Framework para la Evaluación Comparativa de Modelos de Deep Learning para el Reconocimiento de Emociones a partir del Habla

Luis Antonio Bernal Chahuayo

Asesor: Dr. Alvaro Ernesto Cuno Parari

Co-asesor: Dr. Wilber Ramos Lovón

Arequipa, Perú

2021

Framework para la Evaluación Comparativa de Modelos de Deep Learning para el Reconocimiento de Emociones a partir del Habla

Por
Luis Antonio Bernal Chahuayo

Asesor: Dr. Alvaro Ernesto Cuno Parari
Co-asesor: Dr. Wilber Ramos Lovón

Tesis presentada a la
Escuela Profesional de Ciencia de la Computación
de la
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
como requisito
para obtener el título profesional
de
Licenciado en Ciencia de la Computación

A mis padres, Rosa y Luis, por el apoyo incondicional brindado para poder conseguir mis objetivos.

A mis profesores por los conocimientos, enseñanzas y oportunidades brindadas durante mi carrera de pregrado.

A mis asesores, Alvaro y Wilber, por guiarme durante el camino de elaboración de esta tesis.

Índice general

Índice de figuras

Índice de cuadros

Agradecimientos

Esta investigación ha sido financiado por el Proyecto Concytec - Banco Mundial "Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia Tecnología e Innovación Tecnológica" 8682-PE, a través de su unidad ejecutora ProCiencia. [Contrato número 014-2019-FONDECYT-BM-INC.INV]

Resumen

El reconocimiento de emociones en el habla viene siendo estudiado desde hace muchos años. Sin embargo, realizar una comparación objetiva del estado del arte es una tarea complicada debido a la variedad de datasets y métricas usadas para su evaluación.

Esta tesis presenta un framework para la evaluación y comparación de modelos de Deep Learning mediante el uso de múltiples estrategias para reducir la subjetividad en los resultados.

Para demostrar la validez del framework, la propuesta fue aplicada en el entrenamiento de modelos de Deep Learning para los dataset de emociones en el habla RAVDESS y IEMOCAP. Los experimentos mostraron la efectividad de la propuesta al conseguir identificar estadísticamente los mejores modelos.

Abstract

Speeh Emotion Recognition has been studied for many years. However, making an objective comparison of the state of the art is a complicated task due to the variety of datasets and metrics used for its evaluation.

This thesis presents a framework for the evaluation and comparison of Deep Learning models through the use of multiple strategies to reduce subjectivity in the results.

To demonstrate the validity of the framework, the proposal was applied in the training of Deep Learning models for the speech emotion datasets RAVDESS and IEMOCAP. The experiments showed the effectiveness of the proposal in being able to statistically identify the best models.

Abreviaturas

FT Transformada de Fourier

CNN Convolutional Neural Network

LSTM Long Short Term Memory

GRU Gated Recurrent Unit

MFCC Mel Frequency Cepstral Coefficients

SVM Support Vector Machine

HMM Hidden Markov Model

GMM Gaussian Mixture Model

AVEC Audio Visual Emotion Challenge

EmotiW Emotion Recognition in the Wild

SER Speech Emotion Recognition

KNN k Nearest Neighboors

Capítulo 1

Introducción

1.1. Contexto y Motivación

El reconocimiento de emociones a partir del habla es el proceso de inferir cuál es la emoción que un hablante está expresando en un momento dado. Esto se realiza únicamente tomando como entrada las ondas de sonido que se producen al hablar y que son capturadas por uno o varios micrófonos. El interés en lograr reconocer las emociones humanas ha sido motivado por varias áreas como robótica para diseñar robots inteligentes que puedan interactuar con humanos, marketing, para crear anuncios personalizados basados en el estado emocional de potenciales compradores, educación, para mejorar los procesos de aprendizaje o entretenimiento, para mostrar el contenido más adecuado a una audiencia objetivo [Dzedzickis et al., 2020].

En primer lugar, es crucial definir que es una emoción o que emociones existen. Por lo general, se define como una experiencia intensa y de corta duración y la persona generalmente es muy consciente de ella) [Feidakis et al., 2011]. La literatura en el área de psicología presentó varias teorías para modelar las emociones. Por ejemplo, entre los modelos discretos y categóricos, se encuentra *"big six"* de Ekman, que incluye las siguientes seis categorías: enojo, disgusto, miedo, felicidad y tristeza; adicionalmente se añade a menudo la categoría neutral. Otro modelo, en este caso continuo, es el modelo *"arousal-valence"*. Que permite modelar las emociones usando dos dimensiones (excitación y valencia) las cuales pueden ser positivas o negativas [Schuller, 2018].

Tradicionalmente el problema de reconocimiento de emociones a partir del habla ha sido abordado, con selección manual de características y clasificación vía algoritmos de Machine Learning. Sin embargo, tras el incremento de poder computacional y paralelismo, los métodos de Deep Learning como Redes Neuronales Convolucionales, que extraen características automáticamente, han superado en precisión a estos métodos tradicionales [Akçay and Oğuz, 2020].

Elegir los mejores métodos no es una tarea trivial, sobre todo cuando en el estado del arte observamos gran cantidad de métodos propuestos. Las métricas usadas generalmente para determinar cuan preciso y exacto es un método, dependen en gran medida del dataset utilizado, e incluso de que muestra del dataset se usó para entrenamiento y cuál para pruebas.

A esto se suma la falta de publicación de código fuente de los autores, que impide replicar los experimentos y hacer comparaciones equitativas. [Fayek et al., 2017]. Es por esto que es usual que los autores repliquen trabajos del estado del arte con la finalidad de comparar su propuesta, lo cual podría llevar a sesgar la investigación involuntariamente.

Por otro lado cabe la posibilidad que los resultados obtenidos hayan sido producto de intentar mediante fuerza bruta, forzar resultados cambiando la forma de dividir los datasets.

Una alternativa que se suele plantear para facilitar las comparaciones es la realización de benchmarks, donde se distribuye un dataset y se compite por alcanzar la mejor predicción posible. Sin embargo, esto puede llevar a conclusiones erróneas o a que solo se busque optimizar el modelo para ese dataset. Se ha visto por ejemplo, que es posible vulnerar los benchmarks, permitiendo a un modelo alcanzar el podio sin siquiera haberlo entrenado, solo observando los resultados de accuracy obtenidos. [Whitehill, 2018]

La presente tesis busca proponer un framework de evaluación y comparación de propuestas de modelos de Deep Learning para el Reconocimiento de Emociones a partir del Habla, que permitirá a los investigadores obtener resultados menos subjetivos y transparentes.

1.2. Definición del problema

Comparar modelos de Deep Learning en Reconocimiento de Emociones a partir del habla, es una tarea compleja. Las métricas de evaluación como Accuracy o Precision, son sensibles a cambios en la partición (entrenamiento y pruebas) del dataset utilizado, por lo que modelos podrían ser mejores en un experimento pero peores en otro.

Los benchmarks como alternativas son vulnerables a técnicas que solo buscan optimizar un lugar en el podio y siguen siendo afectados por la forma de ordenar el dataset.

1.3. Justificación

Actualmente, al realizar una propuesta de un modelo de Deep Learning para realizar Reconocimiento de Emociones a partir del Habla se debe valorar su precisión frente al estado del arte actual, para poder así determinar si el aporte es significativo. Sin embargo, es complicado hacer esta determinación, ya que cada investigación utiliza diferentes datasets, métricas, formas de entrenamiento y evaluación.

E incluso usando el mismo dataset y métricas los resultados podrían variar dependiendo de la partición de dataset utilizada. Por ello se propone un framework de evaluación y comparación objetiva que permite realizar comparaciones que sean independientes a la partición del dataset. De esta forma, los investigadores podrán medir sus avances en el área de forma menos subjetiva y realizar revisiones de literatura más significativas. Adicionalmente, se podría contribuir en los problemas de repetibilidad, reproducibilidad y replicabilidad de resultados, pues los procesos de entrenamiento y evaluación serán transparentes e iguales para todos los modelos participantes.

1.4. Objetivos

1.4.1. Objetivo General

Proponer un framework que permita evaluar y comparar la precisión de forma objetiva de modelos de Deep Learning aplicados al Reconocimiento de Emociones a partir del Habla.

1.4.2. Objetivos Específicos

- Identificar métricas que permitan reducir la subjetividad de la evaluación de precisión de modelos de Deep Learning.
- Identificar métodos para garantizar que los resultados de los modelos obtenidos son independientes de la partición en datos de entrenamiento y datos de pruebas del dataset.
- Evaluar la eficacia del framework propuesto en modelos existentes en el estado del arte.

Capítulo 2

Marco Teórico

2.1. El habla

2.1.1. La Voz

La voz es el sonido producido por humanos usando los pulmones y las cuerdas vocales en la laringe, o caja de voz. La voz no siempre produce habla, en cambio, los infantes pueden balbucear y los adultos reír, cantar y llorar. La voz es generada por el flujo de aire desde los pulmones mientras las cuerdas vocales están juntas. Cuando el aire es empujado por las cuerdas vocales con suficiente presión las cuerdas vibran. La voz es única como una huella digital y ayuda a definir personalidad, estado de ánimo y salud.

2.1.2. Habla

El habla es una forma de comunicación vocal usando lenguaje. Así los humanos pueden expresar pensamientos, sentimientos e ideas oralmente. Cada lenguaje usa combinaciones fonéticas de sonidos vocales y consonantes que forman los sonidos de las palabras. Estos sonidos son producidos por acciones musculares coordinadas en la cabeza, cuello, pecho y abdomen.

