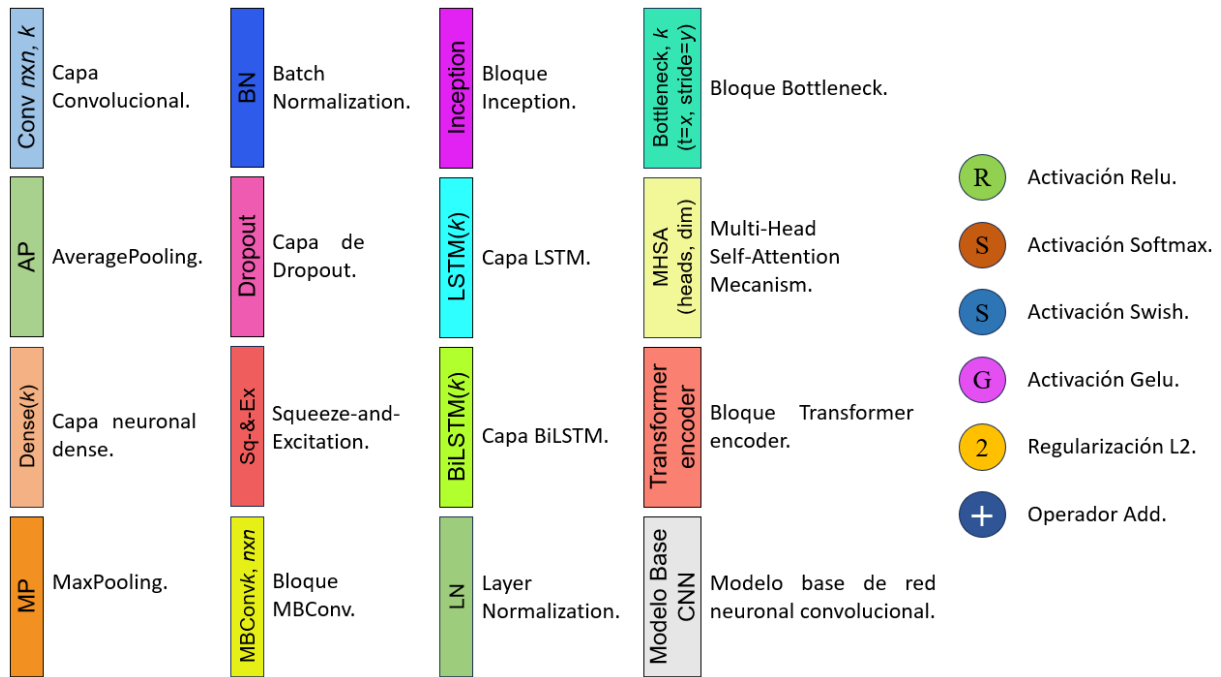


Figura 4.1. Simbología utilizada para describir los modelos.

4.2.1 ResNet50

ResNet integra en su arquitectura atajos de identidad (conexiones residuales), los cuales consisten en conexiones directas que permiten propagar las activaciones de capas previas sin modificación, y establece conexiones de salto que facilitan la reutilización de estas activaciones en capas más profundas. Estas conexiones facilitan el flujo del gradiente durante el entrenamiento, evitando la degradación del desempeño en redes profundas. Variantes como ResNet50 combinan conexiones residuales y tradicionales para aprender representaciones más complejas que si utilizara únicamente conexiones tradicionales, superando los desafíos que suelen surgir al entrenar redes muy profundas [26]. Adquiere el nombre de ResNet50 ya que cuenta con 50 capas a lo largo de la arquitectura; siendo estas las 49 capas convolucionales y la última capa totalmente conectada para la inferencia como se puede ver en la Figura 4.2. La arquitectura de ResNet50 ha demostrado ser altamente efectiva en tareas de SER debido a que es capaz de extraer características complejas de las señales de audio, mejorando así la precisión en la identificación del estado emocional [27].

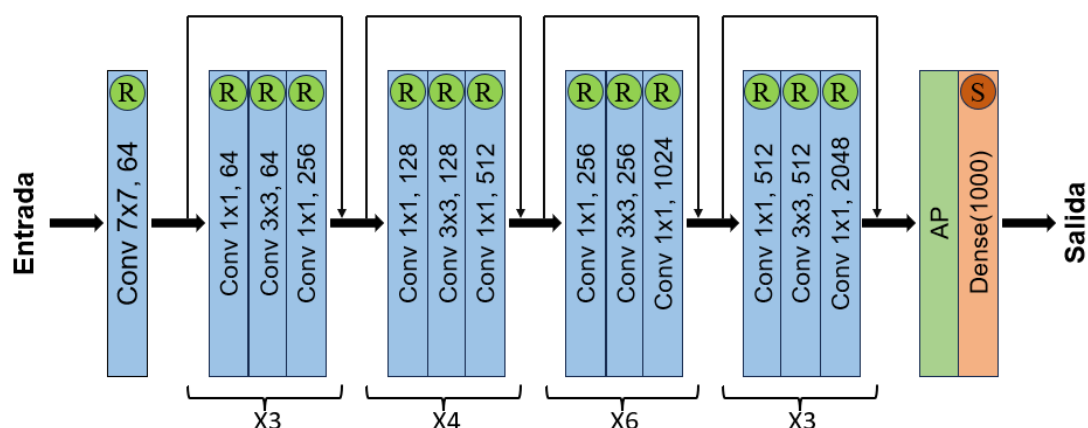


Figura 4.2. Arquitectura de ResNet50.

4.2.2 EfficientNetB0

EfficientNetB0 es una arquitectura de red neuronal convolucional diseñada para equilibrar precisión y eficiencia computacional. La arquitectura se muestra en la Figura 4.3, esta consta de 18 capas, entre las cuales, existe un bloque principal llamado *Mobile Inverted Bottleneck* (MBConv) que se pueden ver en la Figura 4.4. MBConv es una estructura que combina convoluciones por profundidad y bloques cuello de botella invertidos para optimizar la extracción de características. Además, incorpora módulos llamados Squeeze-and-Excitation, que recalibran dinámicamente la importancia de cada canal de características, y la activación Swish, que potencia la capacidad de aprendizaje del modelo. Su diseño escalable, que ajusta profundidad, ancho y resolución, la hace adecuada para aplicaciones en dispositivos con recursos limitados [28].

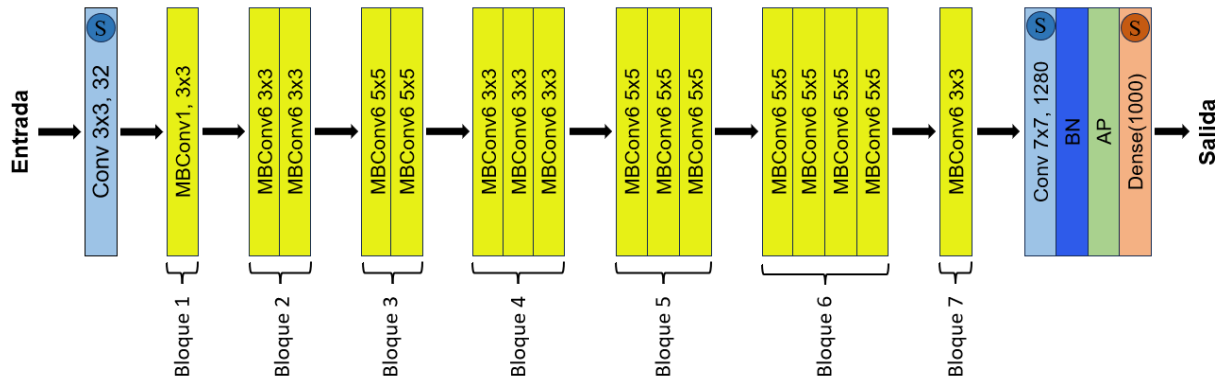


Figura 4.3. Arquitectura de EfficientNetB0.

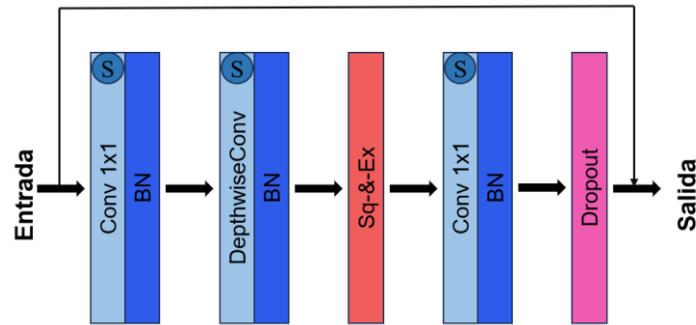


Figura 4.4. Arquitectura de los bloques MBConv.

4.2.3 MobileNetV2

MobileNetV2 es una arquitectura de red neuronal profunda diseñada específicamente para dispositivos con recursos limitados, con el objetivo de mejorar la eficiencia computacional y la precisión en tareas de visión por computadora. La aportación de su diseño se basa en dos innovaciones clave: los bloques residuales invertidos y los bloques de cuello de botella lineales. Los bloques residuales invertidos permiten una propagación más eficiente de la información a través de la red, mientras que los cuellos de botella lineales ayudan a preservar las características más relevantes de la entrada.

La arquitectura MobileNetV2 está compuesta por 17 bloques residuales invertidos, lo que da lugar a una profundidad de 53 capas al considerar todas las operaciones convolucionales que conforman la red, esto se puede ver en la Figura 4.5. Por medio de estas mejoras, MobileNetV2

reduce significativamente el número de parámetros y las operaciones computacionales en comparación con modelos anteriores, como MobileNetV1, logrando un equilibrio entre eficiencia y rendimiento en diversas aplicaciones de aprendizaje profundo [29].

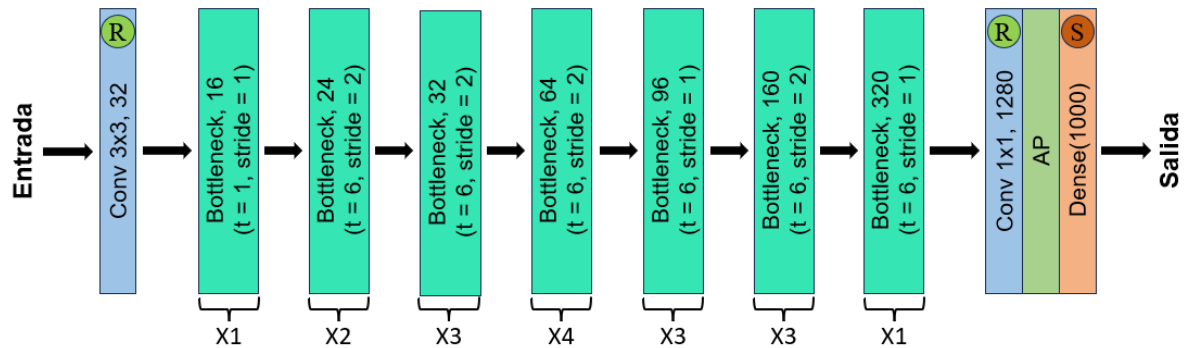


Figura 4.5. Arquitectura de MobileNetV2.

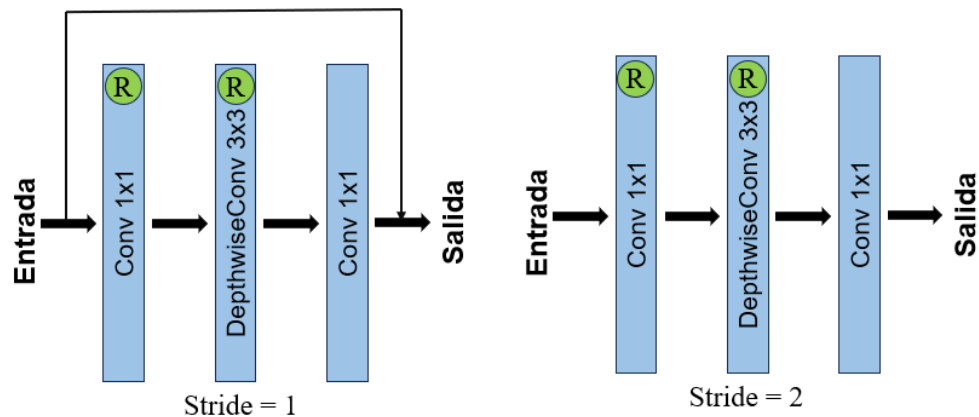


Figura 4.6. Bloques Bottleneck con Stride = 1 y Stride = 2.

4.2.4 InceptionV3

InceptionV3 es una arquitectura de red neuronal convolucional optimizada para lograr un equilibrio entre eficiencia computacional y alta precisión en tareas de clasificación de imágenes. Esta arquitectura está compuesta por 11 módulos Inception, organizados en tres variantes principales: Inception-A, Inception-B e Inception-C, como se muestra en la Figura 4.7.

InceptionV3 introduce la factorización de convoluciones, reemplazando filtros de mayor tamaño (5×5) por múltiples convoluciones más pequeñas, como 3×3 o combinaciones de 1×3 seguidas de 3×1 , lo que permite reducir el costo computacional sin afectar significativamente la capacidad de representación. Además, emplea reducción eficiente de la cuadrícula, combinando convoluciones y operaciones de agrupamiento en paralelo para disminuir la resolución espacial sin perder expresividad. También incorpora normalización por lote, clasificador auxiliar para mejorar la convergencia y suavizado de etiquetas para regularizar la salida del clasificador. Estas mejoras permiten a InceptionV3 alcanzar un mejor desempeño en el conjunto ImageNet con una menor cantidad de parámetros en comparación con arquitecturas previas, como VGG [30].

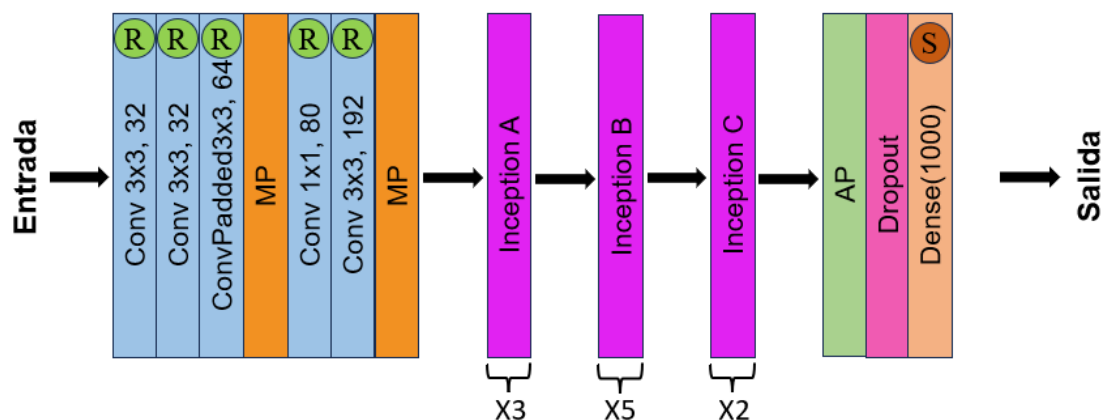


Figura 4.7. Arquitectura de InceptionV3.

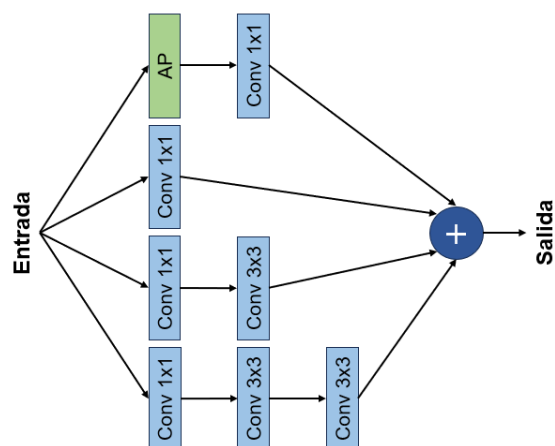


Figura 4.8. Bloque Inception-A.

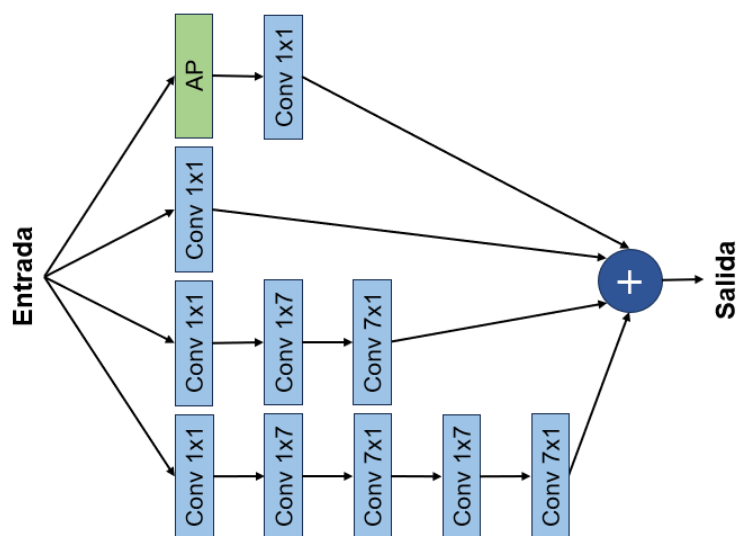


Figura 4.9. Bloque Inception-B.

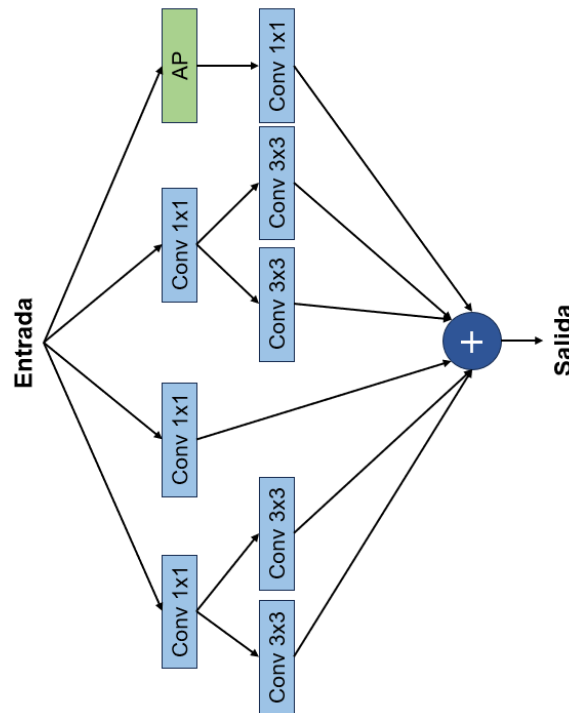


Figura 4.10. Bloque Inception-C.

4.2.5 VGG16

VGG16 es una arquitectura de red neuronal convolucional diseñada para el reconocimiento de imágenes a gran escala. Introducida en el contexto del desafío ILSVRC-2014, su principal contribución fue demostrar que aumentar la profundidad de la red con capas de convoluciones pequeñas (3×3) mejora significativamente la precisión en clasificación. La arquitectura VGG16 está compuesta por 16 capas con parámetros entrenables como se muestra en la Figura 4.11, organizadas en bloques de convolución y capas de agrupación por valor máximo, seguidas de tres capas totalmente conectadas y una capa de salida con activación Softmax. El modelo utiliza filtros pequeños con paso 1 y relleno adecuado para preservar la resolución, permitiendo capturar estructuras complejas en las imágenes con menos parámetros que convoluciones más grandes. Sus resultados en el conjunto ImageNet mostraron que redes más profundas pueden

lograr una mejor generalización, estableciendo las bases para arquitecturas modernas más avanzadas [31].