En el habla, los hablantes realizan una serie de actos intencionales como informar, declarar, preguntar, persuadir, dirigir, etc. Y usan la enunciación, entonación, grados

de volumen, tempo y otros aspectos que llevan al entendimiento. También de forma no intencionada, comunican aspectos sociales como sexo, edad, lugar de origen (acento), estados de físicos (alerta, sueño, fuerza, debilidad, salud o enfermedad), estados psicológicos como emociones o estados de ánimo, estados psicóticos (sobriedad o ebriedad, conciencia o estados de trance), educación y experiencia. [NIH, 2020]

2.2. Emociones

No hay un consenso sobre la definición de emoción y aún es un problema abierto en el área de psicología. Sin embargo, una definición común es la siguiente: Las emociones son estados psicológicos que están compuestos de varios componentes como experiencias personales, psicológicas, comportamientos y reacciones comunicativas.

Por lo general en el área de Speech Emotion Recognition se usan dos modelos de emociones: Modelo discreto emocional y modelo dimensional.

La teoría de emoción discreta está basado en seis categorías de emociones básicas: Tristeza, felicidad, miedo, enojo, disgusto y sorpresa. (Modelo de Ekman). Estas están presentes desde que nacemos y son culturalmente independientes. Estas emociones se presentan por un periodo corto de tiempo. Este modelo se usa en nuestro día a día para describir emociones observadas, por lo tanto la anotación de estos modelos puede ser más sencilla. Sin embargo, estas categorías no son capaces de definir estados emocionales complejos observados en la comunicación diaria.

El modelo dimensional es un modelo alternativo que usa un pequeño número de dimensiones para representar emociones (valence, arousal, power). El modelo preferido es uno de dos dimensiones: Arousal, Activación o excitación en una dimensión versus Valencia o evaluación y los rangos van de agradable a desagradable. La dimensión Arousal define la fuerza de la emoción. Este modelo tiene como desventaja como que no es intuitivo para el etiquetado de emociones en los datasets. Adicionalmente algunas de las emociones se convierten en idénticas como: Miedo y enojo, o emociones como sorpresa no pueden ser categorizadas y quedan por fuera del espacio dimensional porque la sorpresa puede ser positiva o negativa dependiendo del contexto. [Akçay and Oğuz, 2020]

2.3. Datasets de Audios

Las bases de datos en Speech Emotion Recognition se pueden clasificar en:

2.3.1. Bases de datos simuladas

En estas bases de datos, los datos del habla han sido registrados por intérpretes experimentados y bien entrenados, como actores. Esta se considera la forma más sencilla de obtener el conjunto de datos basado en el habla de varias emociones. Se calcula que casi el 60 % de las bases de datos de voz se recopilan mediante esta técnica.

2.3.2. Bases de datos inducidas

El conjunto emocional se recoge creando una situación emocional artificial. Esto se hace sin el conocimiento previo del intérprete. En comparación con la base de datos basada en actores, esta es una base de datos más natural. Sin embargo, puede caer en problemas éticos, porque el hablante debería saber que han sido grabados para actividades basadas en la investigación.

2.3.3. Bases de datos naturales

Estas bases de datos son las más difíciles de obtener debido a la dificultad de reconocimiento y la escasez de ciertas emociones. Las bases de datos de voz emocional natural se registran generalmente a partir de la conversación del público en general, las conversaciones de centros de llamadas, etc. [Khalil et al., 2019]

Las bases de datos más usadas en el estado del arte son:

- IEMOCAP: Interactive Emotional Dyadic MOtion Capture (Inglés) [Busso et al., 2008]
- RAVDESS (Inglés) [Livingstone and Russo, 2018]
- EMODB: Berlin Emotional Database (Alemán) [Burkhardt et al., 2005]
- INTERFACE05 (English, Spanish, French and Slovenian) [Martin et al., 2006]

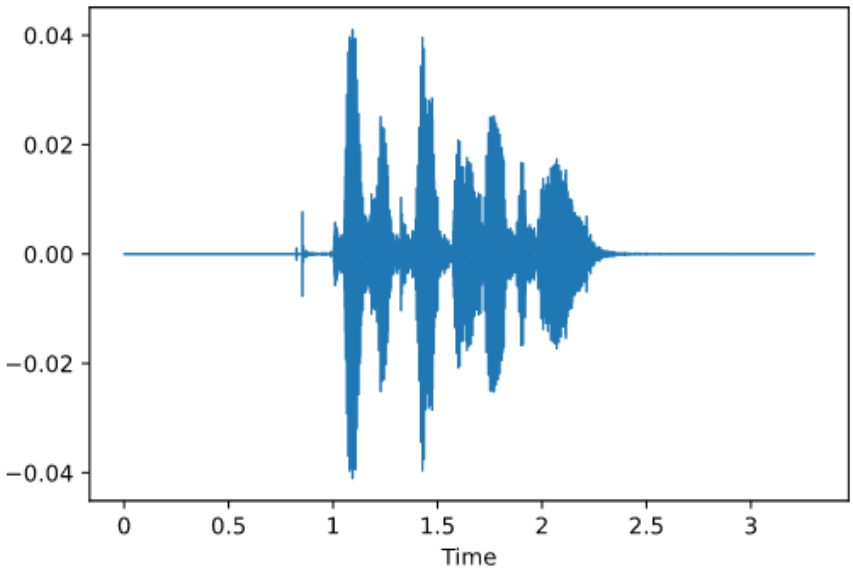

Figura 2.1: Ejemplo de una onda de una muestra de audio del dataset RAVDESS

2.4. Procesamiento de Audio

Los audios de los dataset son digitales, y se toman con frecuencias de muestreo distintas, por lo general 32 kHz. A pesar de ser muestras discretas pueden tratarse como una onda, por lo que el procesamiento de señales es aplicable. Específicamente, la transformada de Fourier.

La transformada de Fourier nos permite analizar el audio en el espectro de frecuencias, lo que nos da características que pueden ser analizadas de forma más simple.

La Transformada de Fourier (FT) se define como:

$$\hat{f}(\xi) := \int_{-\infty}^{\infty} f(x)\ e^{-2\pi i \xi x}\, dx$$

Donde:

- $f(x)$ es una función integrable (La onda)
- ξ es un número real (Frecuencia)

A partir de $\hat{f}(\xi)$ podemos construir un espectrograma. Que es el resultado de aplicar FT en ventanas de tiempo. Más específicamente la Transformada Discreta Rápida de Fourier. Fig. 2.2

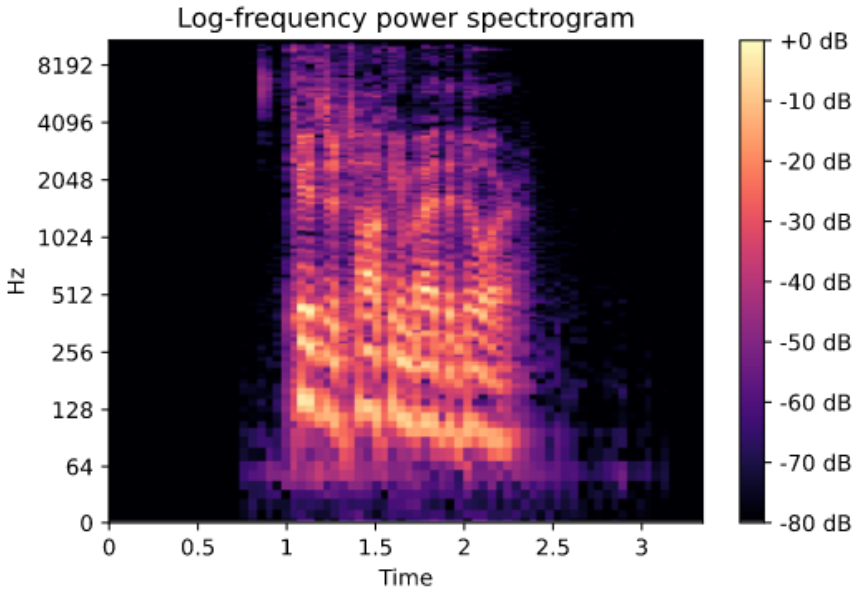

Figura 2.2: Espectrograma de una muestra de audio del dataset RAVDESS

2.5. Aprendizaje Profundo (*Deep Learning*)

2.5.1. Redes Neuronales

Son algoritmos de Aprendizaje de Máquina Supervisado. Están inspirados en la naturaleza de las conexiones de las neuronas en el cerebro.

El objetivo de una Red Neuronal es aproximar una función $y = f(x)$. Es decir, mapear una entrada x a una categoría y. Una red neuronal aproxima esta función de la forma $y = f(x, \theta)$ donde θ son los parámetros que tienen que ser aprendidos para mejorar la función de aproximación.

Las redes neuronales suelen estar formadas por varias funciones, de la forma: $f(x) = f^3(f^2(f^1(x)))$. Cada una de estas funciones son llamadas capas. Siendo f^1 la primera capa y así sucesivamente.

La cantidad de capas se conoce como profundidad (*depth*). De ahí el nombre de aprendizaje profundo o *deep learning*.

Cada capa puede ser expresada como: $y = f(x, \theta, w) = \phi(x, \theta)^T w$. Donde ϕ es una capa oculta. Y si los parámetros del modelo están representados por: w, b, como en una regresión lineal la función de la capa quedaría de la forma: $y = f(x, w, b) = x^T w + b$.

Este modelo solo resolvería problemas linealmente separables, pues resultaría en combinaciones lineales. Para evitar esto, se añade una función de activación no lineal g:

$$f(x, w, b) = g(x^T w + b)$$

A continuación la red neuronal debe buscar minimizar una función de coste o error, por ejemplo:

$$J(\theta) = \frac{1}{2} \sum_x (y - f(x, \theta))^2$$

Para lo cual se deben modificar los parámetros θ, buscando minimizar esta función $J(\theta)$. El proceso de entrenamiento está ligado al dataset de entrenamiento, que está conformado por pares (x, y), donde x es la entrada de la red neuronal y y es la salida esperada.

El proceso de optimizar los parámetros se consigue mediante el algoritmo de Gradiente Descendente. Donde se repite de forma iterativa, la actualización de parámetros:

$$\theta \leftarrow \theta - \alpha \frac{\partial J}{\partial \theta}$$

Donde α es la tasa de aprendizaje o *learning rate*. Este parámetro define que tan rápido se realizan los cambios de los parámetros θ.

2.5.2. Redes Neuronales Convolucionales (*Convolutional Neural Networks*)

Estas redes son una especialización de las Redes Neuronales. Se basan en la idea de tener kernels o filtros que puedan extraer las características más resaltantes de una imagen. Tiempo atrás estos filtros eran introducidos manualmente por un experto. Sin, embargo eran pocos los filtros conocidos y eran útiles para un número limitado de tareas. Actualmente estos filtros o kerneles pueden ser calculados automáticamente usando el algoritmo back propagation.

Las redes convolucionales están conformadas por varias Capas Convolucionales, las cuales aplican un conjunto de filtros de convolución sobre una entrada. A este resultado

se le aplica una función de activación. Por lo general, esta función de activación es RELU. Que está definida como:

$$f(x) = \text{máx}(0, x)$$

Usualmente estas capas suelen estar seguidas de Capas de Pooling, que serán descritas más adelante.

Filtros de Imágenes Lineales

También llamados kernels, F es un elemento $F \in \mathbb{R}^{k_w \times k_h \times d}$, donde k_h es la altura y k_w el ancho del filtro y d es el número de canales del input. El filtro F es operado con la imagen $I \in \mathbb{R}^{w \times h \times d}$ para producir una nueva imagen I'. El output I' tiene solo un canal. Cada pixel $I'(x, y)$ del output se obtiene calculando la multiplicación punto a punto de elemento del filtro con un elemento de la imagen original I.

$$I'(x,y) = \sum_{i_x=1-k_w/2}^{k_w/2} \sum_{i_x=1-k_h/2}^{k_h/2} \sum_{i_c=1}^{d} I(x+i_x, y+i_y, i_c) \cdot F(i_x, i_y, i_c)$$

Algunas tareas comunes que suelen ser realizadas con filtros lineales incluyen detección de bordes, detección de esquinas, smoothing, sharpening, filtros pasa bajo y pasa alto, etc.

Capas Convolucionales

Las capas convolucionales toman muchas características como inputs y producen n canales de características como outputs, donde n es el número de filtros en la capa convolucional. Los pesos del filtro de las convoluciones lineales son los parámetros, los cuales son adaptados a partir de la capa de entrenamiento. El número de filtros, así como el tamaño de el de los filtros son hiperparámetros. Esto también puede expresarse como $n@k_w \times k_h$.

Otro hiperparámetro de las capas de convolución es el stride $s \in \mathbb{N} \geq 1$; así mismo el Padding Zero es usado para asegurarse que el tamaño del canal de características no cambia.

Por lo que los hiperparámetros serían:

- El número de filtros
- Los tamaños de los filtros.
- La función de activación de la capa.
- El stride

Algunas elecciones típicas son: $n \in \{32, 64, 128\}, k_w = k_h = k \in \{1, 3, 5, 7, 11\}$ y la función de activación $ReLU$ y $s = 1$.

Capas de Pooling

La capa de pooling resume un área $p \times p$. Al igual que las capas convolucionales, las capas de pooling también pueden usar strides. Estas son conocidas también como capas de subsampling.

Algunos de las operaciones de pooling son:

- Max Pooling $maxa \in A$
- Average / Mean Pooling $\frac{1}{|A|} \sum_{a \in A} a$

El pooling se aplica por tres razones: Para obtener invariancia local traslacional, para obtener invariancia contra cambios pequeños y para reducir el tamaño de la imagen (usando stride $1/s^2$).

2.5.3. Redes Neuronales Recurrentes

Son una especialización que busca explotar la repetición de patrones en una serie temporal. Esta utiliza la salida de la red en el tiempo t, como entrada para la próxima iteración de tiempo $t + 1$.

Para cada instante de tiempo t, la función de activación $a^{<t>}$ y la salida $y^{<t>}$ están definidas por:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Donde $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ son coeficientes compartidos temporalmente y g_1, g_2 son funciones de activación

Estas redes suelen sufrir de un problema conocido como Desvanecimiento de la gradiente, ya que las derivadas disminuye exponencialmente con el tamaño de la serie temporal (t).

Por este motivo, se han propuesto alternativas que solucionan este problema como las redes Gated Recurrent Unit (GRU) y Long Short Term Memory (LSTM).

2.6. Métricas de Evaluación de Clasificadores

2.6.1. Accuracy

La métrica de accuracy está definida como:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Donde:

TP = True positive; FP = False positive; TN = True negative; FN = False negative

2.6.2. Precision

$$(Recall)\ o\ TPR = \frac{TP}{(TP + FN)}$$

$$TNR = \frac{TN}{(TN + FP)}$$

$$Precision = \frac{TP}{TP + FP}$$

2.6.3. Index of Balanced Accuracy (F-Score)

$$F_\beta score = \frac{(\beta^2 + 1) \times Precision \times Recall}{\beta^2 \times Precision \times Recall}$$

$$F_1 score = \frac{2 \times Precision \times Recall}{\times Precision \times Recall}$$

Un valor β usando comúnmente es 1, lo que se conoce como F_1 Score:

2.7. Test Estadísticos

Un contraste o test de hipótesis es una técnica de Inferencia Estadística que permite comprobar si la información que proporciona una muestra observada concuerda (o no) con la hipótesis estadística formulada sobre el modelo de probabilidad en estudio y, por tanto, se puede aceptar (o no) la hipótesis formulada.

Una hipótesis estadística es cualquier conjetura sobre una o varias características de interés de un modelo de probabilidad.

Una hipótesis estadística puede ser:

- **No Paramétrica:** Es una afirmación sobre los valores de los parámetros poblacionales desconocidos. Las hipótesis paramétricas se clasifican en:
  - **Simple:** Si la hipótesis asigna valores únicos a los parámetros
  - **Compuesta:** Si la hipótesis asigna un rango de valores a los parámetros poblacionales desconocidos
- **Paramétrica:** Es una afirmación sobre alguna característica estadística de la población en estudio. Por ejemplo, cuando las observaciones son independientes, la distribución de la variable en estudio es normal o la distribución es simétrica.

2.7.1. Tipos de Error

En un contraste de hipótesis, al realizar un contraste se puede cometer uno de los dos errores siguientes:

- Error tipo I, se rechaza la hipótesis nula H_0 cuando es cierta.
- Error tipo II, se acepta la hipótesis nula H_0 cuando es falsa.

2.7.2. Paramétricos: T-Test

En esta tesis estudiaremos el caso para comparar dos muestras: Esta prueba se utiliza solo si los dos tamaños muestrales (esto es, el número, n, de participantes en cada grupo) son iguales; se puede asumir que las dos distribuciones poseen la misma varianza.

El estadístico t a probar si las medias son diferentes se puede calcular como sigue:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{X_1X_2} \cdot \sqrt{\frac{2}{n}}}$$

Donde

$$S_{X_1X_2} = \sqrt{\frac{1}{2}(S_{X_1}^2 + S_{X_2}^2)}$$

es la desviación estándar combinada, 1 = grupo uno, 2 = grupo 2. El denominador de t es el error estándar de la diferencia entre las dos medias.

Por prueba de significancia, los grados de libertad de esta prueba se obtienen como $2n - 2$ donde n es el número de participantes en cada grupo.

2.7.3. No Paramétrico: Wilcoxon Signed Rank Test

Para calcular el *Wilcoxon signed rank statistic* T^+, se forman los valores absolutos $|Z_1|, ..., |Z_n|$ de las diferencias y el orden de menor a mayor. Se denota R_i al *rank* de $|Z_i|, i = 1, ..., n$, en esta orden. Se definen las variables indicadoras $\phi_i, i = 1, ..., n$, donde

$$\phi_i = \begin{cases} 1 & si \quad Z_i > 0 \\ 0 & si \quad Z - i < 0 \end{cases}$$

y se obtienen los n productos $R_1...R_n$. Donde el producto R_i ϕ_i es conocido como el *positive signed rank* de Z_i . Se toma el valor de cero si Z_i es negativo e igual al *rank* de $|Z_i|$ cuando Z_i es positivo. El *Wilcoxon signed rank statistic* T^+ es la suma de todos los ranks positivos.

$$T^+ = \sum_{i=1}^{n} R_i \phi_i$$

Se acepta o rechaza la hipótesis nula dependiendo de los valores críticos, según la elección de α y el tamaño n.

Capítulo 3

Estado del Arte

3.1. Reconocimiento de Emociones a partir del habla

Por largo tiempo el problema de Reconocimiento de emociones a partir del habla fue abordado utilizando características extraídas de segmentos de audio de forma manual. Entre los años 90 y 2000, se utilizaron como por ejemplo pitch, speaking rate, average length between voiced regions, frecuencias fundamentales, frecuencias espectrales, estadísticas de ”falling slopes”, etc. Y entre los métodos de clasificación populares se usaban k Nearest Neighboors (KNN) y redes neuronales de pocas neuronas (menos de 20) [Petrushin, 1999, Ververidis et al., 2004].