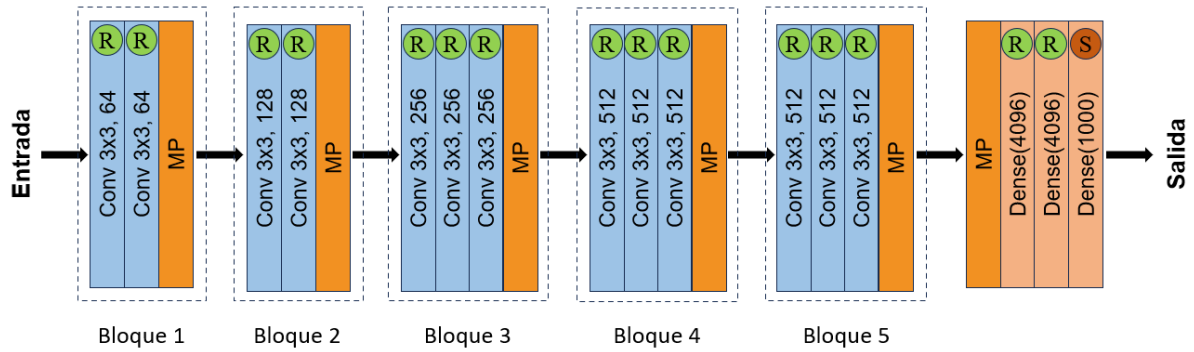


Figura 4.11. Arquitectura de VGG16.

4.2.6 Vision Transformer

El Vision Transformer (ViT) es un modelo basado en la arquitectura de Transformer, originalmente diseñada para procesamiento de lenguaje natural, que se aplica directamente al reconocimiento de imágenes sin necesidad de convoluciones. En lugar de utilizar filtros convolucionales, ViT divide la imagen en parches de tamaño fijo (por poner un ejemplo, 16x16 píxeles), los cuales se tratan como tokens en una secuencia y se proyectan a un espacio de características mediante una transformación lineal. A estos *embeddings* de parches se les agregan *embeddings* de posición para conservar la información espacial, y luego se procesan con un Transformer estándar compuesto por múltiples capas de auto-atención multi-cabeza y redes totalmente conectadas hacia adelante (FFNN) este diagrama de flujo se presenta en la Figura 4.12. Para la clasificación, el modelo introduce un token especial ([CLS]) cuyo estado final se usa como representación de la imagen [32].

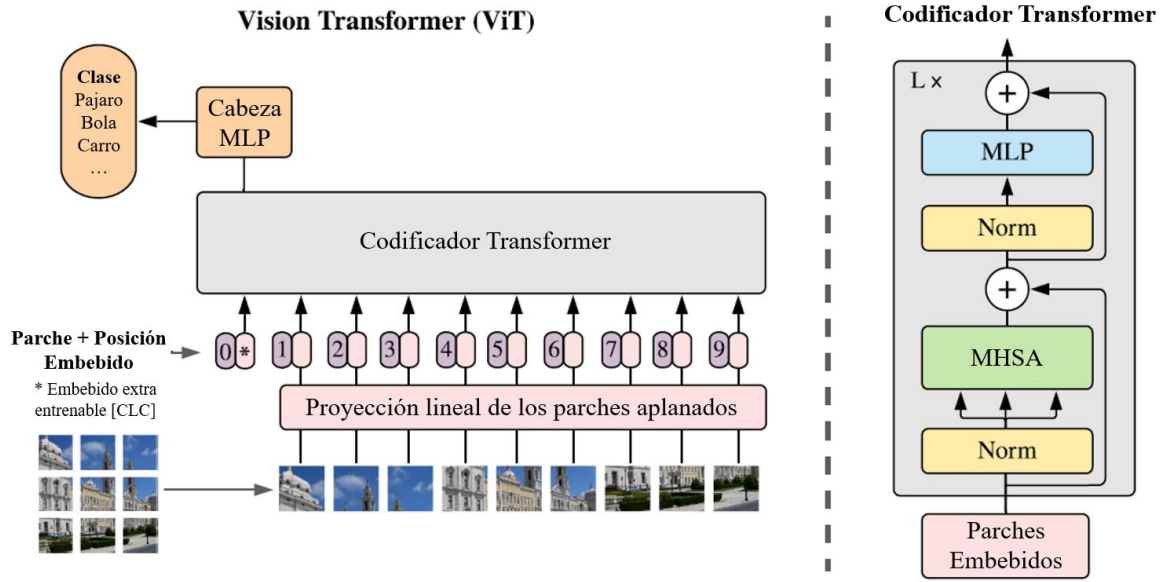


Figura 4.12. Arquitectura de Vision Transformer.

4.3 Preprocesamiento

Primero se obtuvieron los espectrogramas logarítmicos de Mel, S_{logMel} . Para ello, se procesaron los archivos de audio contenidos en cada base de datos, los cuales fueron submuestreados a la mitad de su frecuencia original, dependiendo del formato en que fueron provistos. Esta nueva frecuencia también se utilizó como frecuencia máxima (f_{max}) durante la extracción, junto con 128 bandas Mel, con el objetivo de capturar la distribución energética de la señal en el dominio tiempo-frecuencia. Posteriormente, el espectrograma Mel fue transformado a una escala logarítmica en decibeles, con el fin de mejorar la visibilidad de componentes de baja energía y reflejar mejor la percepción auditiva humana. Cada espectrograma fue guardado en formato png dentro de una carpeta específica, generando así representaciones visuales que capturan de forma efectiva las variaciones espectrales relevantes para el reconocimiento de emociones. Estas imágenes fueron utilizadas como entrada para el entrenamiento de las arquitecturas CNN y el ViT, tras un preprocesamiento adicional que consistió en: conversión a formato RGB para asegurar tres canales por imagen, normalización

de los valores de píxeles y redimensionamiento a 224×224 píxeles, de acuerdo con los requisitos de entrada de los modelos seleccionados.

Para las arquitecturas basadas en redes recurrentes, se implementaron dos esquemas distintos de extracción de características MFCC, adaptados a las características particulares de cada base de datos. En el caso de Spanish MEACorpus 2023, se estableció una longitud de 1000 cuadros, correspondiente a una duración de 10 segundos de audio por muestra que es similar a la duración promedio de los audios en dicha base de datos. Este valor se obtuvo utilizando una longitud de salto de 220 muestras, equivalente a 10 milisegundos por cuadro, y una frecuencia de muestreo de 22,000 Hz. Se extrajeron 13 coeficientes MFCC junto con sus primeras y segundas derivadas temporales (delta y delta-delta), obteniendo así 39 coeficientes por cuadro. Las secuencias de características se normalizaron en longitud mediante relleno con ceros para las más cortas y truncamiento para las más largas. Las etiquetas emocionales fueron obtenidas automáticamente a partir de un archivo con extensión .csv que las otorga. Por otro lado, para la base de datos EmoMatchSpanishDB, se optó por un esquema reducido de 400 cuadros, lo que equivale a 4 segundos de audio en relación a la duración promedio de los audios en esta base de datos, manteniendo las mismas condiciones de extracción (longitud de salto de 220 y frecuencia de muestreo de 22,000 Hz). Esta decisión se tomó considerando que la mayoría de los audios en esta base de datos presentan una duración más breve. Se siguió el mismo procedimiento de extracción de características, así como los mecanismos de rellenado y truncamientos mencionados anteriormente. Ambos esquemas permitieron generar tensores de entrada uniformes y adecuados para el entrenamiento de modelos con capas LSTM y BiLSTM, garantizando compatibilidad estructural entre los datos y respetando las particularidades de cada conjunto.

4.4 Hiperparámetros de entrenamiento para los experimentos

Para garantizar una comparación justa entre las diferentes arquitecturas, se definieron ciertos hiperparámetros de entrenamiento específicos para cada tipo de modelo, respetando las particularidades de sus arquitecturas y requisitos computacionales. Todos los modelos fueron

entrenados utilizando un máximo de 300 épocas y paro temprano para prevenir el sobreajuste. Esta técnica de paro temprano consiste en monitorear la métrica de pérdida de validación y cuando no se mejora en determinada cantidad de épocas, se detiene el entrenamiento.

La función de pérdida utilizada en todos los experimentos fue la entropía cruzada categórica dispersa, la cual está dada por:

$$L = -\frac{1}{N} \sum_{i=1}^N \log(p_{y_i}) \quad (4.1)$$

donde N es el número total de muestras, y_i representa la etiqueta real asociada a la muestra i y p_{y_i} corresponde a la probabilidad predicha por el modelo para la clase correcta. La entropía cruzada categórica dispersa es adecuada para tareas de clasificación multiclase con etiquetas enteras, ya que es consistente con una capa de salida con activación Softmax y evita la necesidad de codificación binaria excluyente.

El optimizador utilizado para el entrenamiento de las redes neuronales convolucionales y recurrentes fue el optimizador de momento adaptivo (Adam), cuya actualización de parámetros se expresa como:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (4.2)$$

donde θ_t representa los parámetros del modelo en la iteración t , α es la tasa de aprendizaje, $\hat{m}_t$ y $\hat{v}_t$ son las estimaciones corregidas del primer y segundo momento del gradiente, respectivamente, y ϵ es una constante pequeña para garantizar estabilidad numérica. Este optimizador se seleccionó por su rápida convergencia y buen desempeño empírico en redes profundas.

Para el caso del modelo ViT, se utilizó el optimizador AdamW, una variante de Adam que incorpora decaimiento de pesos desacoplado, lo cual mejora la regularización y la capacidad de generalización en arquitecturas basadas en Transformers.

Para los modelos basados en redes neuronales recurrentes (LSTM y BiLSTM) se utilizaron los siguientes hiperparámetros:

- Tamaño de lote: 128
- Tasa de aprendizaje: 0.0001
- Optimizador: Adam
- Épocas máximas: 300

Para los modelos basados en redes convolucionales (VGG16, ResNet50, InceptionV3, EfficientNetB0, MobileNetV2) se utilizaron los siguientes hiperparámetros:

- Tamaño de lote: 32
- Tasa de aprendizaje: 0.0001
- Optimizador: Adam
- Épocas máximas: 300

Para el modelo *Vision Transformer* (ViT) se utilizaron los siguientes hiperparámetros:

- Tamaño de lote: 32
- Tasa de aprendizaje: 0.00001
- Decaimiento de pesos: 0.0001
- Optimizador: AdamW
- Épocas máximas: 300

De manera general, los hiperparámetros fueron seleccionados a partir de pruebas preliminares y de valores comúnmente reportados en la literatura para tareas de reconocimiento de emociones a partir del habla. Estas configuraciones permitieron equilibrar la eficiencia del entrenamiento y el rendimiento general de los modelos, adaptando el proceso a las características computacionales de cada arquitectura y al volumen de datos procesado, y manteniéndose constantes dentro de cada tipo de modelo para garantizar condiciones de comparación homogéneas.

4.5 Experimentos con EmoMatchSpanishDB

Una vez que se llevó a cabo el preprocesamiento de los archivos de audio crudos en EmoMatch, se realizó la partición del conjunto en datos de entrenamiento (80%) y de validación (20%). La partición se realizó de forma estratificada para mantener la proporción original de las clases emocionales, esto debido al desbalance de clases en la base de datos.

Para los entrenamientos de los modelos CNN, se eliminó la capa final original de clasificación y se reemplazó por una estructura personalizada. Esta estructura consistió en la CNN base conectada con una capa de GlobalAveragePooling2D, la cual esta seguida de una capa densa de 128 neuronas con función de activación ReLU y un regularizador L2 con valor de 0.1. Posteriormente, se aplicó una capa de normalización por lotes (*BatchNormalization*), un Dropout con tasa de 0.4, y finalmente una capa densa de salida con 7 neuronas y función de activación Softmax, encargada de generar la probabilidad correspondiente a cada una de las siete clases emocionales presentes en la base de datos, en la Figura 4.13 se puede ver esta configuración descrita.

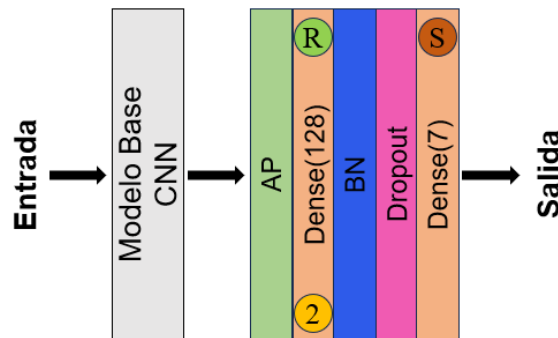


Figura 4.13. Arquitectura general de experimentos con CNNs para EmoMatchSpanishDB.

Para el caso de las RNNs, se utilizaron dos arquitecturas: una conformada por capas LSTM (Figura 4.14) y otra por capas BiLSTM (Figura 4.15). Ambas arquitecturas comparten la misma estructura general, diferenciándose únicamente en el tipo de capa recurrente utilizada.

Cada arquitectura está compuesta por tres capas secuenciales de LSTM o BiLSTM con 256, 128 y 64 unidades, respectivamente. Después de cada capa recurrente, se aplica una normalización por capas (*LayerNormalization*) y una capa de Dropout para evitar sobreajuste, con tasas de 0.4, 0.3 y 0.3, respectivamente. Además, las capas intermedias de 128 y 64 unidades incluyen una regularización L2 igualmente con el fin de evitar sobreajuste con valor de 0.01. Finalmente, el modelo termina con una capa densa de salida compuesta por 7 neuronas y una función de activación Softmax.

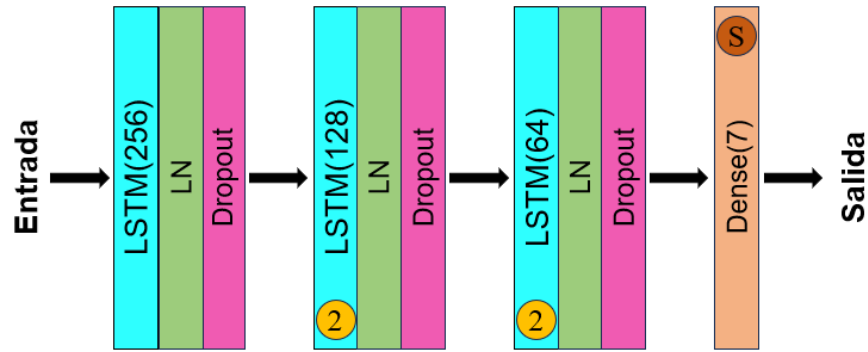


Figura 4.14. Arquitectura con capas LSTM para EmoMatchSpanishDB.

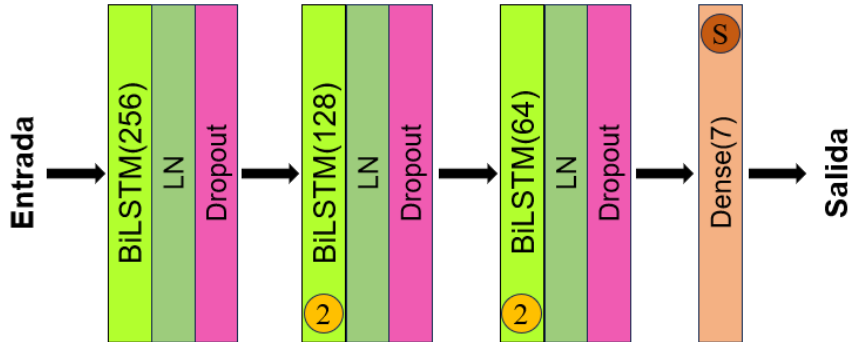


Figura 4.15. Arquitectura con capas BiLSTM para EmoMatchSpanishDB.

Para el ViT, se utilizaron como entrada tensores con dimensiones de $224 \times 224 \times 3$, correspondientes a los espectrogramas log-Mel generados en la etapa de preprocesamiento como se había descrito en secciones anteriores. El tamaño del parche fue configurado en 16×16 píxeles, lo que dio como resultado un total de 196 parches por imagen. Cada parche fue

proyectado a un espacio de características de 64 dimensiones (*projection dimension*), y el módulo Transformer empleó 8 cabezas de atención.

El bloque Transformer estuvo conformado por 8 capas, y en cada una se utilizaron unidades de red totalmente conectada internas de 128 y 64 nodos. Posteriormente, se aplicó una capa MLP como cabeza de clasificación, con dos capas densas de 512 y 256 neuronas. Finalmente, se añadió una capa densa con 7 neuronas y función de activación Softmax.

Con estos experimentos, se completó la evaluación de modelos sobre el conjunto de datos EmoMatchSpanishDB.

4.6 Experimentos con Spanish MEACorpus 2023

Para esta etapa experimental, se utilizaron los mismos modelos previamente descritos en redes neuronales convolucionales (ResNet50, EfficientNetB0, MobileNetV2, VGG16 e InceptionV3) y el modelo ViT. Las arquitecturas, los hiperparámetros de entrenamiento y las estrategias de regularización aplicadas se mantuvieron igual con respecto a EmoMatch, siguiendo los criterios establecidos en las secciones anteriores.

Los únicos modelos que sufrieron modificaciones respecto a los utilizados previamente fueron los modelos basados en redes neuronales recurrentes (LSTM y BiLSTM). El ajuste consistió en aumentar la complejidad de las arquitecturas, añadiendo una capa adicional a cada modelo. Esta decisión se tomó debido a que el conjunto de datos MEACorpus es aproximadamente tres veces mayor que EmoMatch, lo que ocasionaba que las arquitecturas originales presentaran síntomas de subajuste. En la Figura 4.16 se muestra la arquitectura utilizada para el modelo LSTM, mientras que la Figura 4.17 presenta la correspondiente al modelo BiLSTM.

Ambas arquitecturas están compuestas por cuatro capas recurrentes con 512, 256, 128 y 64 unidades, respectivamente, cada una seguida de una capa de normalización por capas y una capa de Dropout con tasas de 0.4, 0.3, 0.3 y 0.3 respectivamente. Además, las capas de 256, 128 y 64 unidades incluyen regularización L2 con valor 0.01. Finalmente, se empleó una capa

densa de salida con 7 unidades y activación Softmax, utilizada para realizar la clasificación multiclase.

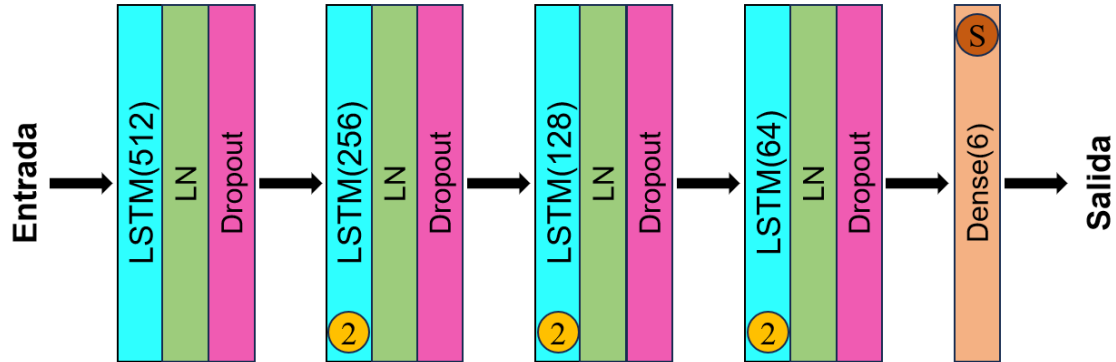


Figura 4.16. Arquitectura con capas LSTM para Spanish MEACorpus 2023.

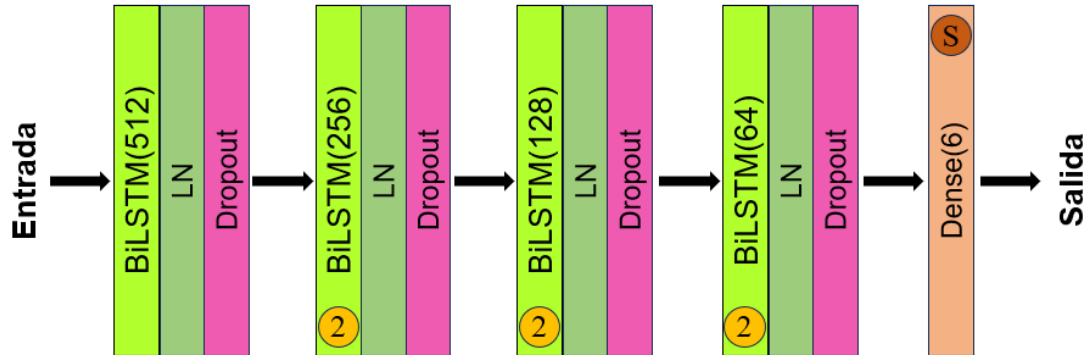


Figura 4.17. Arquitectura con capas BiLSTM para Spanish MEACorpus 2023.