Años más tarde con la aparición de nuevos métodos de clasificación como Support Vector Machine (SVM), Hidden Markov Model (HMM) y Gaussian Mixture Model (GMM). Se consiguió utilizar mayor número características, por lo que se popularizó el uso de Mel Frequency Cepstral Coefficients (MFCC), que se consigue tras la obtención del espectrograma generado por FT [Nwe et al., 2003, Agarwal et al., 2018, Schuller et al., 2004]

A la actualidad tras la consolidación del Deep Learning y el aumento de Datasets, destacan, los enfoques basados en Deep Learning. Algunas arquitecturas comunes son las Redes Neuronales Convolucionales y las Redes Neuronales Recurrentes. Las Redes Convolucionales son usadas porque logran aprovechar los espectrogramas y los tratan como imágenes. Mientras que las Redes Recurrentes son adecuadas, pues modelan series

de tiempo, como las ondas de audio o secuencias de patrones.

Entre los trabajos más resaltantes tenemos: A Convolutional Neural Network (CNN) - Assisted Enhanced Audio Signal Processing for Speech Emotion Recognition [Kwon et al., 2020] que utiliza un modelo de Redes Convolucionales, sin embargo no se utilizan capas de pooling, lo que el autor denomina Deep Stride Convolution. Adicionalmente, realiza un preprocesamiento de la data, eliminando el ruido y silencios. Este método reduce considerablemente la cantidad de capas de la red por lo tanto recursos necesarios; pero a su vez mejora la precisión del modelo.

En "An Ensemble Model for Multi-Level Speech Emotion Recognition "[Zheng et al., 2020] los autores evaluaron distintas arquitecturas de deep learning: Double CNN, GRU + Attention, Convolutional Recurrent Neural Network, y adicionalmente usando machine learning tradicional con descriptores de audio de alto y bajo nivel. Combinando los mejores modelos se crea un nuevo modelo que usa los modelos previos como juez, teniendo mayor peso los que obtuvieron mayor accuracy.

"End-to-End Emotion Recognition From Speech With Deep Frame Embeddings And Neutral Speech Handling"[Sterling and Kazimirova, 2019] utilizan la arquitectura ResNet-18, que es un modelo ampliamente usado en Visión Computacional para clasificación de imágenes. En este caso toma como input el espectrograma del audio por fragmentos de 0.5 segundos. De esta forma se forman varias posibles respuestas para un audio. Las salidas se ponderan y a partir de estas se obtiene un resultado final.

"Deep spectrum feature representations for speech emotion recognition"[Zhao et al., 2018] explora los modelos recurrentes LSTM y Attention concatenándolos con CNN. "Convolutional Neural Networks in Speech Emotion Recognition – Time-Domain and Spectrogram-Based Approach"[Stasiak et al., 2019]. Realizó experimentos utilizando CNN 2D (Espectrograma) y 1D (Dominio de Tiempo) una variante de este trabajo, es "Speech emotion recognition using deep 1D and 2D CNN LSTM networks"[Zhao et al., 2019] adiciona LSTMs.

La extracción de caraterísticas a partir de los espectrogramas es común a todas las técnicas, sin embargo la forma de representar los espectrogramas puede ser variable por las escalas usadas para representarlo como "mel" o "log". "Acoustic Characteristics of Emotional Speech Using Spectrogram Image Classification"[Stola et al., 2018] rea-

liza un estudio sobre las mejores formas de representar espectrograma, llegando a la conclusión que la escala "mel" usando los 3 colores RGB obtiene los mejores resultados.

3.2. Evaluación y Comparación de modelos de Deep Learning

La tarea de comparar las propuestas del estado del arte es complicada debido a la falta de estándares como métricas o datasets. Por lo que las revisiones de literatura se limitan a realizar surveys descriptivos sin proveer información que permita identificar los métodos más precisos.

Por ejemplo, en [Akçay and Oğuz, 2020, Swain et al., 2018, Schuller, 2018] se realizaron revisiones amplias de los métodos y bases de datos presentes en la literatura. Pero si hablamos de comparaciones específicas, "Evaluating deep learning architectures for Speech Emotion Recognition" [Fayek et al., 2017], presentó una comparación de algunos modelos de deep learning, siendo esta forma (Replicar por uno mismo varias modelos y evaluarlos) la más común a la que los autores recurren para comparar una propuesta contra el estado del arte.

Este problema se extiende a otras áreas donde se ha observado problemas de rigurosidad, replicabilidad y falta de estándares para realizar comparaciones. En el área de Sistemas de Recomendación, [Sun et al., 2020] se cuestiona si están evaluando sus sistemas de forma rigurosa y propone una herramienta para lograrlo. De la misma forma en el área de *Graph Neural Networks* [Errica et al., 2019], propone dos fases para conseguir comparaciones justas y reproducibilidad: Selección de modelos y Evaluación de modelos. Este trabajo también explora los fallos más comunes para lograr reproducibilidad como ausencia de información sobre el preprocesamiento y particiones de los datos.

Sin embargo, este tipo de artículos nos demuestran lo lejos que se está de tener comparaciones efectivas. Y los mayores esfuerzos están sumados en la realización de challenges. Estas son competencias, donde se distribuye un dataset etiquetado. Los competidores entrenan sus modelos y finalmente los modelos deben evaluar un dataset de prueba. Según los resultados obtenidos en alguna métrica se posicionan en una

tabla de posiciones. De esta forma se debería garantizar que las comparaciones se están realizando en las mismas condiciones con respecto a la base de datos.

En 2011 se presentó el primer Audio Visual Emotion Challenge (AVEC), que incluyó un sub challenge de solo audio. En este challenge se utilizó la base de datos SEMAINE, y se proporcionó 3 particiones del dataset, evaluándose la precisión en clasificación [Schuller et al., 2011]. Esta competencia se llevó a cabo hasta el año 2015 [Ringeval et al., 2015], cambiando posteriormente los challenges por detección de emociones específicas como afecto o emoción. Aunque AVEC está enfocado en el estudio de detección de emociones multimodal sentó las bases para evaluar los modelos proponiendo datasets, métricas e incluso un baseline.

Emotion Recognition in the Wild (EmotiW) similarmente propone cada año un challenge de detección de emociones audiovisual [Dhall et al., 2018]. En este challenge se propone un dataset dividido en 3 conjuntos de datos: Train, Validation y Test. Aquí se utilizó las siguientes emociones: *Angry, Disgust, Fear, Happy, Neutral, Sad y Surprise* (Enojo, digusto, miedo, neutral, tristeza y sorpresa). Aunque EmotiW está enfocado en video, propuso un baseline basado en Deep Learning, lo cual es innovador en comparación a AVEC que usaban características (features) manuales para entrenar sus modelos.

Sin embargo, cuando se realizan los challenges, no tenemos garantías de que estos resultados no hayan sido fruto de la casualidad, o de un intento de ataque por fuerza bruta, como lo demuestra este artículo. [Whitehill, 2018].

Es por eso que se requieren estrategias más robustas, como la validación cruzada por K-Fold. Que permite hacer varias pruebas sobre el mismo dataset. Esto permite detectar alteraciones de los resultados por causa de la partición train/test de los datos.

Para comprobar si existe una mejora significativa se puede realizar un test de significancia estadística. Que nos permitirá saber si nuestro modelo está por fuera de los márgenes de la casualidad y que si es estadísticamente diferente. Estos test estadísticos pueden ser aplicados sobre los resultados de la validación cruzada. Sin embargo, esto podría llevarnos a conclusiones erróneas. Pues se ha demostrado experimentalmente, que tras varias repeticiones, se puede llegar a distintas conclusiones: Que exista evidencia de significancia estadística y a su vez lo contrario. [Stapor et al., 2021]

Capítulo 4

Propuesta

La presente tesis propone un framework de evaluación y comparación de modelos de Deep Learning. Este framework consiste de una serie de definiciones, metodologías a ser aplicadas y una serie de recomendaciones, que faciliten la creación y comparación de modelos de Deep Learning.

4.1. Datasets

Por lo general el entrenamiento de modelos de Deep Learning parte del análisis de un dataset.

Sin embargo, la falta de estandarización en la lectura de los datos, provoca retrasos, pues para cada formato de dataset se debe escribir un script de lectura diferente.

Librerías como Tensorflow o Pytorch implementan clases para estandarizar la carga de datos. Estas clases permiten incluir datasets como objetos comunes, abstrayendo los detalles de lectura o el formato del dataset.

El siguiente segmento de código muestra como se implementa una DataLoader en Pytorch. Este contiene dos métodos básicos: **len** y **getitem**. Esta estructura simple permite programar internamente diferentes funcionalidades, como paralelización de la lectura de datos, o mantener el dataset en memoria. Los cuales quedan transparentes para el investigador.

```
class CustomDataset(Dataset):
    def __init__(self, args, transform=None) :
        self.args = args
        self.transform = transform

    def __len__(self):
        return len(self.args)

    def __getitem__(self, idx):
        item = self.args[idx]
        if self.transform:
            item = self.transform(item)
        return item
```

Sin embargo, este enfoque previene su uso fuera de otros frameworks o librerías. Y a la actualidad no existe una librería de Carga de Datasets, independiente de algún otro framework, con su uso lo suficientemente popular para usarse como estándar.

Una alternativa de estandarización es utilizar un archivo **dataset.csv**, que mantenga la lista de todos los archivos, sus rutas y en la siguiente columna las etiquetas o características:

```
file_name, feature1, feature2
/folder/audio1.wav, 1, 2
/folder/audio2.wav, 1, 2
/folder/audio3.wav, 1, 2
...
```

Un archivo README.md debe incluirse conteniendo la explicación de cada atributo del archivo.

Un problema que este formato de almacenamiento tiene es que no soporta anotaciones más complejas, por ejemplo, anotaciones por intervalos de tiempo. O la inclusión de más características como transcripciones de texto. Alternativas se han propuesto como por ejemplo: Emotion Markup Language (EmotionML) [Burkhardt and Schröder, 2008]. Sin embargo, no se ha tenido éxito en su adopción y masificación.

4.2. Particionamiento de Datasets

Es usual en deep learning, dividir el dataset en varios segmentos. Uno para entrenamiento, otro para validación y otro para test. Al realizar estas particiones se puede estar condicionando la eficacia de un modelo. Por lo que se recomienda utilizar estrategias que permitan alejarse de esa dependencia, pero a su vez mantenga la reproducibilidad de los experimentos.

Se propone el uso de Validación Cruzada, que busca solucionar el problema de que los resultados varíen dependiendo de como se realizó la partición. La validación cruzada basada en K-fold, divide el conjunto de datos en K particiones. Uno de estos se usa como conjunto de prueba y el resto $(k-1)$ se utiliza como conjunto de entrenamiento. El proceso se repite con cada uno de los subconjuntos de datos de prueba. Finalmente los resultados son promediados.

4.3. Métricas de Evaluación de Modelos

Cada dataset debe incluir métricas de evaluación adecuadas. Por ejemplo, la métrica accuracy no suele ser adecuada por no representar los fallos del modelo. De forma similar métricas como Precisión y Recall por si solas pueden no representan por completo los resultados. Por lo que se propone el uso de la métrica F1-score, detallada en el marco teórico, como estándar de evaluación.

Sin embargo, cuando los datasets son multiclase. Suelen existir clases con baja cantidad de muestras. Por ejemplo, en el caso de Speech Emotion Recognition (SER) la clasificación es multiclase, debido a que un audio se puede clasificar con varias emociones y por lo general desbalanceada, debido a la naturaleza del problema donde por lo general la voz humana es neutra, y algunas emociones se pueden capturar con más frecuencia que otras, como neutral versus tristeza.

Una forma de equilibrar la baja presencia de esta clase es utilizar una métrica balanceada. Se propone la métrica Weighted F1-Score en datasets multiclase.

4.4. Comparación Estadística

Teóricamente, el teorema "No free lunch"demuestra que para cada dos modelos clasificadores, existen varios problemas de clasificación en los que un modelo superará al otro. Por lo que más allá de enfrentar modelos. Debemos reconocer las condiciones en las que uno es mejor que otro. Por ejemplo: Evaluar, modelos con respecto a una métrica de evaluación para un dataset.

Adicionalmente, al comparar los modelos usando métricas debemos asegurarnos que sean estadísticamente diferentes y verificar si se rechaza la hipótesis nula. Es decir que los resultados obtenidos, no hayan sido obtenidos por causa de la casualidad. Para realizar estas comprobaciones, existen diversos métodos estadísticos, como la T-statistics.

Sin embargo, se ha demostrado experimentalmente [Stapor et al., 2021] que incluso aplicando técnicas de K-Fold ocurren casos, en los que en una muestra (Comparación de accuracy obtenido por 2 modelos) se rechaza la hipótesis nula (Uno es significativamente diferente a otro) y en otra muestra, se acepta (No son los suficientemente diferentes). Esto sucede a que a la naturaleza de inicialización aleatoria de los modelos y particionamiento de los datasets, también aleatorio.

Para evitar este tipo de falsos rechazos de hipótesis nula, proponemos la utilización de fuerza bruta para poder genera suficientes muestras y observar el porcentaje de aceptación y rechazo de la hipótesis nula en las muestras de comparación de modelos.

4.4.1. Recolección de muestras

Se recolectará muestras por medio de repetición sucesiva del entrenamiento mediante la técnica de K-Fold para particionar el dataset. Pues previamente [Stapor et al., 2021] demostró que, tras analizar más muestras en modelos de Machine Learning simples y datasets menos complejos, es posible observar la tendencia de comportamientos de los modelos. Es decir que probabilidad existe de que un modelo sea estadísticamente significativo a otro.

En este enfoque, tanto el Modelo A (M_A) con el Modelo B (M_B), serán entrenados y validados con la misma partición de datos.

Cada validación cruzada k-fold creada apartir de un dataset D recolecta un conjunto

C de k pares. De la siguiente forma.

$$C(D, M_A, M_B) = \{(S_A^j, S_B^j), (S_A^{j+1}, S_B^{j+1}), ..., (S_A^k, S_B^k))\}$$

Donde S_A^j se refiere al Score (Métrica utilizada como por ejemplo accuracy) obtenido por el modelo M_A sobre la partición de datos D_j generado por el método k-fold.

Se repetirán N iteraciones de este procedimiento. Obteniendo N conjuntos C_i.

$$G = \{C_i, C_{i+1}, ..., C_N\}$$

El procedimiento se detalla en el Algoritmo 1.

Algorithm 1: Recolección de muestras

Input: M_A, M_B, D, k

Output: G

G = { }

for *i=0; i < N; i++* **do**

  C_i = { }

  foreach *train, test, j = KFold(D)* **do**

    M_A.fit(train)

    M_B.fit(train)

    S_A^j =M_A.predict(test) ;

    S_B^j =M_B.predict(test) ;

    $C_i = C_i \cup \{(S_A^j, S_B^j)\}$

  end

  $G = G \cup \{C_i\}$

end

4.4.2. Prueba de Significancia Estadística

La evaluación estadística se realiza utilizando los resultados recolectados. De la siguiente forma:

Para cada $C_i \in G$, se obtendrán los valores p-value y t-value mediante el test estadístico de Wilcoxon, ya que no se puede garantizar que los datos cumplan una distribución normal. El valor T indica cuanta diferencia existe entre los resultados

	Model A	**Model B**	**T**	**p**	**Distribution**
Avg all	Avg Score STDev	Avg Score STDev	Avg T Values	Avg p Values	100 % = %(=) + %(-) + %(+)
Models (=) where p>0.05	Avg Score STDev	Avg Score STDev	Avg T Values	Avg p Values	%(=)
- where p<0.05 where T<0	Avg Score STDev	Avg Score STDev	Min T Values	p Values of Min T Values	%(-)
+ where p<0.05 where T>0	Avg Score STDev	Avg Score STDev	Max T Values	p Values of Max T Values	%(+)

Cuadro 4.1: Tabla de Resultados

obtenidos por los dos modelos. A mayor t menores valores p se obtendrán. Cuando $p < 0{,}05$, podemos asumir que existe diferencia estadísticamente significativa.

Se deben realizar el procedimiento para cada $C_i \in G$ 2

$$p_i, t_i = Wilcoxon(C_i)$$

Algorithm 2: Algoritmo de comparación estadística

Input: G

Output: P, T

$P = \{\}$ $T = \{\}$

for *i=0; i<N; i++* **do**

  $p_i, t_i = Wilcoxon(C_i)$

  $P = P \cup \{p_i\}$

  $T = T \cup \{t_i\}$

end

Se considerará que la Hipótesis Nula H_0 ha sido aceptada si: $p < 0{,}05$. Una tabla como la siguiente puede detallar, los resultados: Tabla 4.1

La elección de modelos será realizada basándose en el porcentaje de Aceptación o Rechazo de hipótesis nula, observando una tabla de distribución similar a: Tabla 4.1

4.5. Modelos de Deep Learning

Se requiere que los modelos puedan ser reutilizables, para lo cual se recomienda que el código fuente del modelo así como los algoritmos de entrenamiento sean compartidos

con los involucrados, sin embargo, esto requiere conocimiento previo de las librerías y/o técnicas de programación utilizadas.

Varios intentos de estandarizar modelos de redes neuronales se han propuesto. Partiendo desde la implementación de clases abstractas como keras. Model de Tensorflow , hasta esfuerzos conjuntos como ONNX que permite la interoperabilidad entre muchas librerías del ecosistema de Deep Learning.

Los pesos de los modelos suelen ser dejados de lado, sin embargo, son una parte importante para la reproducibilidad de los experimentos y la revisión por pares. Sin los pesos, no se puede comprobar los accuracies obtenidos, ni realizar futuras comparaciones.

Capítulo 5

Experimentos y Resultados

5.1. Experimento: Dataset RAVDESS

Para esta evaluación estamos tomando el dataset RAVDESS [Livingstone and Russo, 2018]. Debido a la baja complejidad de las frases y corta duración de los audios. En total tenemos 1441 audios de hombres y mujeres grabados a 48 kHz. Haciendo un total de 1.27 horas. Las emociones están clasificadas en: Calma, feliz, triste, enojo, miedo, sorpresa y disgusto.

5.1.1. Estandarización de dataset

Con la finalidad de estandarizar el uso del dataset y facilitar futuros experimentos se convirtieron los audios, que estaban en el siguiente formato:

$$XX - YY - ZZ - AA - BB - CC - DD.wav$$

Donde cada par de dígitos representaba distintas características:

- Modalidad (01 = AV completo, 02 = solo video, 03 = solo audio).
- Canal vocal (01 = habla, 02 = canción).
- Emoción (01 = neutral, 02 = calma, 03 = feliz, 04 = triste, 05 = enojado, 06 = temeroso, 07 = disgusto, 08 = sorprendido).

- Intensidad emocional (01 = normal, 02 = fuerte). NOTA: No hay una gran intensidad para la emoción ”neutral”.
- Declaración (01 = ”Los niños están hablando junto a la puerta”, 02 = ”Los perros están sentados junto a la puerta”).
- Repetición (01 = 1ª repetición, 02 = 2ª repetición).
- Actor (01 a 24. Los actores impares son hombres, los actores pares son mujeres).

A un formato .csv

```
file_name, emotion
/folder/1.wav, 1
/folder/2.wav, 3
/folder/3.wav, 2
...
```

De esta forma se elimina la complejidad extra de cargar los archivos de audio.

5.1.2. Preprocesamiento

Todos los audios fueron convertidos a espectrogramas (Fig. 5.1) y posteriormente a su representación MFCC. Adicionalmente, como cada audio tenía una longitud diferente, se preprocesó utilizando una ventana deslizante para extraer frames de igual longitud de cada audio. Se utilizaron los siguientes parámetros: $WindowSize = 55$ y $hop\_length = 10$.