La principal diferencia en esta etapa de experimentos radica en el tratamiento del conjunto de datos MEACorpus, cuyas características particulares fueron detalladas en la sección de preprocesamiento (sección 4.1). En este caso, los audios fueron procesados para generar secuencias de 1000 cuadros MFCC (aproximadamente 10 segundos de duración) o imágenes de espectrogramas log-Mel, dependiendo del modelo.

MEACorpus es un conjunto el cual ya tiene dividido el conjunto de entrenamiento y el conjunto de validación, por lo que no fue necesario aplicar un proceso de partición adicional como en el caso de la base de datos EmoMatch.

En las siguientes subsecciones se presentan los resultados obtenidos por cada tipo de arquitectura al ser entrenadas sobre este conjunto de datos.

4.7 Resultados con EmoMatch y MEACorpus

En la Tabla 4.1 se muestran los resultados obtenidos por medio de los experimentos realizados con los modelos CNN en EmoMatch.

Tabla 4.1. Resultados de los experimentos realizados con CNN en EmoMatchSpanishDB.

Modelos	ResNet50	MobileNetV2	EfficientNetB0	VGG16	InceptionV3
Parámetros	23,851,399	2,426,367	4,214,954	14,781,767	22,066,471
Memoria necesaria	90.99MB	9.24MB	16.08MB	56.39MB	84.18MB
Exactitud	64.84%	51.12%	69.83%	65.34%	56.61%
Perdida	1.2818	2.6444	1.4381	1.5593	1.9829
Precisión	0.6484	0.5137	0.6967	0.7264	0.6124
Sensibilidad	0.6324	0.5112	0.6983	0.6534	0.5661
Medida-F1	0.6325	0.4880	0.6942	0.6397	0.5697

Por otro lado, en la Tabla 4.2 se encuentran los resultados obtenidos en cuanto a las arquitecturas RNN y la arquitectura ViT con la base de datos EmoMatchSpanishDB.

Tabla 4.2. Resultados de los experimentos realizados con RNN y ViT en EmoMatchSpanishDB.

Modelos	LSTM	BiLSTM	ViT
Parámetros	550,983	1,429,639	7,814,087
Memoria necesaria	2.1MB	5.45MB	29.81MB
Exactitud	46.13%	57.36%	52.12%
Perdida	2.5884	3.8018	1.3164
Precisión	0.4446	0.5788	0.5053
Sensibilidad	0.4613	0.5736	0.5112
Medida-F1	0.4426	0.5730	0.5046

En cuanto a la base de datos MEACorpus la distribución de las tablas es la misma, por los que en la Tabla 4.3 se encuentran los resultados obtenidos con las arquitecturas de CNN y en la Tabla 4.4 se encuentran los resultados obtenidos con las arquitecturas RNN y la ViT.

Tabla 4.3. Resultados de los experimentos realizados con CNN en Spanish MEACorpus 2023.

Modelos	ResNet50	MobileNetV2	EfficientNetB0	VGG16	InceptionV3
Parámetros	23,851,270	2,423,238	4,214,825	14,781,638	22,066,342
Memoria necesaria	90.99MB	9.24MB	16.08MB	56.39MB	84.18MB
Exactitud	83.75%	73.40%	78.67%	63.98%	78.47%
Perdida	0.7148	1.4399	1.2134	1.3036	0.9622
Precisión	0.8376	0.7359	0.7919	0.7633	0.7818
Sensibilidad	0.8375	0.7340	0.7867	0.6398	0.7847
Medida-F1	0.8372	0.7306	0.7864	0.6086	0.7790

Tabla 4.4. Resultados de los experimentos realizados con RNN y ViT en Spanish MEACorpus 2023.

Modelos	LSTM	BiLSTM	ViT
Parámetros	2,168,278	5,708,806	7,813,830
Memoria necesaria	8.26MB	21.78MB	29.81MB
Exactitud	68.49%	76.44%	74.22%
Perdida	1.9828	1.7355	0.7977
Precisión	0.6843	0.7618	0.7547
Sensibilidad	0.6894	0.7644	0.7422
Medida-F1	0.6854	0.7625	0.7364

Los resultados experimentales muestran que los modelos evaluados alcanzan un mejor desempeño en la base MEACorpus en comparación con EmoMatch, lo cual puede atribuirse principalmente a la mayor cantidad de muestras y a la mayor duración promedio de los audios en MEACorpus. Otro aspecto a considerar es el contexto de grabación, ya que, aunque MEACorpus representa emociones en contextos naturales (lo que podría introducir ruido o ambigüedad), parece ser más efectiva en transmitir rasgos emocionales distintivos que facilita a los modelos a identificar las emociones. Por otra parte, EmoMatch, aunque fue grabada en

ambientes controlados, involucra sujetos no-actores, lo cual podría haber reducido la expresividad emocional y, por tanto, la separabilidad de las clases.

En cuestiones más específicas de las arquitecturas, los resultados obtenidos permiten comparar el desempeño de distintas redes profundas utilizando dos representaciones distintas de señal de voz: imágenes de espectrogramas log-Mel y secuencias de MFCC.

En EmoMatch, los modelos basados en log-Mel obtuvieron los mejores resultados. EfficientNetB0 alcanzó el mayor rendimiento con una exactitud del 69.83%, seguido por ResNet50 (64.84%) y VGG16 (65.34%) como se ve en la Tabla 4.1. Así mismo, el modelo BiLSTM que utilizó secuencias MFCC como entrada obtuvo una precisión del 57.36%, superando claramente a la arquitectura que utiliza LSTM simples de acuerdo a la Tabla 4.2. Estos resultados indican que los espectrogramas log-Mel, al conservar una representación visual rica en información espectral y temporal, proporcionan ventajas significativas cuando se combinan con arquitecturas de redes convolucionales modernas.

En MEACorpus, todos los modelos mejoraron su desempeño. Siendo ahora ResNet50 quien lideró los resultados con exactitud del 83.75%, superando a todas las demás arquitecturas evaluadas, seguida de EfficientNetB0 (78.67%) e InceptionV3 (78.47%) como se observa en la Tabla 4.3. El modelo BiLSTM con MFCC también mostró un aumento significativo respecto a EmoMatch, alcanzando un 76.44% de exactitud como se ve en la Tabla 4.4. Esta mejora sugiere que los factores discutidos anteriormente (mayor cantidad de datos, mayor duración de los audios y mayor expresividad) impactan positivamente en la capacidad de generalización de los modelos.

Si bien ViT no alcanzó los mejores resultados, es importante destacarlo por su potencial. En EmoMatch obtuvo un modesto 52.12% (Tabla 4.02), pero en MEACorpus escaló hasta 74.22% (Tabla 4.4), lo que indica que este tipo de arquitectura se beneficia fuertemente de conjuntos de datos más grandes. Dado que los modelos con auto-atención representan una tendencia actual

en el estado del arte, su integración con arquitecturas convolucionales constituye una línea de investigación prometedora.

En conjunto, los resultados reafirman el desempeño sobresaliente de las arquitecturas convolucionales modernas, destacando particularmente a ResNet50 y EfficientNetB0. No obstante, ResNet50 presenta el mejor desempeño promedio entre las bases de datos evaluadas, evidenciando una mayor capacidad de generalización y estabilidad frente a distintos escenarios experimentales. Por esta razón, en la siguiente etapa de experimentación se profundizará en el análisis y optimización de la arquitectura ResNet50, explorando variantes y la incorporación de mecanismos de atención.

4.8 Experimentos con variantes de ResNet

Con base en los resultados obtenidos, se llevó a cabo un análisis más detallado de la arquitectura ResNet50 mediante la evaluación de distintas variantes estructurales. En particular, se exploró el impacto de la profundidad de la red modificando el número de bloques residuales, tomando como punto de corte la capa final de cada uno de los cuatro bloques principales. El objetivo de estos experimentos fue analizar cómo la reducción progresiva de la capacidad jerárquica de aprendizaje afecta el desempeño del modelo en la tarea de reconocimiento de emociones.

Las variantes fueron denominadas de la siguiente forma:

- ResNet50-B1: Incluye únicamente el primer bloque residual (Figura 4.18).

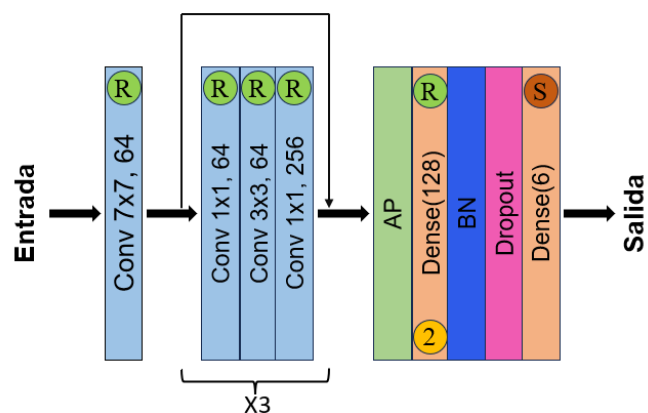


Figura 4.18. Arquitectura ResNet50-B1 con un bloque residual.

- ResNet50-B2: Incluye los dos primeros bloques residuales como en la Figura 4.19.

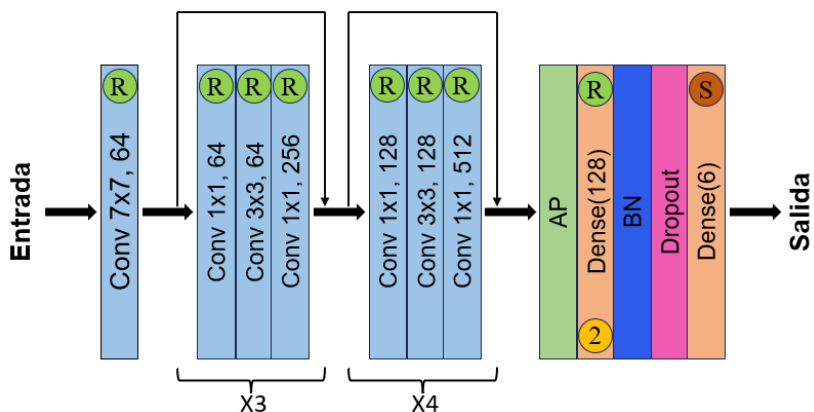


Figura 4.19. Arquitectura ResNet50-B2 con dos bloques residuales.

- ResNet50-B3: Incluye los tres primeros bloques residuales como se muestra en la Figura 4.20.

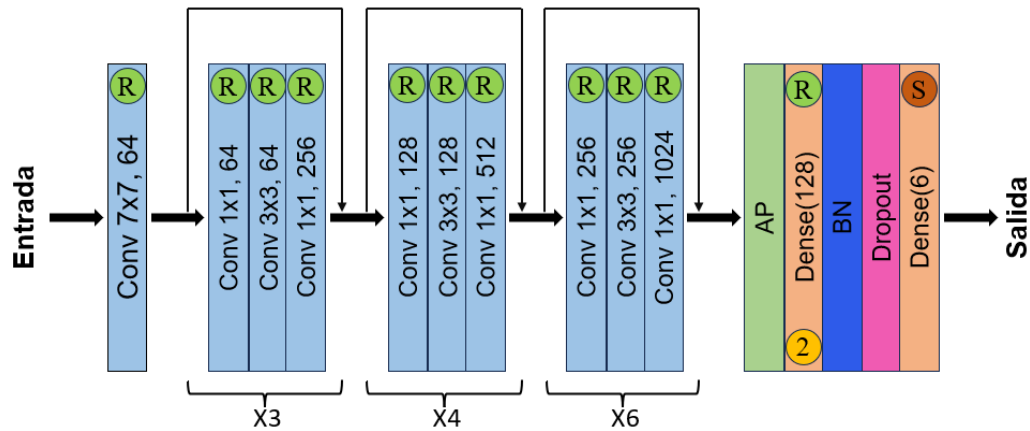


Figura 4.20. Arquitectura ResNet50-B3 con tres bloques residuales.

- ResNet50-B4: Arquitectura completa sin la capa de clasificación del modelo base como se observa en la Figura 4.21.

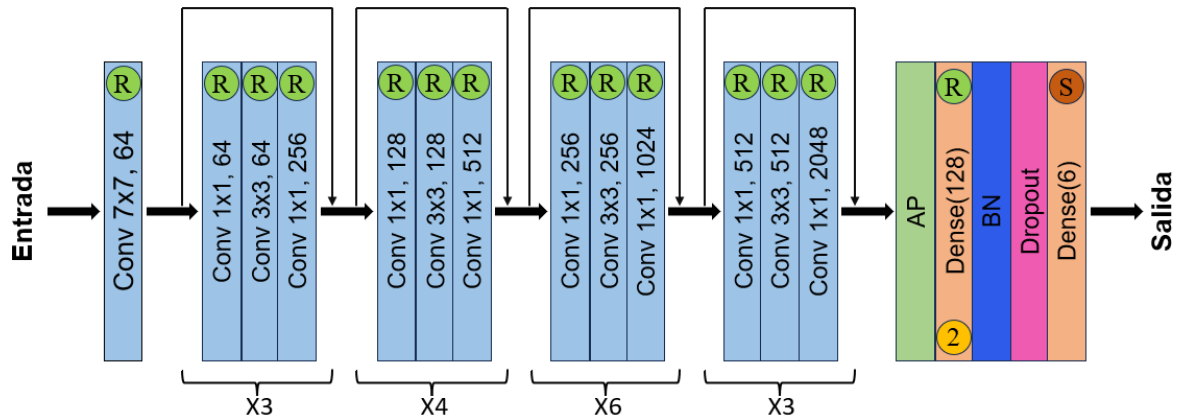


Figura 4.21. Arquitectura ResNet50-B4 con todos los bloques residuales.

En todas las variantes, se mantuvo el mismo bloque de clasificación personalizado utilizado en los modelos CNN previos como el presentado en la Figura 4.13.

Adicional a la experimentación con esas variantes de ResNet50 se llevó a cabo un experimento con el modelo ResNet101 para comprobar si una variante más compleja de la arquitectura puede obtener mejores resultados. Este caso mantuvo una ligera diferencia en cuanto al bloque de clasificación personalizado siendo de la siguiente manera: una capa

GlobalAveragePooling2D, seguida por una capa densa de 256 neuronas con activación ReLU, normalización por lotes, con Dropout y una capa densa de salida con activación Softmax.

4.8.1 Resultados en variantes de ResNet

Los resultados obtenidos a partir de las variantes de ResNet se muestran en la Tabla 4.5.

Tabla 4.5. Resultados con las variantes de ResNet50.

Modelos	ResNet50-B4	ResNet50-B3	ResNet50-B2	ResNet50-B1	ResNet101
Parámetros	23,851,270	8,721,670	1,527,046	263,942	43,185,286
Memoria necesaria	90.99MB	33.27MB	5.83MB	1.01MB	164.74MB
Exactitud	83.23%	80.95%	81.68%	75.16%	81.78%
Perdida	0.7908	0.6327	0.7213	0.9704	0.7602
Precisión	0.8360	0.8075	0.8182	0.7657	0.8189
Sensibilidad	0.8323	0.8095	0.8168	0.7516	0.8178
Medida-F1	0.8300	0.8070	0.8160	0.7444	0.8154

A partir de los resultados obtenidos, se observa que la eliminación progresiva de bloques residuales en la arquitectura ResNet50 conlleva una disminución considerable en el desempeño del modelo. En particular, las variantes con menor número de bloques (ResNet50-B3, ResNet50-B2 y ResNet50-B1) presentan una reducción progresiva en la exactitud, precisión y medida-F1, lo que confirma la importancia de mantener la profundidad y las conexiones residuales completas para una extracción efectiva de características emocionales.

La variante ResNet50-B4, correspondiente a la arquitectura completa, obtuvo el mejor desempeño con una exactitud del 83.23%, por lo que es la configuración más robusta entre las evaluadas. Por otro lado, ResNet101, al ser una versión con mayor profundidad, presenta un número significativamente superior de parámetros y un mayor consumo de memoria. Aún así, no superó el desempeño de ResNet50, alcanzando una exactitud del 81.78%. Este resultado indica que un incremento en la profundidad del modelo no garantiza necesariamente una mejora

en el rendimiento, y confirma que ResNet50 ofrece un mejor equilibrio entre complejidad y desempeño para la tarea.

Estos resultados sugieren que existe un equilibrio óptimo entre complejidad del modelo y capacidad de generalización, el cual se alcanza con la arquitectura ResNet50. Si bien esta red demuestra una alta capacidad para capturar patrones locales en el dominio tiempo-frecuencia mediante convoluciones profundas, su diseño se basa principalmente en operaciones locales.

En consecuencia, y con el objetivo de potenciar la modelación de dependencias globales en la señal de voz, se propone en la siguiente etapa de experimentación el desarrollo de un modelo híbrido que combine la capacidad de extracción local de características de ResNet50 con arquitecturas tipo Transformer. Esta integración busca aprovechar las fortalezas de ambos enfoques para mejorar la discriminación emocional en el habla en español.

4.9 Experimentación de modelos híbridos ResNet50-Transformer en serie y paralelo

Con base en los resultados obtenidos con ResNet50 en ambas bases de datos, así como en las variaciones evaluadas de esta misma arquitectura, se concluyó que dicho modelo constituye un componente clave para el desarrollo de la propuesta final. A partir de esta conclusión, se diseñó un nuevo esquema de experimentación cuyo objetivo fue explorar distintas estrategias para mejorar el desempeño de ResNet50.

En este nuevo esquema se buscan incorporar bloques Transformer básicos, donde ResNet50 se encarga de capturar información espacial y dependencias locales, mientras que la sección correspondiente al bloque Transformer extrae información temporal para obtener características globales. Bajo esta hipótesis, se espera que el modelo híbrido logre extraer características más relevantes y realice una mejor clasificación de las emociones.

Se plantearon dos arquitecturas híbridas: serie (Figura 4.22) y paralelo (Figura 4.23).

- **Modelo híbrido ResNet50-Transformer en serie:** los espectrogramas log-Mel se utilizan como entrada del modelo ResNet50. El último bloque residual de esta red genera un mapa de características que alimenta una serie de bloques Transformer básicos. La salida resultante se procesa mediante una capa de agrupación por promedio bidimensional, seguida de un Dropout, y finalmente pasa a una capa totalmente conectada con activación Softmax para la clasificación (Figura 4.22).

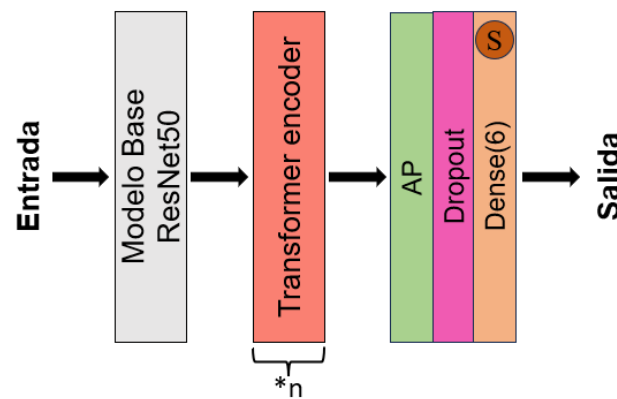


Figura 4.22. Modelo híbrido en serie con ResNet50 y bloques Transformer.

- **Modelo híbrido ResNet50-Transformer en paralelo:** se utilizan dos tipos de entradas. Por un lado, los espectrogramas log-Mel alimentan a ResNet50, mientras que, por otro lado, los MFCC se emplean como entrada a los bloques Transformer, aprovechando su capacidad para modelar información secuencial. Las salidas del bloque convolucional y del bloque Transformer se someten a una capa de agrupación por promedio bidimensional y posteriormente se concatenan. Finalmente, este vector combinado se procesa mediante una FFNN de baja complejidad para obtener la clasificación final (Figura 4.23).