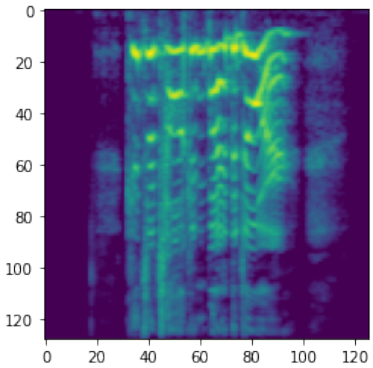

Figura 5.1: Espectrograma generado a partir de audio

5.1.3. Entrenamiento

Se realizó el entrenamiento de 3 Modelos de Redes Convolucionales (A, B, C) que reciben como input frames de audio 2D de tamaño $(20, 55)$.

Los modelos implementados se detallan a continuación:

```
A = Sequential(
        [
            Input(shape=(20, window)),
            Reshape((20, window, 1)),
            Conv2D(64, 3, activation="relu"),
            MaxPool2D((1, 2)),
            Conv2D(64, 3, activation="relu"),
            MaxPool2D((1, 2)),
            Conv2D(64, (1, 3), activation="relu"),
            MaxPool2D((1, 2)),
            Conv2D(64, (1, 3), activation="relu"),
            MaxPool2D((2, 1)),
            Conv2D(128, 3, activation="relu"),
            Dropout(0.3),
            Flatten(),
            Dense(1024, activation="relu"),
            Dropout(0.4),
            Dense(8, activation="softmax"),
        ]
```

El modelo A cuenta con 5 Capas Convolucionales y 2 Capas Full Connected y en total cuenta con **931,784** parámetros.

```
B = Sequential(
        [
            Input(shape=(20, window)),
            Reshape((20, window, 1)),
            Conv2D(64, 3, activation="relu"),
            MaxPool2D((1, 2)),
            Conv2D(64, 3, activation="relu"),
            MaxPool2D((1, 2)),
            Conv2D(128, 3, activation="relu"),
            MaxPool2D((1, 2)),
            MaxPool2D((2, 1)),
```

```
            Conv2D(128, 4, activation="relu"),
            Dropout(0.3),
            Flatten(),
            Dense(1024, activation="relu"),
            Dropout(0.4),
            Dense(8, activation="softmax"),
        ]
```

El modelo A cuenta con 4 Capas Convolucionales y 2 Capas Full Connected y en total cuenta con **1,431,496** parámetros.

```
C = Sequential(
        [
            Input(shape=(20, window)),
            Reshape((20, window, 1)),
            Conv2D(64, 3, activation="relu"),
            MaxPool2D((1, 2)),
            Conv2D(64, (1, 3), activation="relu"),
            MaxPool2D((1, 2)),
            Conv2D(64, (1, 3), activation="relu"),
            MaxPool2D((2, 1)),
            Conv2D(128, 3, activation="relu"),
            Dropout(0.3),
            Flatten(),
            Dense(1024, activation="relu"),
            Dropout(0.4),
            Dense(1024, activation="relu"),
            Dropout(0.4),
            Dense(8, activation="softmax"),
        ]
    ),
```

El modelo A cuenta con 4 Capas Convolucionales y 3 Capas Full Connected y en total cuenta con **8,498,05** parámetros.

El dataset fue particionado con una estrategia K-fold ($k = 5$). Proporcionada por la librería ScikitLearn.

El entrenamiento se realizó utilizando la librería tensorflow y el criterio de parada siguió una estrategia EarlyStopping para evitar overfitting. Con parámetro de paciencia

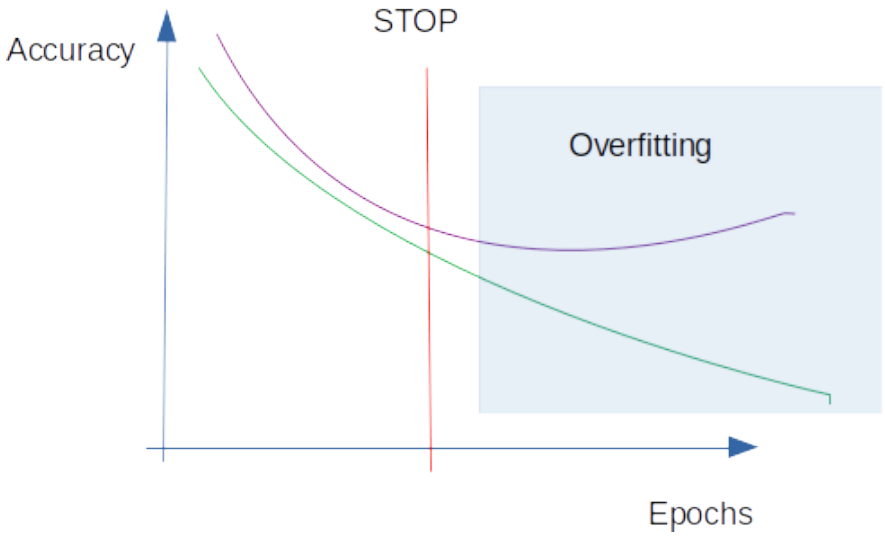

Figura 5.2: Early Stop Concept

a *epochs* = 15 y se monitoreó la métrica "val_accuracy". Finalmente, se restauran los mejores pesos obtenidos. (Fig. 5.2)

El entorno de entrenamiento fue una Tarjeta Nvidia RTX 2060 (6GB Vram), en un Sistema Operativo Linux con 16 GB de ram.

5.1.4. Test estadístico simple

Dada la hipótesis nula:

H_0 : No existe diferencia estadísticamente significante entre los resultados de los modelos X,Y .

Realizamos un test de Wilcoxon (signed-rank) entre cada par de modelos (A,B) y (B,C) sobre los resultados del entrenamiento con K-fold, sobre la métrica F-1 Score. En este se obtuvieron los siguientes resultados: Entro los modelos A y B se obtuvo un $p\_value = 0{,}89$ y $T = 7{,}0$. Mientras que entre los modelos B y C, se obtuvo $p\_value = 0{,}04$ y $T = 0{,}0$.

De aquí podríamos deducir que entre el modelo A y B no existe diferencia significativa, y que entre el modelo B y C, si existe pues el $p\_value < 0{,}05$. Sin embargo, si repetimos el experimento, no se obtienen los mismos resultados. En una próxima iteración, se obtuvo: (A-B) $p = 0{,}043$ y $T = 0{,}0$, mientras que (B-C) $p = 0{,}22$ y $T = 3{,}0$. Lo que es una contradicción al resultado anterior. Por este motivo es necesario repetir

el experimento para tener conclusiones más acertadas.

5.1.5. Test estadístico con varias muestras

Se repitió el experimento de entrenar y probar el modelo 39 veces. Posteriormente se realizaron test de Wilcoxon para cada conjunto de datos. La siguiente gráfica muestra los resultados obtenidos para cada par de modelos comparados.

Observamos que existen gran cantidad de valores cercanos a 0.05 en la comparación del modelo A vs el modelo B (Fig. 5.3). Mientras que en la Figura 5.4 la gráfica está más dispersa.

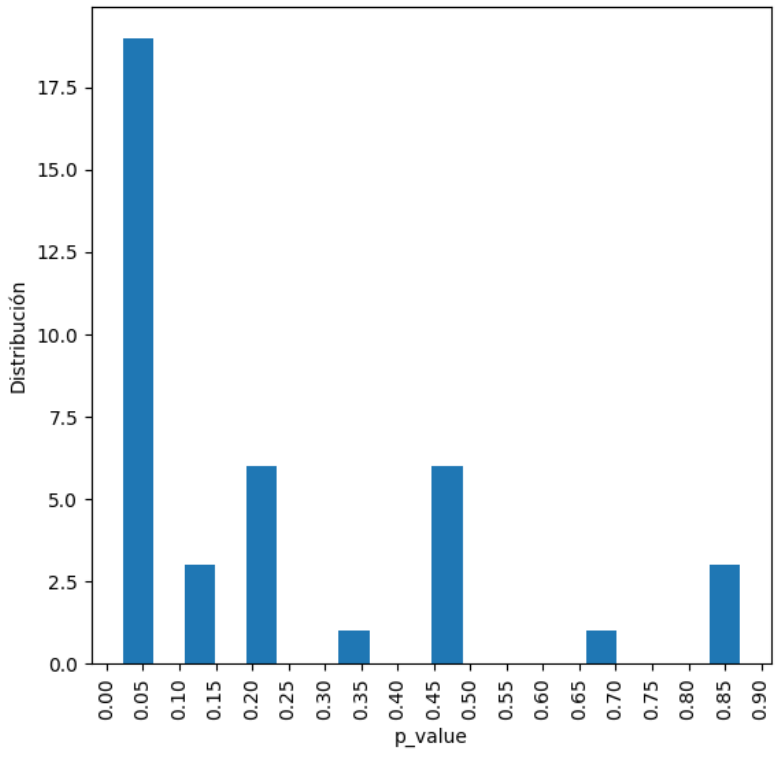

Figura 5.3: p_values (modelo A vs modelo B) Ravdess

Las siguientes cuadros muestran de forma detallada los resultados, obtenidos. En la comparación del modelo A contra el modelo B. Podemos observar que existe diferencia significativa en 31 % de las iteraciones a diferencia de la comparación B vs C, que tiene solo un 20 %. En aquellos casos, el modelo A fue superior al B. Y el modelo B fue