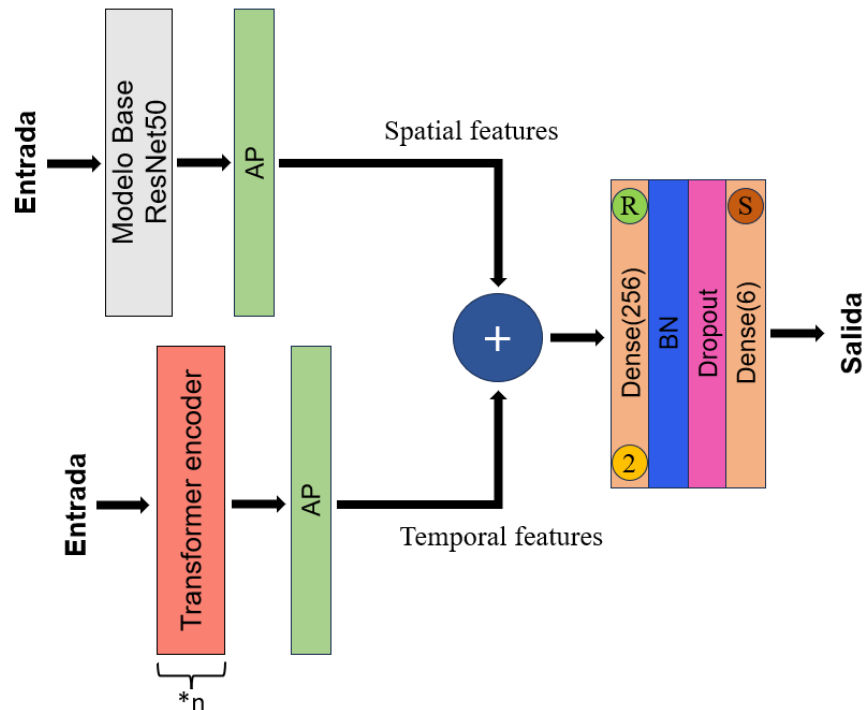


Figura 4.23. Modelo híbrido en paralelo con ResNet50 y bloques Transformer.

La base de datos empleada en ambos esquemas fue MEACorpus, utilizando los siguientes hiperparámetros de entrenamiento:

- Tamaño de lote: 32
- Tasa de aprendizaje: 0.0001
- Optimizador: Adam
- Épocas máximas: 300
- Suavizado de etiquetas: 0.1

4.9.1 Resultados de la arquitectura ResNet50-Transformer en serie

Siguiendo el esquema de la arquitectura híbrida en serie, se realizaron cuatro experimentos con diferentes configuraciones de parámetros. Los resultados obtenidos se resumen en la Tabla 4.6.

Tabla 4.6. Resultados de los experimentos realizados con las variaciones de ResNet50-Transformer en serie.

Modelos	Versión 1	Versión 2	Versión 3	Versión 4
Parámetros	36,223,622	42,517,382	42,491,526	59,272,838
Flops	8.346Gflops	8.346Gflops	9.371Gflops	11.015Gflops
Memoria necesaria	138.18MB	162.19MB	162.09MB	226.11MB
FPS	239.91	224.49	278.72	258.37
Test Accuracy	76.53%	77.12%	80.92%	79.65%
Test Loss	1.0750	1.0560	0.9284	1.0013
Precision	0.7672	0.7819	0.8176	0.7993
Recall	0.7653	0.7712	0.8092	0.7965
F1 Score	0.7634	0.7708	0.8107	0.7957
BAM	0.7558	0.7025	0.7493	0.7109

4.9.2 Resultados de la arquitectura ResNet50-Transformer en paralelo

Para el esquema híbrido en paralelo se llevaron a cabo tres experimentos, también con variaciones en los parámetros de la arquitectura. Los resultados se muestran en la Tabla 4.7.

Tabla 4.7. Resultados de los experimentos realizados con las variaciones de ResNet50-Transformer en paralelo.

Modelos	Versión 1	Versión 2	Versión 3
Parámetros	24,368,964	24,450,350	24,369,454
Flops	8.029Gflops	8.131Gflops	7.97Gflops
Memoria necesaria	92.96MB	93.27MB	92.96MB
FPS	0.30	0.27	0.27
Test Accuracy	73.81%	73.03%	74.29%
Test Loss	1.1239	1.1671	1.1973
Precision	0.7457	0.7386	0.7763
Recall	0.7381	0.7303	0.7429
F1 Score	0.7381	0.7215	0.7500
BAM	0.6134	0.6320	0.6618

4.9.3 Análisis de resultados

Como se observa en la Tabla 4.6 y la Tabla 4.7 la arquitectura híbrida en serie presenta un desempeño superior en comparación con la versión en paralelo. Esta diferencia puede atribuirse

a diversos factores tanto estructurales como relacionados con las características del conjunto de datos.

En primer lugar, la cantidad limitada de muestras en la base de datos dificulta el entrenamiento de modelos Transformer de manera directa, ya que estas arquitecturas suelen requerir grandes volúmenes de datos para generalizar adecuadamente. Este problema se atenúa en la arquitectura en serie al utilizar ResNet50 como extractor inicial de características, lo que permite obtener representaciones más compactas y robustas antes de que los bloques *Transformer* procesen la información.

En segundo lugar, la arquitectura en serie resulta más adecuada para la naturaleza de la tarea. Al incorporar primero la etapa convolucional de ResNet50, el modelo captura dependencias locales presentes en los espectrogramas log-Mel y genera un mapa de características que describe patrones espaciales relevantes. Posteriormente, el bloque Transformer modela dependencias globales y relaciones temporales a partir de estas características previamente extraídas. Esta combinación jerárquica favorece una representación más rica de la señal, mejorando el proceso de inferencia y, por tanto, el desempeño global del modelo.

4.10 Análisis de distintas arquitecturas Transformer

A partir de los experimentos realizados con las arquitecturas híbridas ResNet50-Transformer en serie y en paralelo, se confirmó que la versión de la arquitectura ResNet50-Transformer en serie es superior para cumplir con la tarea. Sin embargo, se consideró que el desempeño del modelo híbrido podría beneficiarse al emplear una arquitectura Transformer distinta a los bloques básicos utilizados en las pruebas anteriores.

Por esta razón, se llevó a cabo una exploración del estado del arte con el propósito de identificar arquitecturas Transformer que pudieran resultar más adecuadas para el procesamiento de señales de audio y para el reconocimiento de emociones a partir del habla. En esta búsqueda se tomó en cuenta el Vision Transformer (previamente descrito y utilizado en los experimentos), así como otras arquitecturas nuevas que se describirán continuación.

4.10.1 Transformer de espectrogramas de audio

El Transformer de espectrogramas de audio (o AST por sus siglas en inglés) es una arquitectura derivada del Vision Transformer, adaptada para trabajar con espectrogramas de audio. Su principio fundamental consiste en tratar los espectrogramas como si fueran imágenes, dividiéndolos en pequeños parches de tamaño fijo (comúnmente de 16×16). La arquitectura se encuentra en la Figura 4.24.

Cada parche se proyecta inicialmente a un vector de características, al cual se le agrega información posicional mediante un vector de posición entrenable y un token especial de clasificación ([CLS]), también aprendido durante el entrenamiento. La secuencia resultante se introduce en un bloque codificador Transformer, que modela las relaciones entre los parches y extrae representaciones de alto nivel. Finalmente, la representación asociada al token [CLS] se utiliza para determinar la clase de salida en la tarea de clasificación [33].

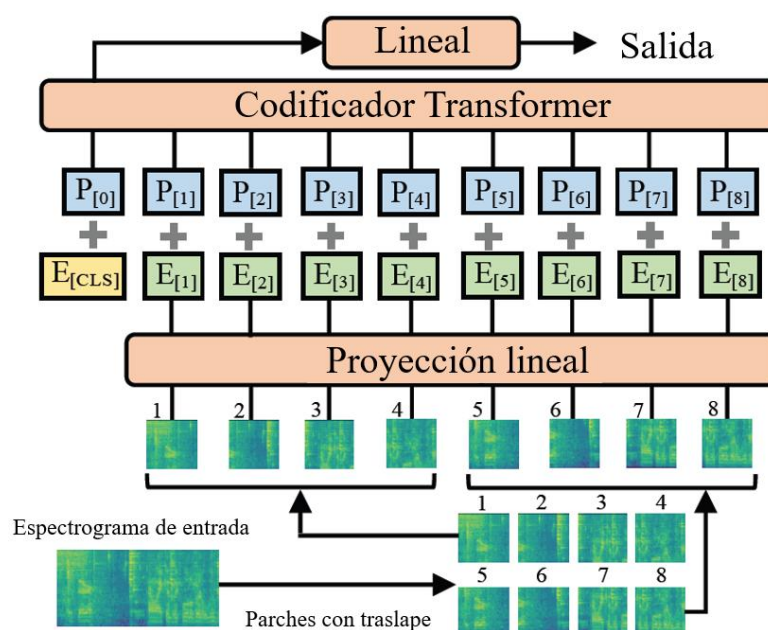


Figura 4.24. Arquitectura del Audio Spectrogram Transformer.

4.10.2 Performer

El Performer es una variante del modelo Transformer que busca resolver el alto costo computacional del mecanismo de atención tradicional. Propone una versión eficiente conocida como atención lineal llamada FAVOR+, la cual reemplaza el cálculo cuadrático de la atención estándar por una aproximación basada en proyecciones aleatorias (*Random Features*), lo que permite reducir la complejidad del mecanismo de atención de $O(n^2)$ a $O(n)$, pero manteniendo un desempeño competitivo. En la Figura 4.25 se puede ver una representación gráfica de la equivalencia de ambos tipos de atención. En esencia, el Performer conserva la capacidad del Transformer para capturar dependencias globales, pero con un consumo de memoria y tiempo mucho menor, lo que lo hace adecuado para secuencias largas y tareas de alto costo computacional [34].

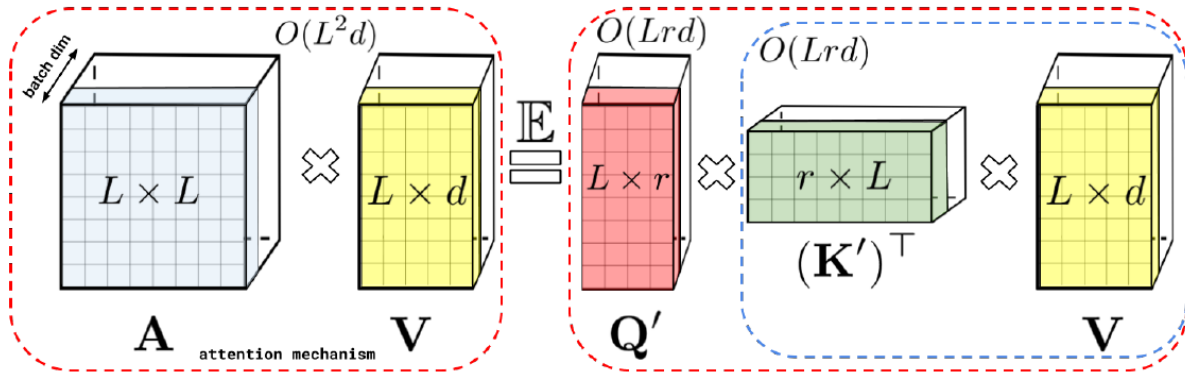


Figura 4.25. Análisis de las matrices en los mecanismos de atención convencionales y en el mecanismo de atención FAVOR+.

4.10.3 Conformer

El Conformer (*Convolution-augmented Transformer*) es una arquitectura que combina las ventajas de las CNN y los Transformers aplicados al reconocimiento de voz. Su diseño busca capturar tanto las propiedades locales de las señales de audio que se modelan con convoluciones, como las características globales que se aprenden mediante la auto-atención. Cada bloque Conformer integra módulos de FFNN, auto-atención multi-cabeza con codificación posicional relativa, y convoluciones unidimensionales. Estos módulos están

organizados en una estructura simétrica de doble red totalmente conectada hacia adelante con conexiones residuales parciales tal como se puede ver en la Figura 4.26. Esta combinación permite representar de forma eficiente las características temporales y espaciales del audio, logrando un equilibrio entre precisión y costo computacional. Mediante esta integración, el Conformer ha alcanzado resultados de vanguardia en tareas de reconocimiento automático del habla, superando a arquitecturas basadas únicamente en Transformers o CNN [35].

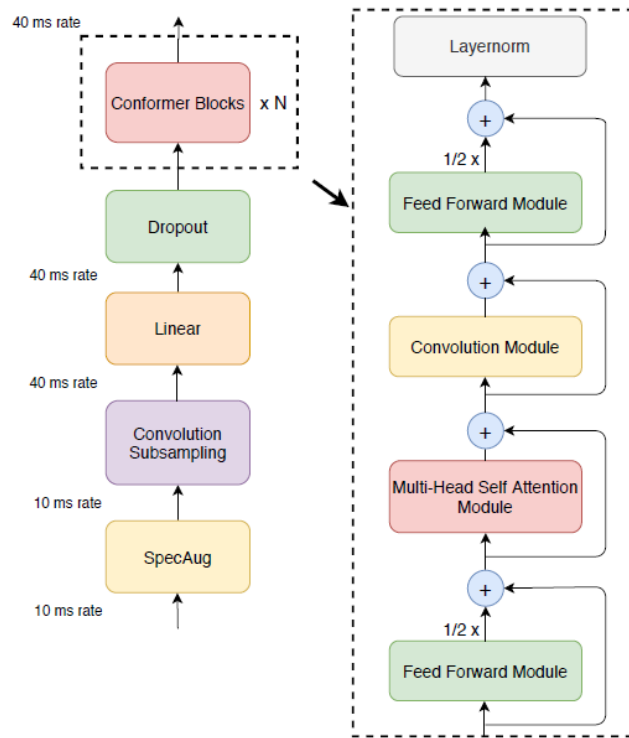


Figura 4.26. Arquitectura de un bloque Conformer.

A partir de los resultados obtenidos y del análisis de las distintas arquitecturas evaluadas, se confirma que el modelo híbrido ResNet50–Transformer en serie constituye una solución sólida y competitiva para la tarea de reconocimiento de emociones en el habla, alcanzando una exactitud máxima del 80.92%. No obstante, el análisis realizado sugiere que el desempeño de la arquitectura podría beneficiarse al incorporar arquitecturas tipo Transformer diseñadas específicamente para el procesamiento de señales de audio. En este contexto, arquitecturas

como el Conformer, que integran de manera explícita convoluciones y auto-atención, representan una alternativa prometedora para mejorar la captura conjunta de dependencias locales y globales. Estas observaciones motivan el desarrollo de la arquitectura propuesta, la cual se describe en el siguiente capítulo, donde se explora una arquitectura híbrida basada en ResNet50 como extractor de características y un bloque Transformer/Conformer.

V. ARQUITECTURA PROPUESTA

A partir de los resultados obtenidos en el capítulo anterior, se determinó que la estructura más adecuada corresponde a una arquitectura híbrida basada en ResNet50 combinada con bloques Transformer o bloques Conformer. Basándose en estos resultados, en este capítulo se describe en detalle la arquitectura del modelo propuesto. La organización del capítulo es la siguiente: la sección 5.1 define la entrada del modelo, la sección 5.2 define las técnicas de regularización utilizadas, la sección 5.3 describe detalladamente la red neuronal convolucional, la sección 5.4 aborda el bloque codificador Transformer/Conformer y la Sección 5.5 describe la etapa de clasificación.

La Figura 5.1 presenta una vista general de la arquitectura propuesta, donde la entrada es un espectrograma log-Mel. La siguiente etapa es una serie de capas convolucionales y conexiones residuales características de la arquitectura ResNet50. Posteriormente, se incorpora un bloque codificador Transformer (o Conformer) y, finalmente, una sección totalmente conectada encargada de realizar la clasificación de las emociones.

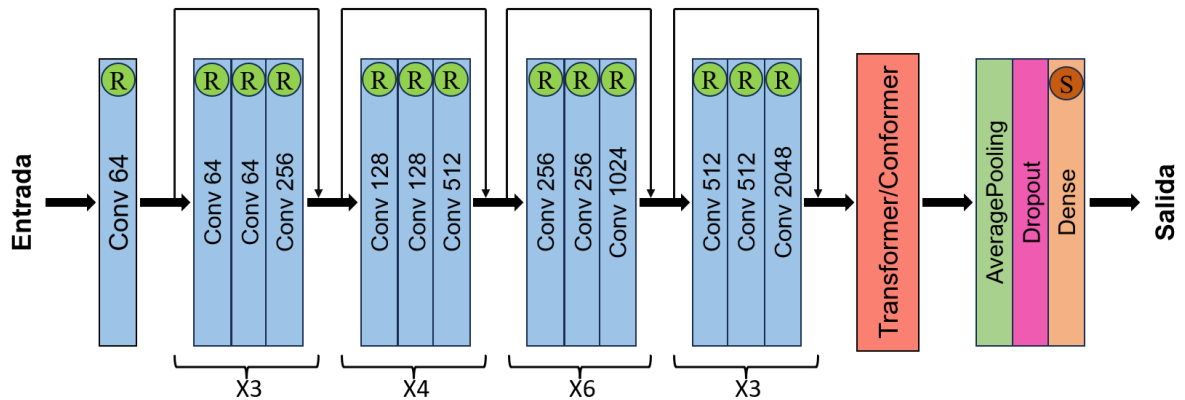


Figura 5.1. Arquitectura propuesta ResNet50-Transformer/Conformer.

5.1 Entrada del modelo

La entrada del modelo estará constituida por un espectrograma logarítmico de Mel con 128 coeficientes, calculados sobre ventanas temporales de 85.33ms y un solapamiento del 75% entre cada ventana. Este proceso produce espectrogramas con una duración aproximada de 13

segundos por muestra en MEACorpus o 4.5 segundos por muestra en el caso de EmoMatch. La obtención de dichos espectrogramas se realiza a partir de la señal de audio original en el dominio temporal mediante la Transformada Rápida de Fourier (FFT, por sus siglas en inglés), definida como:

$$X(k, m) = FFT\{x[n]\} = \sum_{n=0}^{N-1} x[n + mH] * w[n] * e^{-j2\pi kn/N} \quad (5.1)$$

donde $x[n]$ es la señal en el tiempo, $w[n]$ representa una ventana Hamming, N es el tamaño de la Transformada Rápida de Fourier, H corresponde al salto o desplazamiento entre ventanas, m es el índice temporal de la ventana que se está obteniendo y k el índice de frecuencia. A partir del espectrograma obtenido, se realiza un mapeo a la escala Mel mediante la aplicación de un banco de filtros Mel, según la expresión:

$$S_{mel}(m, i) = \sum_k M(p, k) * |X(m, k)|^2 \quad (5.2)$$

donde i indica el índice de coeficientes Mel, M es la matriz del banco de filtros Mel y S_{mel} representa el espectrograma en la escala Mel. Posterior al mapeo se aplica una transformación logarítmica para obtener el espectrograma logarítmico de Mel, definido como:

$$S_{logmel}(m, i) = \log(S_{mel}(m, p) + \epsilon) \quad (5.3)$$

a través de este proceso se obtiene una representación bidimensional que combina información espectral y prosódica de la señal de voz, la cual se empleará como entrada a la arquitectura híbrida en serie propuesta. La representación formal de los espectrogramas log-Mel se define como:

$$S_{logMel} \in \mathbb{R}^{n_{mels} \times T} \quad (5.4)$$

donde n_{mels} es la cantidad de coeficientes Mel y T es la cantidad de ventanas de tiempo generadas a partir de la FFT. Esta matriz constituye una representación tiempo-frecuencia comprimida en escala Mel, utilizada como entrada para el modelo propuesto.

En las siguientes secciones se describirán de manera detallada los componentes principales del modelo, incluyendo la sección convolucional basada en ResNet50, el módulo de atención (Transformer o Conformer) y el bloque de clasificación final.

5.2 Regularizaciones

Antes de describir en detalle los componentes que conforman la arquitectura propuesta, resulta necesario definir las técnicas de regularización empleadas en el modelo. Estas técnicas tienen como objetivo mejorar la capacidad de generalización y mitigar el problema de sobreajuste durante el entrenamiento. En este trabajo se utilizan dos mecanismos de regularización complementarios: la penalización L2 (*weight decay*) y el Dropout, aplicado sobre las activaciones de las capas convolucionales.

5.2.1 Penalización L2

La regularización con penalización L2 se aplica de forma homogénea a lo largo de todo el bloque de red neuronal convolucional del modelo. Matemáticamente, la función de pérdida total utilizada durante el entrenamiento queda definida como la suma de la pérdida de datos y la penalización L2 sobre los parámetros entrenables de la siguiente forma:

$$L_{total} = L_{data}(Y_{true}, Y_{pred}) + \lambda_2 \sum_{l \in C_{conv}} \|W_l\|_2^2 \quad (5.5)$$

donde L_{data} corresponde a la entropía cruzada categórica multiclase, C_{conv} es el conjunto de capas convolucionales en el modelo, W_l son los tensores de pesos de la capa l y λ_2 es el hiperparámetro de regularización L2, cuyo valor se estableció en $\lambda_2 = 1^{-6}$.

Al aplicar esta penalización, cada término $\|W_l\|_2^2$ se expresa explícitamente como la suma de los cuadrados de los elementos del tensor de pesos:

$$\|W_l\|_2^2 = \sum_{i,j,ch_{in},ch_{out}} (W_l(i,j,ch_{in},ch_{out}))^2 \quad (5.6)$$

donde los índices i y j recorren las dimensiones espaciales del kernel, ch_{in} corresponde a los canales de entrada y ch_{out} a los canales de salida. En la práctica, este tipo de regularización no

afecta en el flujo hacia delante de la red, sino que actúa en la optimización al añadirse a la función de costo.

5.2.2 Dropout

A diferencia de la regularización L2, el Dropout sí modifica el flujo hacia adelante ya que consiste en desactivar aleatoriamente una fracción p de las neuronas durante el entrenamiento. Esto impide que la red dependa excesivamente de combinaciones específicas de neuronas y, por tanto, mejora su capacidad de generalización. Lo que hace el *dropout* es que introduce una máscara aleatoria r del mismo tamaño del vector de activaciones, cuyos elementos siguen una distribución Bernoulli:

$$r_i \sim \text{Bernoulli}(1 - p) \quad (5.7)$$

es decir,

$$r_i = \begin{cases} 1, & \text{con probabilidad } 1 - p, \\ 0, & \text{con probabilidad } p. \end{cases} \quad (5.8)$$

las activaciones regularizadas se obtienen mediante el producto elemento a elemento:

$$h'_i = r_i h_i \quad (5.9)$$

donde h_i corresponde a la activación original y h'_i a la activación después de aplicar el Dropout. Durante la fase de inferencia, el Dropout no se aplica, utilizando todas las activaciones completas. En este modelo, se empleó una probabilidad $p = 0.1$ en las capas convolucionales y residuales, lo que permite mantener un equilibrio adecuado entre regularización y retención de información.

En resumen, el bloque convolucional basado en ResNet50 constituye la base de la arquitectura híbrida propuesta, integrando regularización L2 sobre los parámetros y Dropout sobre las activaciones con el fin de mejorar la capacidad de generalización del modelo. A continuación, se describen los bloques que conforman esta sección convolucional, detallando sus operaciones y estructura matemática.

5.3 Módulo ResNet50

El módulo ResNet50 desarrollado para este trabajo consiste en un conjunto de capas convolucionales agrupados en bloques los cuales cuentan con regularizaciones L2 y Dropout. Este módulo se basa de manera casi íntegra a la arquitectura ResNet50 original. Por lo tanto, es adecuado explicar matemáticamente en qué consisten los bloques residuales que utilizan este tipo de arquitecturas.

Un bloque residual consiste en un bloque con una o más capas con alimentación hacia adelante, donde se suma la entrada y la salida del bloque de la siguiente manera:

$$X = x + f(x) \quad (5.10)$$

donde x corresponde a la entrada del bloque, $f(x)$ es la salida del bloque residual. En el caso de ResNet, el bloque consta de capas convolucionales, formándose una estructura como la mostrada en la Figura 5.2.

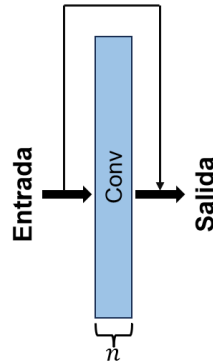


Figura 5.2. Arquitectura básica de un bloque residual.

Partiendo del concepto de bloque residual, ResNet se divide de la siguiente manera: el bloque inicial y el conjunto de bloques residuales 1, 2, 3 y 4. Antes de explicar a detalle las subsecciones de ResNet, se definirán las regularizaciones utilizadas.

5.3.1 Bloque inicial (Conv1_x)

El bloque inicial recibe el S_{logmel} y genera un mapa de características de bajo nivel. Para ello, se aplica un relleno de ceros (*padding*) en S_{logmel} con el propósito de conservar sus

dimensiones tras la operación convolucional de la ecuación 5.11. La capa convolucional posterior utiliza un kernel de tamaño 7×7 con un *stride* de 2 y 64 filtros orientados a capturar patrones espaciales amplios presentes en el espectrograma.

$$Z^{(64)} = S_{logmel} * W^{(64)} + b^{(64)} \quad (5.11)$$

Posteriormente, se realiza una normalización por lotes (BN) para estabilizar la activación de las neuronas y acelerar el proceso de entrenamiento. El BN se obtiene de la siguiente manera:

$$BN(Z^{(64)}) = \frac{Z^{(64)} - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \cdot \gamma + \beta \quad (5.12)$$

donde γ es un parámetro de escala y β es el parámetro de desplazamiento, ambos parámetros entrenables. Esta normalización por lotes es seguida de una activación ReLU que mantiene gradientes para activaciones positivas, mitigando el problema del desvanecimiento del gradiente y favoreciendo un entrenamiento más estable del modelo como se observa en la ecuación 5.13.

$$Y_{Conv1_x}^{(64)} = MaxPooling \left(ReLU \left(BN \left(Z^{(64)} \right) \right) \right) \quad (5.13)$$

Finalmente, se tiene una capa de *MaxPooling* para reducir la dimensionalidad espacial del espectrograma, pero reteniendo las características más relevantes. Este bloque puede ver en la Figura 5.3.

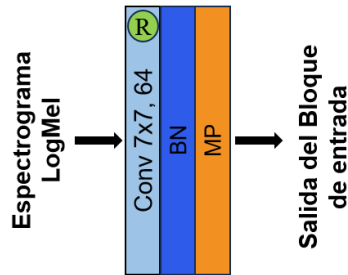


Figura 5.3. Arquitectura del Bloque de entrada.

5.3.2 Bloque residual 1 (Conv2_x)

El Bloque residual 1 recibe como entrada la salida del bloque inicial $Y_{Conv1_x}^{(64)}$. Este conjunto residual está formado por tres sub-bloques: *conv_block* con proyección (ajusta las dimensiones de canal) dados por las ecuaciones 5.14-5.18 y dos *identity_blocks* (atajo identidad) dados por las ecuaciones 5.19-5.22. Cada sub-bloque está constituido por tres capas convolucionales con kernels de tamaño 1×1 , 3×3 y 1×1 , que devuelven 3 mapas de activación de tamaño 64, 64 y 256, respectivamente.

Partiendo de la entrada del bloque que es $Y_{Conv1_x} \in \mathbb{R}^{H \times W \times 64}$ (salida del bloque inicial). Se definen como Z_1, Z_2, Z_3 las salidas intermedias de las tres convoluciones dentro de un sub-bloque y como Y_2 la salida del sub-bloque tras la suma residual y la activación. Para el primer sub-bloque *conv_block* (con proyección de la conexión residual) las ecuaciones se escriben como:

$$Z_1^{(64)} = ReLU \left(BN \left(W_1^{(64)} * Y_{Conv1_x}^{(64)} + b_1^{(64)} \right) \right) \quad (5.14)$$

$$Z_2^{(64)} = ReLU \left(BN \left(W_2^{(64)} * Z_1^{(64)} + b_2^{(64)} \right) \right) \quad (5.15)$$

$$Z_3^{(256)} = BN \left(W_3^{(256)} * Z_2^{(64)} + b_3^{(256)} \right) \quad (5.16)$$

$$R_1^{(256)} = BN \left(W_{s1}^{(256)} * Y_{Conv1_x}^{(64)} + b_{s1}^{(256)} \right) \quad (5.17)$$

$$Y_{out}^{(256)} = ReLU \left(Z_3^{(256)} + R_1^{(256)} \right) \quad (5.18)$$

Para los dos sub-bloques siguientes (*identity_blocks*), la entrada ya tiene 256 canales, por lo que el atajo es la identidad y las ecuaciones se simplifican (ya no es necesaria la proyección de la conexión residual):

$$Z_1^{(64)} = ReLU \left(BN \left(W_1^{(64)} * Y_{in}^{(256)} + b_1^{(64)} \right) \right) \quad (5.19)$$

$$Z_2^{(64)} = ReLU \left(BN \left(W_2^{(64)} * Z_1^{(64)} + b_2^{(64)} \right) \right) \quad (5.20)$$

$$Z_3^{(256)} = BN\left(W_3^{(256)} * Z_2^{(64)} + b_3^{(256)}\right) \quad (5.21)$$

$$Y_{out}^{(256)} = ReLU\left(Z_3^{(256)} + Y_{in}^{(256)}\right) \quad (5.22)$$

Por último, la salida total del Bloque residual 1 posterior a los tres sub-bloques es un tensor con 256 canales:

$$Y_{Conv2_x} \in \mathbb{R}^{H \times W \times 256} \quad (5.23)$$

Se puede apreciar de forma gráfica la arquitectura del Bloque residual 1 en la Figura 5.4.

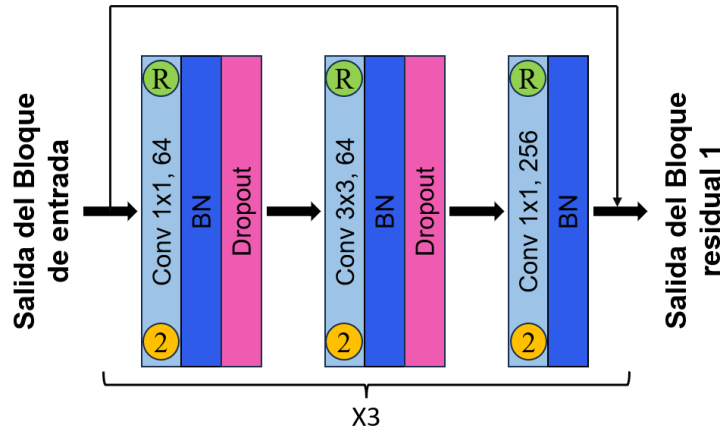


Figura 5.4. Arquitectura del Bloque residual 1.

5.3.3 Bloque residual 2 (Conv3_x)

Este grupo de capas denominado como Bloque residual 2 a diferencia del bloque anterior está conformado por cuatro sub-bloques como se observa en la Figura 5.5. El primero es un *conv_block*, dado por las ecuaciones 5.24-5.28, el cual cuenta con la proyección de la conexión residual para ajustar la dimensionalidad de la entrada con la salida y los siguientes tres serán *identity_block*, definidos por las ecuaciones 5.29-5.32, sin esta proyección. Como entrada se tendrá $Y_{Conv2_x} \in \mathbb{R}^{H \times W \times 256}$, la salida del bloque anterior.

Primero tenemos las ecuaciones del *conv_block* siendo las siguientes:

$$Z_1^{(128)} = \text{ReLU}\left(\text{BN}\left(W_1^{(128)} * Y_{\text{Conv2}_x}^{(256)} + b_1^{(128)}\right)\right) \quad (5.24)$$

$$Z_2^{(128)} = \text{ReLU}\left(\text{BN}\left(W_2^{(128)} * Z_1^{(128)} + b_2^{(128)}\right)\right) \quad (5.25)$$

$$Z_3^{(512)} = \text{BN}\left(W_3^{(512)} * Z_2^{(128)} + b_3^{(512)}\right) \quad (5.26)$$

$$R_2^{(512)} = \text{BN}\left(W_{s2}^{(512)} * Y_{\text{Conv2}_x}^{(256)} + b_{s2}^{(512)}\right) \quad (5.27)$$

$$Y_{\text{out}}^{(512)} = \text{ReLU}\left(Z_3^{(512)} + S_2^{(512)}\right) \quad (5.28)$$

Mientras que las ecuaciones de los sub-bloques *identity_block* serían las siguientes:

$$Z_1^{(128)} = \text{ReLU}\left(\text{BN}\left(W_1^{(128)} * Y_{\text{in}}^{(512)} + b_1^{(128)}\right)\right) \quad (5.29)$$

$$Z_2^{(128)} = \text{ReLU}\left(\text{BN}\left(W_2^{(128)} * Z_1^{(128)} + b_2^{(128)}\right)\right) \quad (5.30)$$

$$Z_3^{(512)} = \text{BN}\left(W_3^{(512)} * Z_2^{(128)} + b_3^{(512)}\right) \quad (5.31)$$

$$Y_{\text{out}}^{(512)} = \text{ReLU}\left(Z_3^{(512)} + Y_{\text{in}}^{(512)}\right) \quad (5.32)$$

De esta manera se obtendrá del Bloque residual 2 la siguiente salida:

$$Y_{\text{Conv3}_x} \in \mathbb{R}^{H \times W \times 512} \quad (5.33)$$

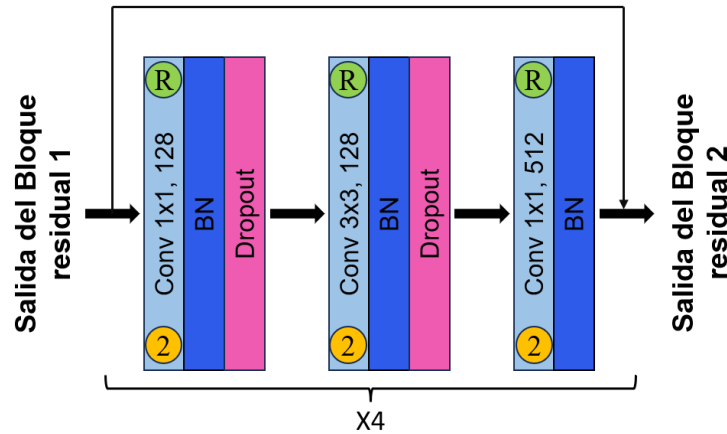


Figura 5.5. Arquitectura del Bloque residual 2.

5.3.4 Bloque residual 3 (Conv4_x)

El Bloque residual 3 (Conv4_x) es el más grande de toda la arquitectura, ya que se compone de seis sub-bloques: un *conv_block* inicial definido por las ecuaciones 5.34-5.38 y cinco *identity_blocks* dados por las ecuaciones 5.39-5.42, manteniendo una estructura general similar a la de los bloques anteriores. La principal diferencia radica en la cantidad de filtros utilizados en cada sub-bloque, que ascienden a 256, 256 y 1024 en las tres capas convolucionales respectivamente. La entrada del bloque corresponde al tensor $Y_{(\text{Conv3_x})} \in \mathbb{R}^{H \times W \times 512}$, el cual es la salida del Bloque residual 2.

Para este bloque se tendrán las siguientes ecuaciones del *conv_block*:

$$Z_1^{(256)} = \text{ReLU}\left(\text{BN}\left(W_1^{(256)} * Y_{\text{Conv3_x}}^{(512)} + b_1^{(256)}\right)\right) \quad (5.34)$$

$$Z_2^{(256)} = \text{ReLU}\left(\text{BN}\left(W_2^{(256)} * Z_1^{(256)} + b_2^{(256)}\right)\right) \quad (5.35)$$

$$Z_3^{(1024)} = \text{BN}\left(W_3^{(1024)} * Z_2^{(256)} + b_3^{(1024)}\right) \quad (5.36)$$

$$R_3^{(1024)} = \text{BN}\left(W_{s3}^{(1024)} * Y_{\text{Conv3_x}}^{(512)} + b_{s3}^{(1024)}\right) \quad (5.37)$$

$$Y_{\text{out}}^{(1024)} = \text{ReLU}\left(Z_3^{(1024)} + S_3^{(1024)}\right) \quad (5.38)$$

Por otro lado, las ecuaciones de los sub-bloques *identity_block* serían de la siguiente manera:

$$Z_1^{(256)} = \text{ReLU}\left(\text{BN}\left(W_1^{(256)} * Y_{\text{in}}^{(1024)} + b_1^{(256)}\right)\right) \quad (5.39)$$

$$Z_2^{(256)} = \text{ReLU}\left(\text{BN}\left(W_2^{(256)} * Z_1^{(256)} + b_2^{(256)}\right)\right) \quad (5.40)$$

$$Z_3^{(1024)} = \text{BN}\left(W_3^{(1024)} * Z_2^{(256)} + b_3^{(1024)}\right) \quad (5.41)$$

$$Y_{\text{out}}^{(1024)} = \text{ReLU}\left(Z_3^{(1024)} + Y_{\text{in}}^{(1024)}\right) \quad (5.42)$$

Posterior a esto se obtendrá la salida del Bloque residual 3 el cual servirá como entrada para el ultimo bloque:

$$Y_{Conv4_x} \in \mathbb{R}^{H \times W \times 1024} \quad (5.43)$$

La representación gráfica de la arquitectura del Bloque residual 3 se puede ver en la Figura 5.6.

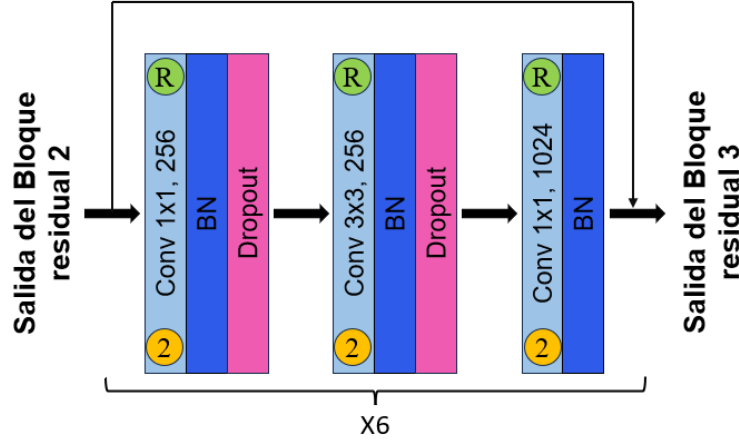


Figura 5.6. Arquitectura del Bloque residual 3.

5.3.5 Bloque residual 4 (Conv5_x)

El Bloque residual 4 constituye la última etapa convolucional de la arquitectura ResNet50 propuesta. Este bloque es responsable de extraer las características de más alto nivel dentro de las características locales. El bloque está compuesto por tres sub-bloques: el primero es un *conv_block* con proyección en la conexión residual para igualar las dimensiones entre la entrada y la salida, mientras que los dos restantes son *identity_blocks* que mantienen la misma dimensionalidad. En este bloque, el número de filtros por capa convolucional aumenta a 512, 512 y 2048, respectivamente, lo que permite ampliar la capacidad representacional del modelo.

Para el sub-bloque *conv_block*, las operaciones convolucionales se definen de la siguiente manera:

$$Z_1^{(512)} = \text{ReLU} \left(\text{BN} \left(W_1^{(512)} * Y_{Conv4_x}^{(1024)} + b_1^{(512)} \right) \right) \quad (5.44)$$

$$Z_2^{(512)} = \text{ReLU} \left(\text{BN} \left(W_2^{(512)} * Z_1^{(512)} + b_2^{(512)} \right) \right) \quad (5.45)$$

$$Z_3^{(2048)} = BN(W_3^{(2048)} * Z_2^{(512)} + b_3^{(2048)}) \quad (5.46)$$

$$R_4^{(2048)} = BN(W_{s4}^{(2048)} * Y_{Conv3_x}^{(1024)} + b_{s4}^{(2048)}) \quad (5.47)$$

$$Y_{out}^{(2048)} = ReLU(Z_3^{(2048)} + S_4^{(2048)}) \quad (5.48)$$

En los dos sub-bloques *identity_blocks* las ecuaciones que definen su operación son las siguientes:

$$Z_1^{(512)} = ReLU\left(BN\left(W_1^{(512)} * Y_{in}^{(2048)} + b_1^{(512)}\right)\right) \quad (5.49)$$

$$Z_2^{(512)} = ReLU\left(BN\left(W_2^{(512)} * Z_1^{(512)} + b_2^{(512)}\right)\right) \quad (5.50)$$

$$Z_3^{(2048)} = BN(W_3^{(2048)} * Z_2^{(512)} + b_3^{(2048)}) \quad (5.51)$$

$$Y_{out}^{(2048)} = ReLU(Z_3^{(2048)} + Y_{in}^{(2048)}) \quad (5.52)$$

El resultado final del Bloque residual 4 es un tensor tridimensional que contiene las características de mayor nivel de abstracción extraídas por la red convolucional:

$$Y_{Conv5_x} \in \mathbb{R}^{H \times W \times 2048} \quad (5.53)$$

Estas representaciones profundas son posteriormente utilizadas como entrada al módulo de atención (Transformer o Conformer), encargado de modelar las dependencias temporales y contextuales entre los marcos de tiempo del espectrograma.