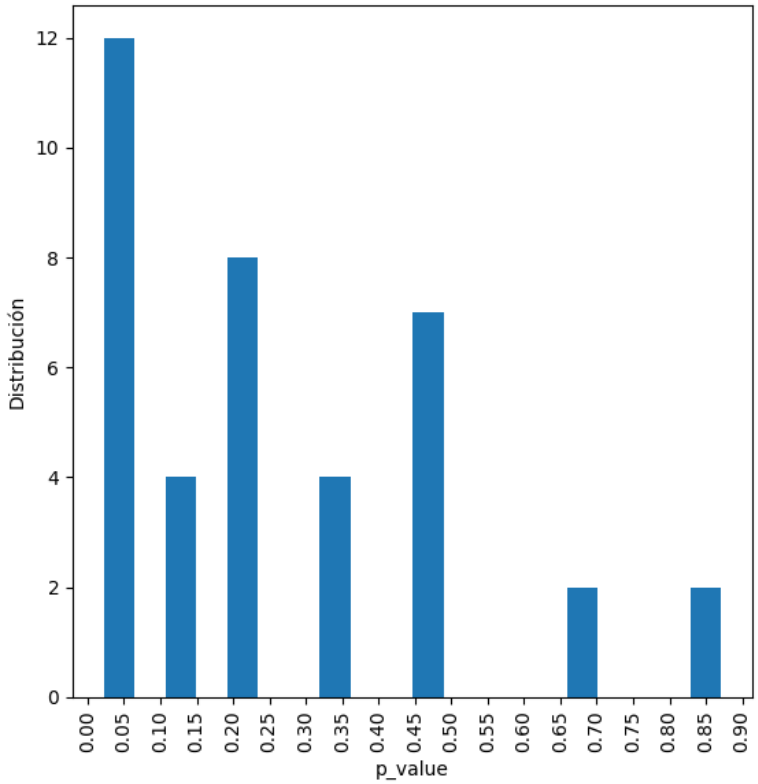

Figura 5.4: p_values (modelo B vs modelo C) Ravdess

superior al modelo C.

Con estos resultados, queda a criterio de los investigadores definir un umbral de porcentaje para declarar a un modelo superior a otro. Por ejemplo en nuestro caso podría ser 30 %. Por lo que se afirmaría que:

El modelo A es superior al modelo B, dado que existe 30 % de ocasiones en las que existe diferencia significativa.

No se puede afirmar lo mismo para la comparación del modelo B y C pues solo un 20 % de los entrenamientos mostraron diferencia significativa.

Adicionalmente, se debe destacar que este experimento logra mostrar la tendencia de porcentajes que tendrían nuestros modelos si se repitieran más experimentos.

	Model A	**Model B**	**T**	**p**	**Distribución**
Avg all	0.60 (0.019)	0.59 (0.030)	2.36	0.24	100 % = Perc(=) + Perc(-) + Perc(+)
(=) where p>0.05	0.60	0.59	3.4	0.33	69 %
-/+ where p<0.05	0.60	0.58	0	0.043	31 %

Cuadro 5.1: Modelo A vs Modelo B

	Model B	**Model C**	**T**	**p**	**Distribución**
Avg all	0.59 (0.030)	0.57 (0.019)	2.89	0.28	100 % = Perc(=) + Perc(-) + Perc(+)
(=) where p>0.05	0.58	0.57	3.6	0.34	80 %
-/+ where p<0.05	0.59	0.57	0	0.043	20 %

Cuadro 5.2: Modelo B vs Modelo C

5.2. Experimento: Dataset IEMOCAP

El dataset IEMOCAP (Interactive Emotional Dyadic Motion Capture) [Busso et al., 2008] es una base de datos de emociones categóricas en audio y video con cerca de 10000 muestras. Por lo que presenta una mayor complejidad.

5.2.1. Estandarización del dataset

Se siguió la estandarización utilizada en el primer experimento en formato de archivo CSV. Obteniendo un resultado como se muestra a continuación:

```
file,emotion
0.wav,Neutral state
1.wav,Frustration
2.wav,Frustration
3.wav,Frustration
4.wav,Happiness
5.wav,Neutral state
6.wav,Excited
7.wav,Other
8.wav,Frustration
9.wav,Excited
```

```
10.wav,Neutral state
11.wav,Neutral state
12.wav,Anger
13.wav,Sadness
14.wav,Sadness
15.wav,Sadness
...
```

5.2.2. Entrenamiento

Se entrenaron los tres modelos de Redes Neuronal Convolucionales, mencionados en el experimento anterior con el dataset IEMOCAP.

5.2.3. Test Estadístico

Tras recolectar datos de la métrica de Accuracy durante 30 iteraciones y k-fold 5 por un periodo de tiempo de cerca 41 horas. Se obtuvieron los siguientes resultados:

Podemos observar en las imágenes 5.5 , 5.6 y 5.7 que todos los valores p obtenidos estan por encima de 0.05 .Y dado que definimos un umbral de $p\_value < 0{,}05$ para rechazar la hipótesis nula, no podemos afirmar que exista alguna diferencia significativa entre los tres modelos para este Dataset.

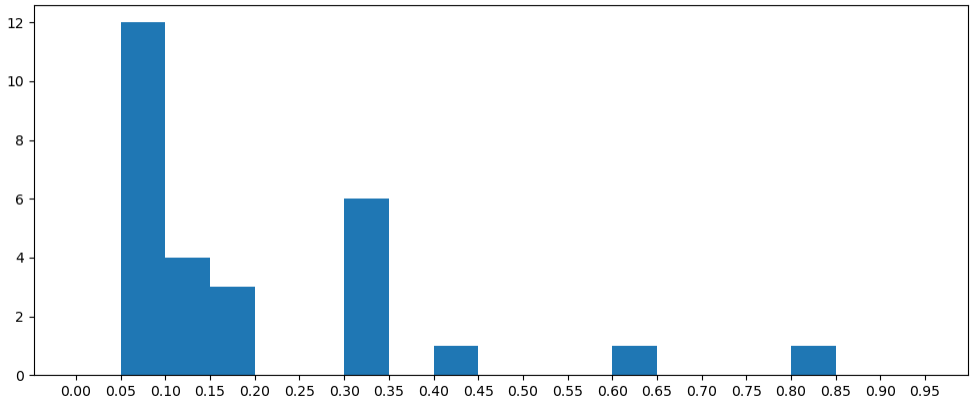

Figura 5.5: p_values (modelo A vs modelo B) IEMOCAP

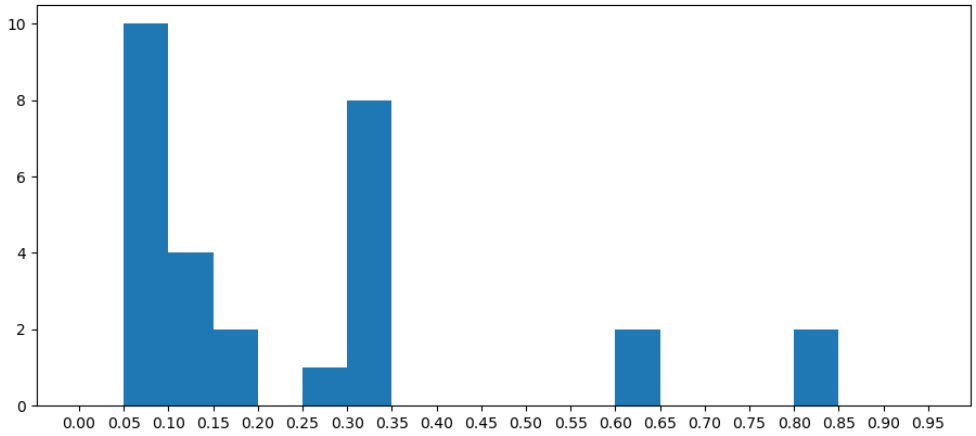

Figura 5.6: p_values (modelo A vs modelo C) IEMOCAP

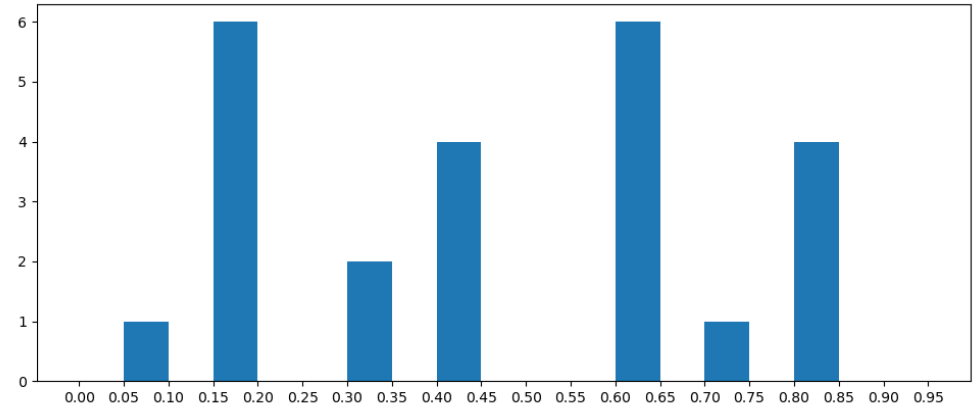

Figura 5.7: p_values (modelo B vs modelo C) IEMOCAP

5.3. Discusión

Los resultados obtenidos en el experimento RAVDESS demuestran que incluso realizando una estrategia de K-fold y una prueba estadística. Los resultados podrían cambiar e inclusive contradecirse. Pues con el mismo modelo se podría aceptar y rechazar

la hipótesis H_0 si se repitiera el experimento con el mismo dataset, pero diferente orden en el K-fold. Por lo que se podría llegar a conclusiones erróneas o sin la suficiente evidencia al afirmar que un nuevo modelo es superior a otro. Siendo esa ocasión un producto de la casualidad o azar.