La arquitectura general del Bloque residual 4 se ilustra en la Figura 5.7.

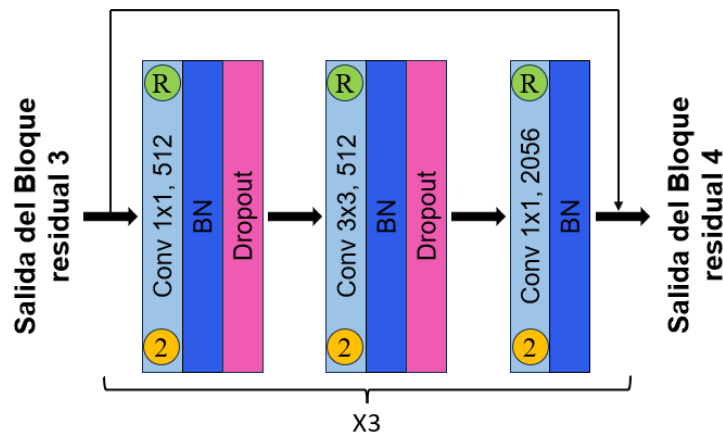


Figura 5.7. Arquitectura del Bloque residual 4.

Esta sección describe lo que conforma la parte correspondiente a la red neuronal convolucional inspirada en ResNet50 con regularizaciones L2 y dropout. Esta porción de la arquitectura actúa como extractor de características locales y espaciales a partir de los espectrogramas log-Mel, permitiendo obtener representaciones robustas y jerárquicas de la señal de voz.

Dichas representaciones son posteriormente transferidas al módulo de atención, donde se exploran las relaciones globales y temporales entre las ventanas de tiempo. De esta forma, la arquitectura propuesta combina la capacidad de las redes convolucionales para capturar información espacial con la potencia de los bloques Transformer o Conformer para modelar dependencias secuenciales de largo alcance, integrando así una representación más completa del habla emocional.

5.4 Módulo Transformer/Conformer

La intención principal de desarrollar una arquitectura híbrida fue diseñar un modelo capaz de extraer simultáneamente características locales y globales a partir de los espectrogramas log-Mel, logrando así una representación integral de la información contenida en la señal de voz. Esta combinación busca mejorar la capacidad del modelo para realizar inferencias precisas y generalizables en el reconocimiento de emociones.

La extracción de características locales se encuentra cubierta por la red neuronal convolucional (basada en ResNet50), dado que las capas convolucionales son eficaces para capturar patrones espaciales de baja y media escala. Sin embargo, las características globales y las dependencias a largo plazo requieren de mecanismos de atención capaces de modelar relaciones entre marcos temporales distantes dentro del espectrograma.

En este contexto, el módulo Transformer/Conformer se incorpora dentro de la arquitectura propuesta como una etapa encargada de capturar dichas relaciones contextuales. El Transformer emplea mecanismos de auto-atención (*self-attention*) para modelar interacciones globales entre las características extraídas por la red convolucional, mientras que el Conformer, al integrar convoluciones y auto-atención, ofrece una alternativa más robusta al combinar sensibilidad temporal local con dependencias globales.

En las siguientes secciones se describe en detalle la estructura del Transformer implementado, los fundamentos del mecanismo de auto-atención multi-cabeza (*Multi-Head Self-Attention*), así como la adaptación del bloque Conformer como posible sustituto o complemento dentro del modelo híbrido propuesto.

5.4.1 Mecanismos de atención

En toda arquitectura basada en Transformers, el componente fundamental es el mecanismo de atención, denominado formalmente mecanismo de auto-atención multi-cabeza (MHSA, por sus siglas en inglés). Este mecanismo define una función de atención, $\text{Attention}(Q, K, V)$, que establece un mapeo entre una consulta (Q) y un conjunto de pares llave–valor (K, V) para generar una salida. La salida se calcula como una combinación lineal ponderada de los valores, donde cada peso representa el grado de correspondencia o similitud entre la consulta y su llave respectiva, determinado por una función de compatibilidad, generalmente el producto punto escalado [36].

A este tipo de atención se le conoce como Atención de Producto Punto Escalada, esta se ve en la Figura 5.8. En este mecanismo, las entradas consisten en consultas y llaves de dimensión

d_k , y valores de dimensión d_y . Se lleva a cabo el producto punto de las consultas con todas las llaves y, posteriormente, se divide entre $\sqrt{d_k}$ para estabilizar los gradientes. Finalmente, se aplica la función de activación Softmax para obtener las ponderaciones de atención, las cuales se utilizan para realizar una combinación lineal de los valores. El proceso completo se describe mediante la siguiente expresión:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5.54)$$

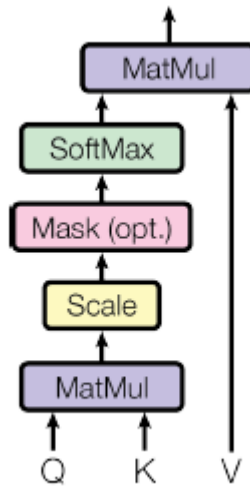


Figura 5.8. Atención de Producto Punto Escalado.

La intención de este tipo de arquitectura es no utilizar una sola función de atención, sino múltiples funciones de atención en paralelo (con h cabezas), cada una con proyecciones lineales diferentes de los tensores de entrada Q , K y V . A este esquema se le denomina Atención Multi-Cabeza, $MultiHead(Q, K, V)$, y su funcionamiento general se ilustra en la Figura 5.9. Cada cabeza de atención puede enfocarse en distintos tipos de relaciones o patrones dentro de los datos, lo que permite al modelo representar de manera más completa las dependencias entre elementos.

El mecanismo multi-cabeza se define de la siguiente manera:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (5.55)$$

Donde cada cabeza individual realiza la operación de atención:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (5.56)$$

Las matrices de proyección son parámetros entrenables del modelo y se definen como:

$$W_i^Q \in \mathbb{R}^{d_{model} \times d_k}, W_i^K \in \mathbb{R}^{d_{model} \times d_k}, W_i^V \in \mathbb{R}^{d_{model} \times d_v}, W^O \in \mathbb{R}^{hd_v \times d_{model}} \quad (5.57)$$

Cada cabeza utiliza dimensiones reducidas ($d_k = d_v = d_{model}/h$) para mantener el costo computacional equivalente al de una cabeza con la dimensión completa. Este esquema permite que la dimensionalidad total del modelo (d_{model}) se divida entre las h cabezas de atención, conservando la eficiencia computacional y aumentando la capacidad representativa del modelo.

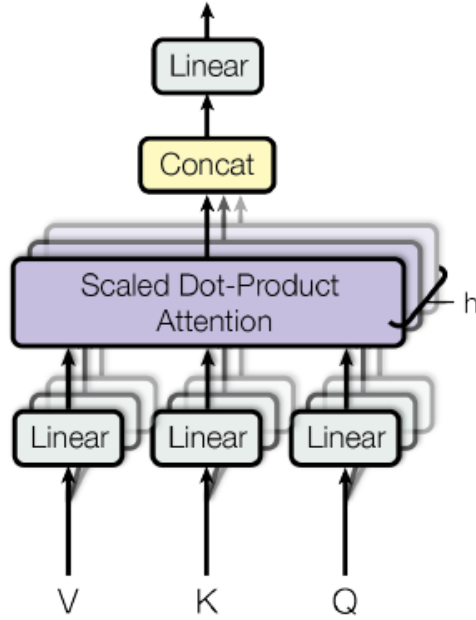


Figura 5.9. Atención Multi-Cabeza.

Para la arquitectura propuesta, se emplearon $h = 4$ cabezas de atención paralelas, cada una con $d_k = d_v = 64$, de modo que $d_{model} = 256$. Este esquema permite que cada cabeza aprenda dependencias distintas dentro de los mapas de características obtenidos a partir de la red

neuronal convolucional, capturando patrones locales y globales de la señal del habla que son relevantes para la identificación de emociones.

5.4.2 Red Totalmente Conectada

Otra parte fundamental de una arquitectura Transformer es la red totalmente conectada hacia-adelante (*Feed-Forward Network*, FNN). Esta red tiene como fin el proyectar las representaciones obtenidas por el mecanismo de atención hacia un espacio de mayor dimensión.

La red está compuesta por dos transformaciones lineales separadas por una activación no lineal ReLU en la versión básica de la arquitectura. Podemos representar su funcionamiento con la siguiente expresión:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (5.58)$$

Donde W_1 , W_2 son matrices de pesos entrenables, y b_1 , b_2 son los respectivos vectores de sesgo. La primera proyección $(xW_1 + b_1)$ expande la dimensionalidad de la representación, mientras que la segunda proyección la reduce nuevamente a la dimensión original, preservando el flujo de información.

Aunque las transformaciones lineales son iguales para todas las posiciones, los parámetros son distintos en cada capa del modelo. En términos de implementación, este bloque puede entenderse también como dos convoluciones con tamaño de kernel igual a 1.

En esta investigación, la dimensionalidad de entrada y salida se define como $d_{model} = 256$, mientras que la capa de entrada tiene una dimensionalidad expandida de $d_{ff} = 512$. Este diseño permite mantener un equilibrio entre capacidad de representación y eficiencia computacional, mejorando la generalización del modelo en la tarea de reconocimiento de emociones por medio del habla.

5.4.3 Arquitectura del bloque Transformer/Conformer

La arquitectura del bloque Transformer está compuesta principalmente por dos submódulos fundamentales:

1. El mecanismo de atención multi-cabeza.
2. La red totalmente conectada hacia-adelante.

El submódulo de atención está configurado con cuatro cabezas de atención y una dimensionalidad de representación de 256, lo que permite capturar relaciones globales entre las características extraídas por la red convolucional. Después del bloque de atención se aplican las operaciones de Dropout y normalización por capas (*LayerNormalization*), además de una conexión residual que favorece la estabilidad durante el entrenamiento.

Posteriormente, la salida se procesa mediante la red hacia-adelante compuesta por dos capas densas con 512 y 256 neuronas respectivamente, separadas por una función de activación GELU. Esta red también incorpora Dropout y una conexión residual adicional seguida de normalización por capas, se observa la arquitectura en la Figura 5.9. En conjunto, ambos submódulos constituyen un bloque Transformer completo, donde la secuencia de operaciones permite refinar las representaciones aprendidas y preservar la coherencia contextual de la información a lo largo de las dimensiones del modelo.

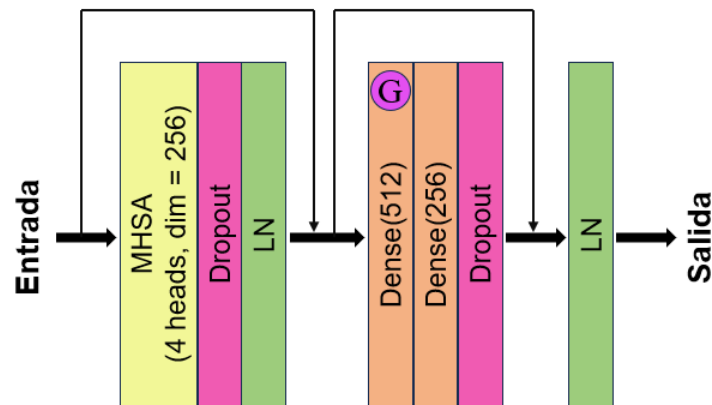


Figura 5.10. Bloque Transformer desarrollado.

Por otro lado, la arquitectura del bloque Conformer se inspira en el trabajo propuesto por Anmol Gulati et al. [35], y se implementó siguiendo de forma íntegra su diseño original. Este bloque extiende la estructura del Transformer al incorporar un componente convolucional, lo

cual permite capturar tanto las dependencias globales (a través de la atención) como las relaciones locales (mediante convoluciones).

El bloque Conformer está conformado por cuatro etapas principales:

1. Una red totalmente conectada hacia-adelante inicial.
2. Un bloque de auto-atención multi-cabeza.
3. Una sección convolucional.
4. Finalmente, una red totalmente conectada hacia-adelante.

Las etapas con redes totalmente conectadas comparten la misma configuración: están formadas por dos capas densas con 512 y 256 neuronas, separadas por una función de activación GeLU y acompañadas de regularización Dropout en cada capa densa. En la primera etapa, además, se incluye una normalización por capas al inicio, lo cual contribuye a una mejor propagación del gradiente y a la estabilidad del modelo durante el entrenamiento.

La segunda etapa corresponde al mecanismo de atención, configurado con cuatro cabezas de atención y una dimensionalidad de representación de 256, compartiendo los mismos parámetros que el bloque Transformer. Esta etapa también incorpora regularización Dropout y normalización por capas al inicio y al final.

La tercera etapa corresponde a la sección convolucional, la cual está compuesta por una convolución 1×1 con 512 filtros, seguida de una convolución separable en profundidad con un kernel de tamaño 31×31 y activación Swish. Posteriormente, se aplica una normalización por lotes, una convolución 1×1 con 256 filtros con regularización Dropout y, finalmente, una normalización por capas.

En conjunto, esta arquitectura híbrida combina la capacidad del Transformer para modelar relaciones de largo alcance con la sensibilidad temporal y local que aportan las convoluciones,

esperando que así se obtendrá una representación más completa y robusta de las características acústicas del habla emocional.

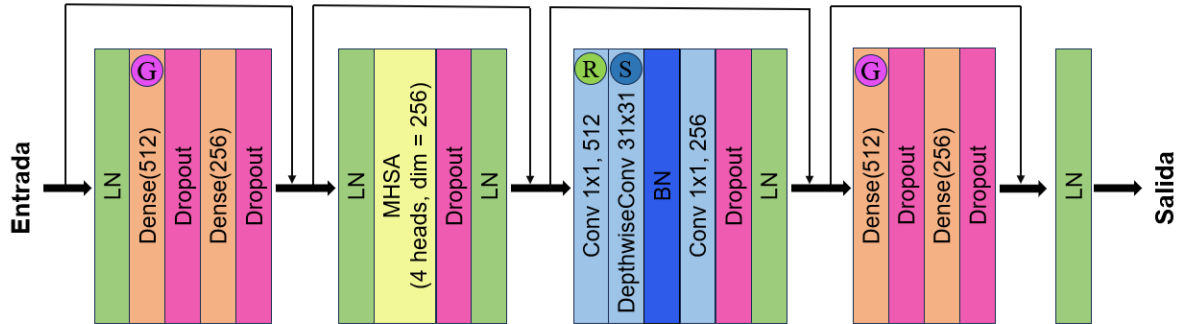


Figura 5.11. Bloque Conformer desarrollado.

5.5 Bloque de clasificación

La última sección de la arquitectura se denominó como el bloque de clasificación, este bloque recibe como entrada la salida del bloque Transformer/Conformer para posteriormente llevar a cabo el proceso de clasificación. Podemos representar matemáticamente la entrada de la siguiente forma:

$$Y_{Tran/Conf} = \in \mathbb{R}^{(T) \times d} \quad (5.59)$$

Donde $T = H * W$ corresponde a la longitud de la secuencia obtenida tras aplanar la salida espacial de la red convolucional, y $d = 2048$ es la dimensionalidad del embedding de cada token.

El bloque de clasificación está compuesto por tres operaciones: una operación de agrupación por promedio, regularización Dropout y una capa densa con activación Softmax cuya salida corresponde a las probabilidades asociadas a cada clase. Formalmente, este proceso puede describirse como:

$$Y_{AP} = AveragePooling(Y_{Tran/Conf}) \quad (5.60)$$

$$Y_{ClasePred} = Softmax(WY_{AP} + b) \quad (5.61)$$

Donde $W \in \mathbb{R}^{C \times 2048}$ y $b \in \mathbb{R}^C$ son los parámetros entrenables de la capa densa, y C corresponde al número de clases. Gráficamente se puede observar el bloque en la Figura 5.12.

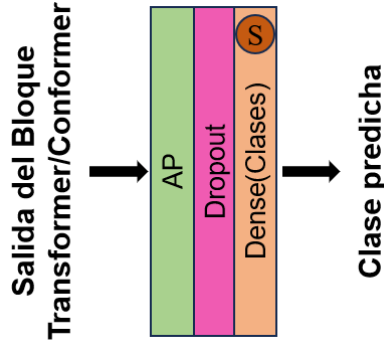


Figura 5.12. Bloque de clasificación.

La salida de este bloque consiste en un vector de dimensión C , donde cada componente del vector corresponde a la probabilidad asignada por el modelo a cada una de las clases. Estos valores son números reales en el intervalo $[0,1]$ y su suma es igual a 1, ya que se obtienen a partir de una función Softmax. La clase predicha será aquella cuyo valor de probabilidad sea el mayor dentro del vector resultante. Formalmente, la salida puede representarse como:

$$Y_{ClasePred} \in \mathbb{R}^C \quad (5.62)$$

En este capítulo se presentó de manera detallada la arquitectura propuesta para la tarea de reconocimiento de emociones por medio del habla. Se describió primero la naturaleza de los datos de entrada, compuestos por espectrogramas log-Mel calculados a partir de la señal de audio mediante la STFT y el posterior mapeo a la escala Mel.

Posteriormente, se explicó la sección convolucional basada en la arquitectura ResNet50, destacando el papel de sus bloques residuales para la extracción jerárquica de características espectro-temporales relevantes. También se describieron las dos alternativas consideradas para la parte secuencial del modelo: el bloque Transformer y el bloque Conformer. En ambos casos se detallaron sus componentes internos (mecanismos de auto-atención multicabeza, capas totalmente conectadas y, en el caso del Conformer, la incorporación de convoluciones) así como

la función que desempeñan en la modelación de dependencias locales y globales dentro de los mapas de características.

Finalmente, se presentó el bloque de clasificación, compuesto por una operación de agrupación por promedio, una etapa de regularización y una capa final totalmente conectada con activación Softmax, encargada de producir las probabilidades asociadas a cada una de las clases emocionales.

En conjunto, este capítulo establece formalmente la estructura completa del modelo propuesto, desde la representación de las señales de entrada hasta la producción de la predicción final. En el capítulo siguiente se evaluará el desempeño de esta arquitectura bajo distintos escenarios experimentales con el fin de analizar su eficacia en la tarea de reconocimiento de emociones en el habla.

VI. RESULTADOS Y DISCUSIÓN

Este capítulo documenta los resultados obtenidos de las pruebas cuantitativas de la arquitectura híbrida propuesta. El capítulo se divide de la siguiente manera: la sección 6.1 presenta los resultados obtenidos con el conjunto de datos MEACorpus, La sección 6.2 presenta los resultados obtenidos con el conjunto de datos EmoMatch y la sección 6.3 presenta los resultados de las pruebas de significancia estadística. Finalmente, la sección 6.4 hace una comparación de los resultados obtenidos con el estado del arte.

Una vez definida la arquitectura del modelo propuesto, se llevaron a cabo evaluaciones cuantitativas con el objetivo de analizar su capacidad para clasificar emociones a partir de señales de habla. Para ello, se utilizó la validación cruzada como estrategia de evaluación. Adicionalmente, con el fin de reducir el sobreajuste y mejorar la capacidad de generalización del modelo, se incorporó una técnica de aumento de datos denominada SpecAugment [37].