Repetir el experimento la suficiente cantidad de veces puede mostrar una tendencia clara sobre la superioridad en términos de precisión o la métrica deseada. Lo que nos permitiría aseverar de forma más correcta que: "El modelo X es significativamente superior al modelo Y, en el n % de las veces". O en caso contrario como en el experimento IEMOCAP que no existe diferencia significativa entre los modelos.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

El área de Speech Emotion Recognition, ha sido ampliamente estudiada, por lo que se pueden encontrar varios surveys y trabajos de recopilación sobre el estado del arte. Estos muestran la amplia variedad de métodos y clasificadores que se han propuesto para la tarea. Sin embargo, la gran mayoría de trabajos, no cuenta con la suficiente rigurosidad para afirmar que su propuesta es superior a otra. Esto se debe a la falta de estandarización de datasets, la forma de particionar el dataset, las métricas de evaluación, etc. Además, son pocos los trabajos que implementan pruebas de significancia estadística.

El presente trabajo busca lidiar con estos problemas, presentando un framework de evaluación de modelos de reconocimiento de emociones. El cual permite realizar comparaciones objetivas, para lo cual propone la estandarización de datasets, métricas de evaluación y una prueba de significancia estadística. El framework propuesto fue implementado y puesto a prueba con tres arquitecturas basadas en el estado del arte del área de Speech Emotion Recognition.

Los resultados demostraron la efectividad de la propuesta para identificar los modelos con mayor precisión. En los experimentos se logró identificar el porcentaje de veces en las que los modelos obtendrían resultados similares y en qué porcentaje habría

una diferencia significativa. Si bien, se encontró que en la mayoría de los casos (70 % y 80 %) los modelos eran similares, el restante señalaba diferencia significativa. Esa discrepancia prueba que si no se realizara la evaluación propuesta, podríamos llegar a cualquier de las dos conclusiones (que existe diferencia o que son modelos similares). Con la propuesta aplicada podemos identificar la tendencia en porcentajes de diferencia o similitud que tienen los modelos.

Si bien, el presente trabajo está enfocado y ha sido validado en un entorno de Speech Emotion Recognition. La metodología llevada a cabo puede ser replicado en cualquier ambiente donde se necesite comparar múltiples clasificadores multiclase.

6.2. Contribuciones

Se presentó un framework para comparar modelos de deep learning, mediante test estadísticos no paramétricos. La propuesta se aplicó y evaluó en el entrenamiento de 3 modelos de Deep Learning para los dataset RAVDESS y IEMOCAP. Los experimentos demostraron la efectividad de la propuesta para identificar si un modelo es significativamente superior a otro.

Adicionalmente, presentó una propuesta de metodología para estandarizar datasets y métricas para evaluar precisión, para el área de Speech Emotion Recognition.

6.3. Trabajo futuro

Próximos trabajos deberían evaluar la propuesta con modelos más complejos y datasets con más muestras, afrontando las dificultades que conlleva en tiempo de entrenamiento y complejidad computacional. Algunas alternativas para solucionar este problema son distribuir el entrenamiento en múltiples nodos, lo cual permitiría tener más muestras para llegar a conclusiones más acertadas.

Una limitación de este trabajo es que limita la evaluación a una sola estrategia de preprocesamiento para todos los modelos. Sin embargo, usar diferentes estrategias en el preprocesamiento son comunes en el estado del arte (Ejm: Diferentes parámetros al generar el espectrograma). Y estos pasos previos afectan y varían la precisión de

modelos. Por lo que, una trabajo futuro debería contemplar el preprocesamiento, como parte del modelo.

Bibliografía

[Agarwal et al., 2018] Agarwal, B., Nayak, R., Mittal, N., and Patnaik, S. (2018). Deep learning-based approaches for sentiment analysis.

[Akçay and Oğuz, 2020] Akçay, M. B. and Oğuz, K. (2020). Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Communication*, 116:56–76.

[Burkhardt et al., 2005] Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., and Weiss, B. (2005). A database of german emotional speech. In *Ninth European Conference on Speech Communication and Technology*.

[Burkhardt and Schröder, 2008] Burkhardt, F. and Schröder, M. (2008). Emotion markup language: Requirements with priorities. *W3C Incubator Group report*.

[Busso et al., 2008] Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

[Dhall et al., 2018] Dhall, A., Kaur, A., Goecke, R., and Gedeon, T. (2018). Emotiw 2018: Audio-video, student engagement and group-level affect prediction. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 653–656.

[Dzedzickis et al., 2020] Dzedzickis, A., Kaklauskas, A., and Bucinskas, V. (2020). Human emotion recognition: Review of sensors and methods. *Sensors*, 20(3):592.

[Errica et al., 2019] Errica, F., Podda, M., Bacciu, D., and Micheli, A. (2019). A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893*.

[Fayek et al., 2017] Fayek, H. M., Lech, M., and Cavedon, L. (2017). Evaluating deep learning architectures for speech emotion recognition. *Neural Networks*, 92:60–68.

[Feidakis et al., 2011] Feidakis, M., Daradoumis, T., and Caballé, S. (2011). Emotion measurement in intelligent tutoring systems: what, when and how to measure. In *2011 Third International Conference on Intelligent Networking and Collaborative Systems*, pages 807–812. IEEE.

[Khalil et al., 2019] Khalil, R. A., Jones, E., Babar, M. I., Jan, T., Zafar, M. H., and Alhussain, T. (2019). Speech emotion recognition using deep learning techniques: A review. *IEEE Access*, 7:117327–117345.

[Kwon et al., 2020] Kwon, S. et al. (2020). A cnn-assisted enhanced audio signal processing for speech emotion recognition. *Sensors*, 20(1):183.

[Livingstone and Russo, 2018] Livingstone, S. R. and Russo, F. A. (2018). The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391.

[Martin et al., 2006] Martin, O., Kotsia, I., Macq, B., and Pitas, I. (2006). The enterface'05 audio-visual emotion database. In *22nd International Conference on Data Engineering Workshops (ICDEW'06)*, pages 8–8. IEEE.

[NIH, 2020] NIH (2020). What is voice? what is speech? what is language? `https://www.nidcd.nih.gov/health/what-is-voice-speech-language`.

[Nwe et al., 2003] Nwe, T. L., Foo, S. W., and De Silva, L. C. (2003). Speech emotion recognition using hidden markov models. *Speech communication*, 41(4):603–623.

[Petrushin, 1999] Petrushin, V. (1999). Emotion in speech: Recognition and application to call centers. In *Proceedings of artificial neural networks in engineering*, volume 710, page 22. Citeseer.

[Ringeval et al., 2015] Ringeval, F., Schuller, B., Valstar, M., Jaiswal, S., Marchi, E., Lalanne, D., Cowie, R., and Pantic, M. (2015). Av+ ec 2015: The first affect recognition challenge bridging across audio, video, and physiological data. In *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, pages 3–8.

[Schuller et al., 2004] Schuller, B., Rigoll, G., and Lang, M. (2004). Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages I–577. IEEE.

[Schuller et al., 2011] Schuller, B., Valstar, M., Eyben, F., McKeown, G., Cowie, R., and Pantic, M. (2011). Avec 2011–the first international audio/visual emotion challenge. In *International Conference on Affective Computing and Intelligent Interaction*, pages 415–424. Springer.

[Schuller, 2018] Schuller, B. W. (2018). Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends. *Communications of the ACM*, 61(5):90–99.

[Stapor et al., 2021] Stapor, K., Ksieniewicz, P., García, S., and Woźniak, M. (2021). How to design the fair experimental classifier evaluation. *Applied Soft Computing*, 104:107219.

[Stasiak et al., 2019] Stasiak, B., Opałka, S., Szajerman, D., and Wojciechowski, A. (2019). Convolutional neural networks in speech emotion recognition–time-domain and spectrogram-based approach. In *International Conference on Information Technologies in Biomedicine*, pages 167–178. Springer.

[Sterling and Kazimirova, 2019] Sterling, G. and Kazimirova, E. (2019). End-to-end emotion recognition from speech with deep frame embeddings and neutral speech handling. In *Future of Information and Communication Conference*, pages 1123–1135. Springer.

[Stola et al., 2018] Stola, M., Lech, M., Bolia, R. S., and Skinner, M. (2018). Acoustic characteristics of emotional speech using spectrogram image classification. In *2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–5. IEEE.

[Sun et al., 2020] Sun, Z., Yu, D., Fang, H., Yang, J., Qu, X., Zhang, J., and Geng, C. (2020). Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison. In *Fourteenth ACM Conference on Recommender Systems*, pages 23–32.

[Swain et al., 2018] Swain, M., Routray, A., and Kabisatpathy, P. (2018). Databases, features and classifiers for speech emotion recognition: a review. *International Journal of Speech Technology*, 21(1):93–120.

[Ververidis et al., 2004] Ververidis, D., Kotropoulos, C., and Pitas, I. (2004). Automatic emotional speech classification. In *2004 IEEE international conference on acoustics, speech, and signal processing*, volume 1, pages I–593. IEEE.

[Whitehill, 2018] Whitehill, J. (2018). Climbing the kaggle leaderboard by exploiting the log-loss oracle. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

[Zhao et al., 2019] Zhao, J., Mao, X., and Chen, L. (2019). Speech emotion recognition using deep 1d & 2d cnn lstm networks. *Biomedical Signal Processing and Control*, 47:312–323.

[Zhao et al., 2018] Zhao, Z., Zhao, Y., Bao, Z., Wang, H., Zhang, Z., and Li, C. (2018). Deep spectrum feature representations for speech emotion recognition. In *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and first Multi-Modal Affective Computing of Large-Scale Multimedia Data*, pages 27–33.

[Zheng et al., 2020] Zheng, C., Wang, C., and Jia, N. (2020). An ensemble model for multi-level speech emotion recognition. *Applied Sciences*, 10(1):205.