SpecAugment es una técnica de aumento de datos aplicada directamente sobre los espectrogramas, y su objetivo es permitir que el modelo aprenda características robustas que sean invariantes ante deformaciones temporales y pérdidas parciales de información en la representación tiempo–frecuencia. La técnica propone tres transformaciones principales:

- Deformación en el tiempo (Time Warping)
- Enmascaramiento en frecuencia (Frequency Masking)
- Enmascaramiento en el tiempo (Time Masking)

En este trabajo se emplearon únicamente enmascaramientos en frecuencia y enmascaramientos en el tiempo, debido a que la deformación en el tiempo no mostró un beneficio claro para los espectrogramas utilizados ya que aumenta la duración temporal de los datos, mientras que los modelos requieren entradas de tamaño fijo, además de implicar mayor costo computacional.

El procedimiento consistió en duplicar el conjunto de entrenamiento: se mantuvieron las muestras originales y, para cada una de ellas, se generó una versión aumentada con enmascaramientos aleatorios en frecuencia y tiempo. Sea un espectrograma log-Mel representado como:

$$S_{logMel} \in \mathbb{R}^{F \times T} \quad (6.1)$$

donde F es el número de bandas de frecuencia y T el número de cuadros temporales. Los enmascaramientos se aplicaron seleccionando intervalos aleatorios de longitud f y t , comenzando en posiciones también aleatorias f_0 y t_0 . Estos intervalos se anularon mediante:

$$S_{logMel}[f_0:f_0 + f, :] = 0, \quad S_{logMel}[:, t_0:t_0 + t] = 0 \quad (6.2)$$

Este procedimiento fuerza al modelo a no depender exclusivamente de regiones específicas del espectrograma, promoviendo la extracción de características más generales y robustas.

6.1. Resultados con Spanish MEACorpus 2023

Se puso a prueba las dos variantes del modelo propuesto inicialmente con Spanish MEACorpus 2023. Para validar el desempeño se llevó a cabo un proceso de validación cruzada con los conjuntos de datos aumentada separando en 5 particiones (o *folds*) distintos y llevando a cabo entrenamientos con cada partición. De esta manera, se puede evaluar que tan robusto es el modelo, así como su capacidad de generalización.

En primer lugar, se evaluó la variante basada en la arquitectura ResNet50-Conformer. La configuración del bloque Conformer se definió con cuatro cabezas de atención, una dimensión de embebido de 64 por cabeza y una red totalmente conectada de dimensionalidad de 512. En cuanto a los hiperparámetros de entrenamiento, se empleó un tamaño de lote de 16, una tasa de aprendizaje de 0.0001, el optimizador Adam, un máximo de 300 épocas, la función de pérdida de entropía cruzada categórica y un suavizado de etiquetas de 0.1.

La Tabla 6.1 presenta los resultados obtenidos para cada una de las cinco particiones consideradas, así como el promedio final de las métricas de desempeño evaluadas.

Tabla 6.1. Resultados de validación cruzada en ResNet50-Conformer con MEACorpus.

	Partición 1	Partición 2	Partición 3	Partición 4	Partición 5	Promedio resultados
Exactitud	0.9479	0.9288	0.9254	0.9200	0.9415	0.9327
Perdida	0.6086	0.6576	0.6690	0.6740	0.6310	0.6480
Medida-F1	0.9482	0.9289	0.9244	0.9205	0.9415	0.9327
Precisión	0.9491	0.9293	0.9250	0.9216	0.9422	0.9334
Sensibilidad	0.9479	0.9288	0.9254	0.9200	0.9415	0.9327
BAM	0.9537	0.9325	0.9172	0.9277	0.9450	0.9352

Como se observa en la Tabla 6.1, el modelo ResNet50-Conformer presenta un desempeño consistente a lo largo de las distintas particiones, alcanzando valores promedio cercanos a 0.93 en exactitud y *F1-score*, así como una métrica BAM de 0.9352. Estos resultados indican una baja variabilidad entre las particiones, lo que sugiere una buena estabilidad y capacidad de generalización del modelo propuesto.

Entre las particiones evaluadas, la partición 1 destaca por presentar los mejores resultados en todas las métricas consideradas. Por esta razón, dicha partición fue seleccionada para la obtención del modelo definitivo entrenado bajo la arquitectura ResNet50-Conformer. Como se aprecia en la Figura 6.1, el proceso de entrenamiento correspondiente a esta partición muestra un comportamiento estable, con una diferencia reducida entre las curvas de entrenamiento y validación, lo cual indica la ausencia de sobreajuste significativo.

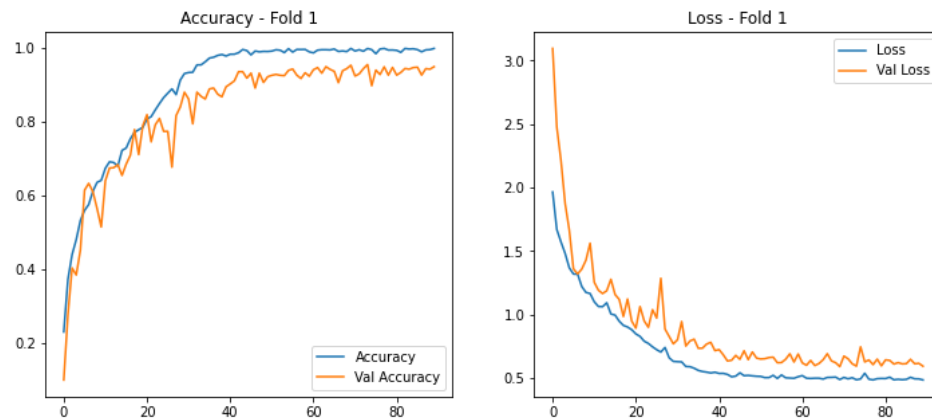


Figura 6.1. Gráficas de desempeño en el entrenamiento del modelo ResNet50-Conformer con MEACorpus.

En la Figura 6.2 se presenta la matriz de confusión correspondiente a la primera partición, en esta matriz se observa un desempeño global alto y balanceado del modelo. En particular, la clase minoritaria es clasificada correctamente en la totalidad de sus muestras, esto refleja en el valor de la métrica BAM, ya que esta penaliza con mayor severidad los errores de clasificación en clases con menor representación. Si bien en el resto de las clases se presentan algunos errores de clasificación, en su gran mayoría las muestras son clasificadas correctamente, lo cual se refleja en la diagonal principal prominente de la matriz.

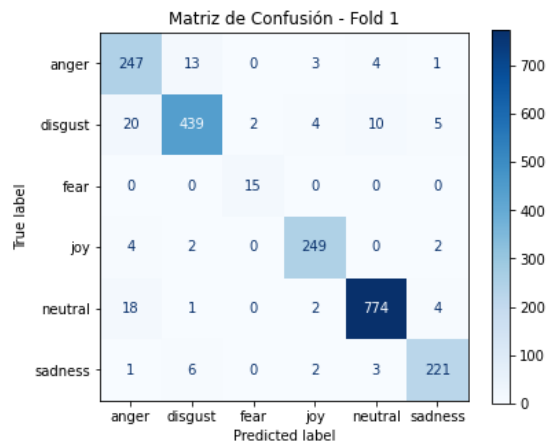


Figura 6.2. Matriz de confusión de la partición uno en ResNet50-Conformer con MEACorpus.

Posteriormente, se evaluó la segunda variante del modelo propuesto, basada en la arquitectura ResNet50-Transformer. La configuración del bloque Transformer fue similar a la utilizada en el Conformer, empleando cuatro cabezas de atención, una dimensión de embebido de 64 por cabeza y una red totalmente conectada con una dimensionalidad de 512. Los hiperparámetros de entrenamiento se mantuvieron constantes con el fin de asegurar una comparación justa entre ambas variantes.

Los resultados obtenidos mediante validación cruzada para esta arquitectura se muestran en la Tabla 6.2.

Tabla 6.2. Resultados de validación cruzada en ResNet50-Transformer con MEACorpus.

	Partición 1	Partición 2	Partición 3	Partición 4	Partición 5	Promedio resultados
Exactitud	0.9522	0.9381	0.9469	0.9410	0.9395	0.9436
Perdida	0.5911	0.6251	0.6016	0.6168	0.6208	0.6110
Medida-F1	0.9520	0.9379	0.9469	0.9408	0.9397	0.9435
Precisión	0.9523	0.9384	0.9470	0.9419	0.9404	0.9440
Sensibilidad	0.9522	0.9381	0.9469	0.9410	0.9395	0.9436
BAM	0.9387	0.9353	0.9480	0.9428	0.9407	0.9411

Los resultados obtenidos con la arquitectura ResNet50-Transformer son similares a los de su contraparte que utiliza al bloque Conformer. Sin embargo, presentan valores promedio ligeramente superiores, con métricas cercanas a 0.94 y un valor de BAM promedio de 0.9411. A pesar de que la partición 1 es la que obtiene los mejores resultados en la mayoría de las métricas evaluadas, se seleccionó la partición 3 como la versión definitiva del modelo entrenado, ya que se le otorgó mayor importancia a la métrica BAM, siendo esta partición la que alcanza el valor más alto con 0.9480.

La Figura 6.3 muestra las gráficas de desempeño durante el entrenamiento correspondiente a la partición 3, donde se observa un proceso de entrenamiento estable, con un bajo nivel de sobreajuste, y un comportamiento similar al presentado en el caso anterior.

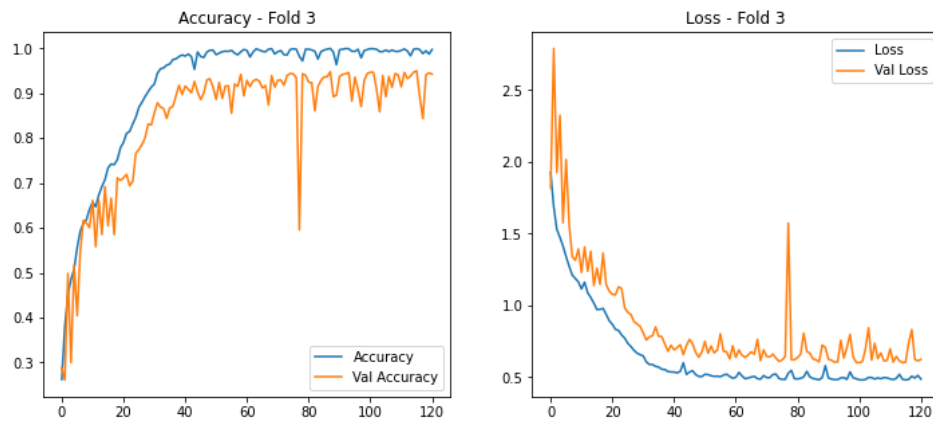


Figura 6.3. Gráficas de desempeño en el entrenamiento del modelo ResNet50-Transformer con MEACorpus.

En la Figura 6.4 se presenta la matriz de confusión obtenida a partir de la partición 3. Al igual que en el caso anterior, el modelo clasifica correctamente la totalidad de las muestras correspondientes a la clase minoritaria; sin embargo, el desempeño en el resto de las clases es ligeramente inferior. Como consecuencia, los valores de la métrica BAM y del resto de las métricas de desempeño resultan menores en comparación con la variante final basada en el bloque Conformer.

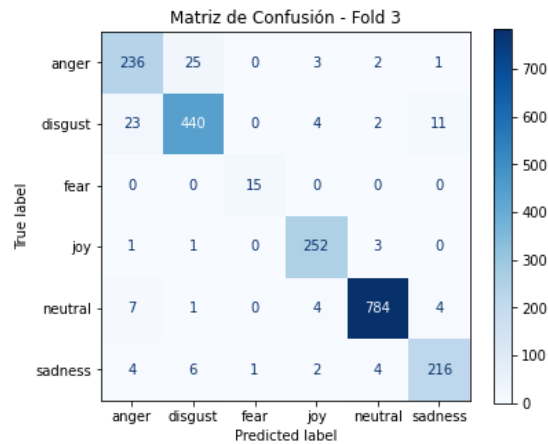


Figura 6.4. Matriz de confusión de la partición uno en ResNet50-Transformer con MEACorpus.

6.2. Resultados con EmoMatchSpanishDB

Una vez obtenidos los resultados con el conjunto de datos Spanish MEACorpus 2023, se evaluaron los modelos utilizando el conjunto de datos EmoMatchSpanishDB. El esquema experimental presentó un cambio principal respecto al caso anterior; si bien se continuó empleando la técnica SpecAugment para el aumento de datos, se mantuvieron los mismos parámetros de los modelos y los mismos hiperparámetros de entrenamiento, en esta etapa se utilizaron los pesos del mejor modelo obtenido en cada una de las dos variantes de la arquitectura como punto de inicialización (o semilla para los pesos). En particular, se emplearon los pesos correspondientes a la partición 1 en el modelo ResNet50-Conformer y a la partición 3 en el modelo ResNet50-Transformer, a partir de los cuales se realizó un reentrenamiento utilizando el conjunto de datos EmoMatch.

Al igual que con el conjunto de datos anterior, se llevó a cabo inicialmente el entrenamiento del modelo ResNet50-Conformer, cuyos resultados se presentan en la Tabla 6.3.

Tabla 6.3. Resultados de validación cruzada en ResNet50-Conformer con EmoMatch.

	Partición 1	Partición 2	Partición 3	Partición 4	Partición 5	Promedio resultados
Exactitud	0.8142	0.8566	0.8653	0.8342	0.8392	0.8419
Perdida	0.9676	0.8772	0.8562	0.9354	0.9186	0.9110
Medida-F1	0.8138	0.8563	0.8648	0.8331	0.8397	0.8415
Precisión	0.8189	0.8582	0.8664	0.8393	0.8437	0.8453
Sensibilidad	0.8142	0.8566	0.8653	0.8342	0.8392	0.8419
BAM	0.7919	0.8484	0.8479	0.8151	0.8363	0.8379

En la Tabla 6.3 se observa que los resultados obtenidos con el conjunto de datos EmoMatch son inferiores en comparación con los obtenidos con MEACorpus. Esto no implica que los resultados sean deficientes como tal, sino que puede atribuirse a ciertas características propias del conjunto de datos. En particular, EmoMatch cuenta con una cantidad considerablemente menor de muestras en relación con MEACorpus, además de que la duración promedio de los archivos de audio emocional es más reducida. Adicionalmente, al tratarse de audios obtenidos a partir de hablantes no actores en un entorno controlado, se pierde parte de la naturalidad emocional que puede resultar útil para que el modelo aprenda características distintivas de cada emoción. Como consecuencia de estos factores, se obtienen valores promedio de exactitud y F1-score cercanos a 0.84, así como un valor promedio de BAM de 0.8379.

La partición que obtuvo un mayor valor de la métrica BAM, y a la cual se le atribuye el mejor desempeño, fue la partición 2. Las gráficas de desempeño correspondientes a esta partición durante el entrenamiento se muestran en la Figura 6.5. Tal como se aprecia en dichas gráficas, el proceso de entrenamiento con este conjunto de datos presenta un comportamiento menos favorable en comparación con MEACorpus, ya que, al disponer de una menor cantidad de datos para el entrenamiento, el modelo tiende a un mayor sobreajuste. Como resultado, se observa una mayor separación entre las curvas de entrenamiento y validación.



Figura 6.5. Gráficas de desempeño en el entrenamiento del modelo ResNet50-Conformer con EmoMatch.

La Figura 6.6 muestra la matriz de confusión obtenida durante la validación de la partición 2. En este conjunto de datos, las clases continúan siendo desbalanceadas; sin embargo, presentan un mayor grado de balance en comparación con MEACorpus, por lo que la clase minoritaria no difiere de forma tan marcada respecto al resto. En EmoMatch se consideran siete clases emocionales, como se describió previamente; aun así, la matriz de confusión evidencia una clasificación adecuada, ya que, aunque se presentan errores en todas las clases, estos son mínimos. La diagonal principal claramente definida permite observar que la mayoría de las muestras son clasificadas correctamente.

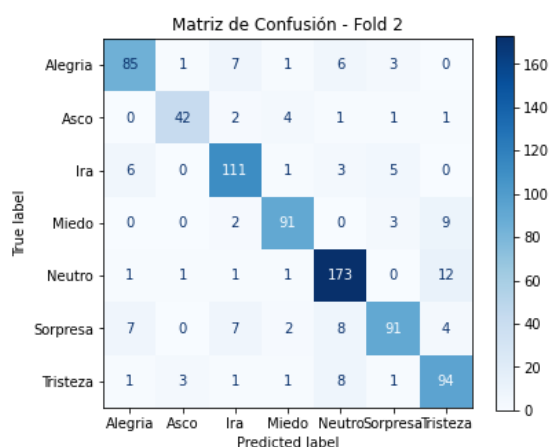


Figura 6.6. Matriz de confusión de la partición uno en ResNet50-Conformer con EmoMatch.

Una vez obtenidos los resultados con la arquitectura ResNet50-Conformer, se evaluó la variante basada en ResNet50-Transformer, cuyos resultados se presentan en la Tabla 6.4. En este caso, la variante ResNet50-Transformer obtiene valores promedio ligeramente superiores, aunque consistentes con los de la versión basada en Conformer. En particular, la exactitud y el F1-score presentan valores cercanos a 0.85, mientras que el valor promedio de la métrica BAM a lo largo de las particiones es de 0.8414.

Tabla 6.4. Resultados de validación cruzada en ResNet50-Transformer con EmoMatch.

	Partición 1	Partición 2	Partición 3	Partición 4	Partición 5	Promedio resultados
Exactitud	0.8367	0.8703	0.8703	0.8479	0.8666	0.8583
Perdida	0.8777	0.8234	0.7971	0.8584	0.8310	0.8375
Medida-F1	0.8353	0.8697	0.8696	0.8473	0.8664	0.8577
Precisión	0.8383	0.8715	0.8714	0.8496	0.8686	0.8599
Sensibilidad	0.8367	0.8703	0.8703	0.8479	0.8666	0.8583
BAM	0.8159	0.8536	0.8533	0.8310	0.8532	0.8414

La partición con mejores resultados corresponde a la partición 2, la cual alcanza un valor de BAM de 0.8536. Tal como se observa en la Figura 6.7, las curvas de entrenamiento asociadas a esta partición presentan un comportamiento similar al observado en la otra variante del modelo. Las causas del sobreajuste apreciado en las gráficas pueden atribuirse a los mismos factores previamente mencionados, principalmente a la limitada cantidad de datos disponibles para el entrenamiento.

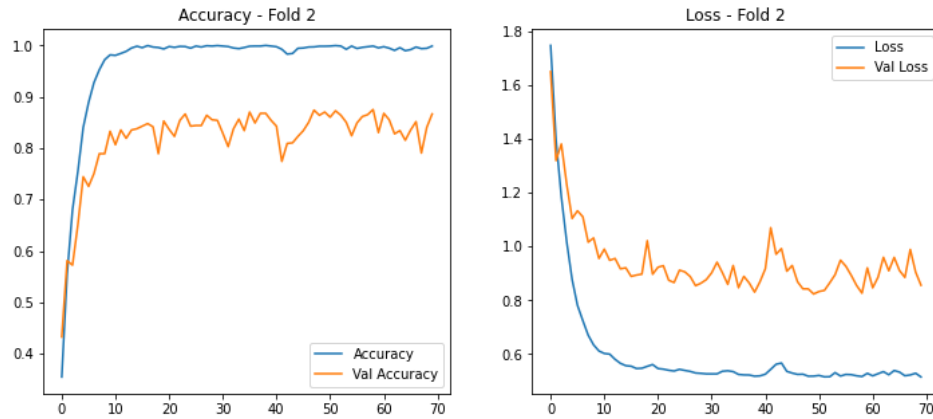


Figura 6.7. Gráficas de desempeño en el entrenamiento del modelo ResNet50-Transformer con EmoMatch.

La matriz de confusión correspondiente a esta variante se presenta en la Figura 6.8. Al igual que en el caso anterior, la diagonal principal se encuentra claramente definida, lo que denota la tendencia del modelo a clasificar correctamente la mayoría de las muestras. Asimismo, se observa una menor cantidad de errores en comparación con el modelo anterior, lo cual es coherente con los valores de la métrica BAM obtenidos.

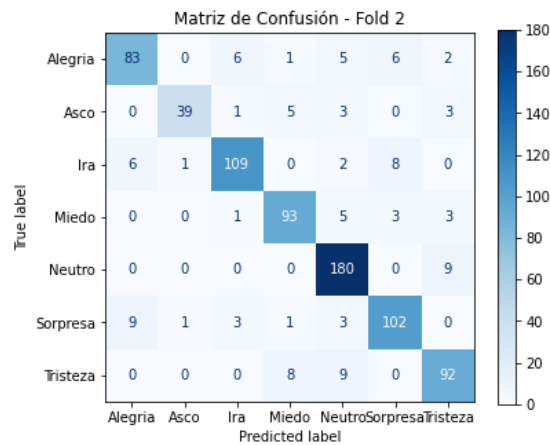


Figura 6.8. Matriz de confusión de la partición uno en ResNet50-Transformer con EmoMatch.

6.3. Pruebas de significancia estadística

Con el fin de complementar el análisis cuantitativo, se realizaron pruebas de significancia estadística sobre las predicciones obtenidas en cada una de las particiones, empleando la prueba

no paramétrica de Wilcoxon Signed-Rank. La Tabla 6.5 resume los resultados obtenidos para ambas arquitecturas y para ambos conjuntos de datos.

Como se observa, en todos los casos, tanto para la arquitectura ResNet50-Conformer como para ResNet50-Transformer, y en ambos conjuntos de datos, los valores promedio de valor- p obtenidos indican que no existe una diferencia estadísticamente significativa entre las predicciones del modelo y las etiquetas reales ($p > 0.05$), lo que sugiere un comportamiento consistente y estable de los modelos. Estos resultados respaldan la confiabilidad de las métricas de desempeño reportadas y confirman que el desempeño observado no es producto del azar.

Tabla 6.5. Resultados de las pruebas de significancia estadística.

Arquitectura	Conjunto de datos	Valor-p promedio	Conclusión
ResNet50-Conformer	MEACorpus	0.1902	Sin diferencia significativa
ResNet50-Transformer	MEACorpus	0.6036	Sin diferencia significativa
ResNet50-Conformer	EmoMatch	0.3952	Sin diferencia significativa
ResNet50-Transformer	EmoMatch	0.8347	Sin diferencia significativa

De manera adicional, se observó que las predicciones del modelo presentan un alto grado de correlación con las etiquetas reales, con coeficientes de correlación de Pearson superiores a 0.8 en la mayoría de las particiones, lo que refuerza la consistencia del comportamiento del modelo.

6.4. Comparación con modelos del estado del arte

Con el fin de contextualizar los resultados obtenidos, se realizó una comparación con el estado del arte, considerando trabajos de investigación que emplean los mismos conjuntos de datos. En la Tabla 6.6 se presentan distintos estudios que utilizaron el conjunto de datos MEACorpus, comparando sus resultados con el modelo que obtuvo el mejor desempeño en este trabajo.

A partir de esta comparación, se observa que el modelo ResNet50-Conformer propuesto alcanza un desempeño competitivo frente a las arquitecturas reportadas en la literatura, superando a la mayoría de los trabajos considerados y obteniendo la mayor medida F1 dentro de los estudios analizados para este conjunto de datos.

Tabla 6.6. Comparación con el estado del arte con MEACorpus.

Cita del artículo	Característica utilizada	Arquitectura utilizada	Resultados obtenidos
[24]	MFCC	SVM, CNN, LSTM, F-W2V, J-W2V y W2VB	Medida-F1 de 61.86%, 62.99%, 64.36%, 71.47%, 77.03% y 84.48%
[38]	Audio waveform	XLSR-wav2vec 2.0	Medida-F1 89.73%
[39]	Audio y Texto	Fine-tuning Qwen-Audio usando LoRa	Medida-F1 82.48%
[40]	MFCC	Audio Recurrent Encoder	Medida-F1 71.23%
[41]	Whisper-large-v3 encoder (121 Mel filters) y Qwen encoder (80 Mel filters)	Qwen-Audio-7B-Chat y Qwen Audio encoder with Linear Projector and Gemma Model	Medida-F1 24.29% y 50.30%
Este Trabajo	Espectrograma log-Mel	ResNet50-Conformer	Medida-F1 94.82%

Por otro lado, en la Tabla 6.7 se muestra la comparación con trabajos que emplearon el conjunto de datos EmoMatch. En este caso, el modelo que presentó el mejor desempeño fue la variante ResNet50-Transformer, la cual supera los resultados reportados en el estado del arte para este conjunto de datos. Estos resultados indican que el modelo propuesto es competitivo dentro de la literatura y representa una alternativa viable para el reconocimiento de emociones en el habla.

Tabla 6.7. Comparación con el estado del arte con EmoMatch.

Cita del artículo	Características utilizadas	Arquitecturas utilizadas	Resultados obtenidos
[42]	eGeMAPS y wav2vec 2.0	Support Vector Classifier (SVC) con base radial	Sensibilidad de 54.4% con eGeMAPS y 56.9% con wav2vec 2.0
[25]	eGeMAPS, COMPARE y espectrogramas Mel	MLP CNN-LSTM	Exactitud: MLP-eGeMaps de 65% MLP-COMPARE de 69% CNN-LSTM de 65%
[43]	Large Speech Model	Audio Spectrogram Transformer	Medida-F1 de 36.1%
[44]	Espectrogramas	CNN	Precisión 41%
[1]	MFCCs con deltas, eGeMAPS y COMPARE	Support Vector Machine (SVM)	Exactitud de 64.3%, 65.4% y 65%.
Este Trabajo	Espectrogramas log-Mel	ResNet50-Transformer	Exactitud 87.03% Precisión 87.15% Sensibilidad 87.03% Medida-F1 86.97%

VII. CONCLUSIONES

Este trabajo propone el desarrollo de un modelo de aprendizaje profundo capaz de reconocer emociones a partir del habla en español. Por lo tanto, se diseñó una arquitectura híbrida basada en una red neuronal convolucional ResNet50, integrada con bloques Transformer/Conformer. Dicha arquitectura fue evaluada sobre conjuntos de datos de habla emocional en español, demostrando un desempeño adecuado en la tarea de clasificación de emociones.

Como hipótesis se planteó que un modelo híbrido, capaz de extraer características relevantes de información espectro-temporal y de capturar tanto dependencias locales como globales de los datos, resultaría adecuado para abordar la tarea propuesta. Con base en los resultados experimentales obtenidos a lo largo del Capítulo 6, se puede concluir que dicha hipótesis es válida.

La arquitectura propuesta fue evaluada utilizando los conjuntos de datos MEACorpus y EmoMatch. De acuerdo con los resultados presentados en las Tablas 6.1 y 6.2, en MEACorpus se obtuvieron valores de BAM superiores al 90 % de manera consistente. Por su parte, en las Tablas 6.3 y 6.4 se reportan valores promedio de BAM superiores al 80 % en EmoMatch. Estos resultados indican un desempeño adecuado para la correcta clasificación de las emociones. Asimismo, este comportamiento se ve reflejado en las matrices de confusión, donde se observa una tendencia clara a asignar las muestras a sus clases correspondientes, con errores mínimos.

Por otro lado, las evaluaciones realizadas sobre ambos conjuntos de datos mostraron un desempeño superior o comparable al reportado en la literatura, como se observa en las Tablas 6.6 y 6.7. Esto valida nuevamente la hipótesis planteada y confirma que el uso de espectrogramas log-Mel, en conjunto con arquitecturas modernas de aprendizaje profundo, constituye una alternativa efectiva para el reconocimiento de emociones en el habla en español.

En relación con las pruebas de significancia estadística presentadas en la Tabla 6.5, los resultados permiten confirmar que el aprendizaje del modelo es robusto y consistente al no haber significancia estadística entre las predicciones del modelo y las etiquetas reales.

Entre las principales aportaciones de este trabajo se encuentra el diseño de una arquitectura basada en ResNet50 combinada con arquitecturas tipo Conformer y Transformer, así como su evaluación exhaustiva en dos bases de datos de referencia. Los resultados obtenidos evidencian que el modelo propuesto puede competir e incluso superar a varias aproximaciones reportadas en el estado del arte, incluyendo modelos de mayor complejidad, manteniendo un equilibrio adecuado entre desempeño y eficiencia.

No obstante, este estudio presenta algunas limitaciones. La evaluación se realizó únicamente sobre dos conjuntos de datos específicos, por lo que la capacidad de generalización del modelo en escenarios *cross-corpus* no fue analizada. Asimismo, las comparaciones con el estado del arte se basan en métricas y protocolos reportados por distintos autores, lo cual puede introducir variaciones en la comparabilidad de los resultados. Adicionalmente, el análisis se centró en un entorno offline, sin considerar restricciones de tiempo real.

Como trabajo futuro, se propone extender el enfoque hacia modelos multimodales que integren información adicional, como texto, así como evaluar el desempeño del modelo en escenarios *cross-corpus*. De igual forma, resulta de interés optimizar la arquitectura para su implementación en sistemas con recursos limitados y profundizar en el análisis de los mecanismos de atención con el objetivo de mejorar la interpretabilidad del modelo.

GLOSARIO

Acrónimo	Descripción
AP	Agrupación por Promedio
AST	Transformer de Espectrograma de Audio
b	Sesgo
BAM	Métrica de exactitud Balanceada
BiLSTM	Memoria a Corto y Largo Plazo Bidireccional
BN	Normalización por Lotes
c	Clase
c_e	Índice del coeficiente cepstral
CNN	Red Neuronal Convolutacional
d	Dimensión
DCT	Transformada Discreta del Coseno
DL	Aprendizaje Profundo
EmoMatch	Conjunto de datos EmoMatchSpanishDB
F	Bandas de frecuencia
FFNN	Red Neuronal Totalmente Conectada Hacia-Adelante
FFT	Transformada Rápida de Fourier
GELU	Unidad Lineal de Error Gaussiano
h	Salto de ventana
$H_i(k)$	Respuesta del i -ésimo filtro Mel
i	Índice de filtro Mel
IA	Inteligencia Artificial
K	Llave
k	Índice de frecuencia discreta
LN	Normalización por Capas
Log-Mel	Logarítmico de Mel
LSTM	Memoria a Corto y Largo Plazo
M	Número total de filtros Mel
m	Índice de ventana temporal
MBConv	Cuello de Botella Invertido Móvil

GLOSARIO

MEACorpus	Conjunto de datos Spanish MEACorpus 2023
MFCC	Coefficientes Cepstrales de Frecuencias Mel
MHSA	Auto-Atención Multi-Cabeza
ML	Aprendizaje Automático
MLP	Perceptrón Multicapa
MP	Agrupación por Valor Máximo
n	Índice temporal de muestras
NLP	Procesamiento del Lenguaje Natural
p	Probabilidad
Q	Consulta
R	Conexión residual con proyección
r	Mascara aleatoria
ReLU	Unidad Lineal Rectificada
RNA	Red Neuronal Artificial
RNN	Red Neuronal Recurrente
S	Espectrograma
SER	Reconocimiento de Emociones por medio del habla
S_{logMel}	Espectrograma logarítmico de Mel
S_{Mel}	Espectrograma de Mel
SVM	Máquina de Vectores de Soporte
T	Cantidad de ventanas de tiempo o longitud de secuencia
V	Valor
ViT	Vision Transformer
W	Pesos entrenables
w	Venta de Hamming
X	Tensor de salida
x	Tensor de entrada
Y	Salida de bloque
y	Etiqueta real asociada a una muestra
Z	Salida de sub-bloque
α	Tasa de aprendizaje
ε	Constante pequeña positiva

GLOSARIO

θ	Parámetros del modelo
λ_2	Hiperparámetro de regularización L2

REFERENCIAS

- [1] E. Garcia-Cuesta, A. Barba-Salvador y Gachet-Páez, «EmoMatchSpanishDB: study of speech emotion recognition,» *Multimedia Tools and Applications*, vol. 83, pp. 13093-13112, 2023.
- [2] V. B.-V. E. S. N. U. R. A. Smith K. Khare, «Emotion recognition and artificial intelligence: A systematic review (2014-2023) and research recommendations,» *Information Fusion*, p. 102019, 2024.
- [3] A. E.-M. Mohammed Jawad Al-Dujaili, «Speech Emotion Recognition: A Comprehensive Survey,» *Wireless Personal Communications*, vol. 129, pp. 2525-2561, 2023.
- [4] L. B. T. M. T. T. C. C. Felicia Andayani, «Hybrid LSTM-Transformer Model for Emotion Recognition From Speech Audio Files,» *IEEE Access*, vol. 10, pp. 36018-36027, 2022.
- [5] Q. Ouyang, «Speech Emotion Detection Based on MFCC and CNN-LSTM Architecture,» *arXiv*, 2025.
- [6] V. M. S. G. Poorna S.S., «Hybrid CNN-BiLSTM architecture with multiple attention mechanisms to enhance speech emotion recognition,» *Biomedical Signal Processing and Control*, vol. 100, nº B, p. 106967, 2025.
- [7] M. G. Javier de Lope, «An ongoing review of speech emotion recognition,» *Neurocomputing*, vol. 528, pp. 1-11, 2023.
- [8] P. M. I. Swapna Mol George, «A review on speech emotion recognition: A survey, recent advances, challenges, and the influence of noise,» *Neurocomputing*, vol. 568, p. 127015, 2024.
- [9] T. B. Bernd J. Kröger, *Neural Modeling of Speech Processing and Speech Learning*, Springer Cham, 2019.
- [10] A. R. J. Y. A. D. Md. Shah Fahad, «A survey of speech emotion recognition in natural environment,» *Digital Signal Processing*, vol. 110, p. 102951, 2021.
- [11] N. M. R. B. R. M. S. P. Ambuj Mehrish, «A review of deep learning techniques for speech processing,» *Information Fusion*, vol. 99, p. 101869, 2023.
- [12] T. Y. F. Y. H. W. Hao Meng, «Speech Emotion Recognition From 3D Log-Mel Spectrograms With Deep Learning Network,» *IEEE Access*, vol. 7, pp. 125868-125881, 2019.
- [13] M. K. Kishor Barasu Bhangale, «Survey of Deep Learning Paradigms for Speech Processing,» *Wireless Personal Communications*, vol. 125, pp. 1913-1949, 2022.
- [14] Y. L. T. D. Y. Z. J. C. Xiaoyu Tang, «Speech Emotion Recognition Via CNN-Transformer and Multidimensional Attention Mechanism,» *Speech Communication*, vol. 171, p. 103242, 2025.
- [15] V. P. Y. C. P. B. Liam Johnston, «Revisiting the problem of learning long-term dependencies in recurrent neural networks,» *Neural Networks*, vol. 183, p. 106887, 2025.

REFERENCIAS

- [16] R. G. Pooja Lalwani, «A Novel CNN-BiLSTM-GRU Hybrid Deep Learning Model for Human Activity Recognition,» *International Journal of Computational Intelligence Systems*, vol. 17, 2024.
- [17] L. G. Q. M. Mohammed Tellai, «An efficient speech emotion recognition based on a dual-stream CNN-transformer fusion network,» *International Journal of Speech Tecnology*, vol. 26, pp. 541-557, 2023.
- [18] A. T. H. W. M. S. F. B. F. E. B. W. S. JohannesWagner, «Dawn of the Transformer Era in Speech Emotion Recognition: Closing the Valence Gap,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, nº 9, pp. 10745-10759, 2023.
- [19] Y. W. X. L. X. Q. Tianyang Lin, «A survey of transformers,» *AI Open*, vol. 3, pp. 111-132, 2022.
- [20] M. I. C. Murguía, Modelos Actuales de Aprendizaje Profundo en Inteligencia Artificial, Ciudad de México: Secretaría de Educación Publica - Tecnológico Nacional de México, 2024.
- [21] P. Ekman, «Expression and the Nature of Emotion,» de *Approaches to Emotion*, Hillsdale, NJ, Lawrence Erlbaum, 1984, p. 319–344.
- [22] J. A. G.-D. M. Á. R.-G. R. V.-G. Ronghao Pan, «Spanish MEACorpus2023:A multimodal speech–text corpus for emotion analysis in Spanish from natural environments,» *Computer Standards & Interfaces*, vol. 90, p. 103856, 2024.
- [23] B. T. A. y. A. Sasou, «Multilingual, Cross-lingual, and Monolingual Speech Emotion Recognition on EmoFilm Dataset,» de *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Taipei, 2023.
- [24] G. C. A. B. A. B. B. S. Emilia Parada-Cabaleiro, «Categorical vs Dimensional Perception of Italian Emotional Speech,» *Interspeech 2018*, pp. 3638-3642, 2018.
- [25] E. G.-C. Daniel Cabrera-Rodriguez, «Sistema de Identificación de Emociones a partir de Audios en Español Basado en Redes Neuronales Multiobjetivo,» Madrid, 2023.
- [26] X. Z. S. R. J. S. Kaiming He, «Deep Residual Learning for Image Recognition,» *arXiv*, 2015.
- [27] M. L. a. M. Lee, «Performance Improvement of Speech Emotion Recognition Using ResNet Model with Data Augmentation–Saturation,» *applied sciences*, vol. 15, p. 2088, 2015.
- [28] Q. V. L. Mingxing Tan, «EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,» *arXiv*, 2019.
- [29] A. H. M. Z. A. Z. L.-C. C. Mark Sandler, «MobileNetV2: Inverted Residuals and Linear Bottlenecks,» *arXiv*, 2018.
- [30] V. V. S. I. J. S. Z. W. Christian Szegedy, «Rethinking the Inception Architecture for Computer Vision,» *arXiv*, 2015.
- [31] A. Z. Karen Simonyan, «Very Deep Convolutional Networks for Large-Scale Image Recognition,» *arXiv*, 2015.

REFERENCIAS

- [32] L. B. A. K. D. W. X. Z. T. U. M. D. M. M. G. H. S. G. J. U. N. H. Alexey Dosovitskiy, «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,» *arXiv*, 2020.
- [33] Y.-A. C. J. G. Yuan Gong, «AST: Audio Spectrogram Transformer,» *arXiv*, 2021.
- [34] V. L. D. D. X. S. A. G. T. S. P. H. J. D. A. M. L. K. D. B. L. C. A. W. Krzysztof Choromanski, «Rethinking Attention with Performers,» *arXiv*, 2020.
- [35] J. Q. C.-C. C. N. P. Y. Z. J. Y. W. H. S. W. Z. Z. Y. W. R. P. Anmol Gulati, «Conformer: Convolution-augmented Transformer for Speech Recognition,» *arXiv*, 2020.
- [36] N. S. N. P. J. U. L. J. A. N. G. L. K. I. P. Ashish Vaswani, «Attention Is All You Need,» *arXiv*, 2017.
- [37] W. C. Y. Z. C.-C. C. B. Z. E. D. C. Q. V. L. Daniel S. Park, «SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,» *arXiv*, 2019.
- [38] F. C. M. I. J. H. Marc Casals-Salvador, «BSC-UPC at EmoSpeech-IberLEF2024: Attention Pooling for Emotion Recognition,» *arXiv*, 2024.
- [39] J. B.-S. I. M.-F. M. G.-M. L. F. D. F. F.-M. Sergio Esteban-Romero, «THAU-UPM at EmoSpeech-IberLEF2024: Efficient Adaptation of Mono-modal and Multi-modal Large Language Models for Automatic Speech Emotion Recognition,» *CEUR Workshop Proceedings*, vol. 3756, 2024.
- [40] C. M. M. C.-M. T. A. G. H. C. Miguel Soto, «CogniCIC at EmoSpeech-IberLEF2024: Exploring Multimodal Emotion Recognition in Spanish: Deep Learning Approaches for Speech-Text Analysis,» *CEUR Workshop Proceedings*, vol. 3756, 2024.
- [41] M. R.-C. M. E.-G. R. C. L. F. D. Jaime Bellver-Soler, «Multilingual Speech Emotion Recognition combining Audio and Text Information using Small Language Models,» de *IberSPEECH 2024*, Aveiro, 2024.
- [42] A. A. S. A. B. S. B. W. S. Adria Mallol-Ragolta, «Prototypical Networks for Speech Emotion Recognition in Spanish,» de *IberSPEECH 2024*, Aveiro, 2024.
- [43] J. J. M. V. G. R. A. B. Federico D'asaro, «Using Large Speech Models for Feature Extraction in Cross-Lingual Speech Emotion Recognition,» de *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*, Pisa, 2024.
- [44] Á. G.-A. Laredo, «Sistema de identificación de emociones independiente del hablante basado en redes neuronales convolucionales,» Madrid, 2023.