Facultad de Ingeniería
Escuela de Ingeniería Informática

EVALUACIÓN DE DATASETS PARA RECONOCIMIENTO DE EMOCIONES EN ROBÓTICA SOCIAL

Por

Felipe Andrés Rojas Gonzalez

Trabajo realizado para optar al Título de
INGENIERO (CIVIL) EN INFORMÁTICA
Prof. Guía: Ana Aguilera Faraco
Octubre 2023

Resumen

El reconocimiento de emociones es un factor importante en la interacción humano-computador, dado que permite dirigir adecuadamente las acciones del robot en función del entorno que percibe. Los robots suelen utilizar sensores para capturar su entorno. Existen diversas propuestas notablemente en Deep Learning que han surgido para el reconocimiento de emociones. Como parte de la metodología se utilizan datasets para entrenar y construir modelos adecuados para el reconocimiento de emociones, sin embargo, estos datasets tienen problemas de calidad de los datos, no están balanceados, falta de modalidades y que se llevan a cabo en ambientes controlados. Lo cual no es lo más adecuado y similar al ambiente en donde ocurre la interacción humano-robot. Es por esto que en este trabajo de título se evaluaron distintos datasets en una arquitectura con técnicas Deep Learning ya implementada para detectar las emociones de las personas a través de distintas modalidades y todo esto enfocado en el contexto de la robótica social. Para ir realizando un análisis de datos pertinente a cada dataset por sus modalidades a través de las métricas de evaluación F1 score y exactitud. Con el fin de ir comparando e interpretando los resultados obtenidos y tomar ciertas decisiones en el ámbito de la robótica que permitirán realizar diferentes tareas en nuestra sociedad a futuro.

Como resultado de este trabajo se evaluaron 4 dataset el Iemocap, Sfew, Afew y el Meld pasando por la misma arquitectura y realizando las diferentes evaluaciones unimodales y multimodales, se puede decir que el dataset Meld se adecua de mejor manera a la arquitectura con respecto a sus resultados y la emoción más destacable es la neutral en texto con un F1 score del 66,0 % y la exactitud de un 72,0 %. Por otro lado, en su enfoque multimodal se encuentra la emoción joy en la modalidad de texto+rostro con un F1 score de 56,0 % y una exactitud del 73,1 %.

Agradecimientos

Quiero agradecer en primera instancia a mi familia que son los más importantes para mí, quienes estuvieron en cada momento de toda mi etapa universitaria, me han apoyado en cada decisión tomada, me han formado a ser quien soy hoy en la vida. A mis amigos y compañeros de carrera que han sido parte de mi proceso, les deseo lo mejor de los éxitos en su vida en general, ya que son parte de mi historia de vida con quienes compartimos muy buenas experiencias a lo largo de la carrera universitaria. A mi profesora guía Dra. Ana Aguilera Faraco por brindarme sus consejos, conocimiento y la confianza para sacar adelante mi trabajo de título, además de estar siempre cuando lo necesitaba y por sobre todo su paciencia y su capacidad de entender en ciertos momentos complejos que tenía que afrontar de salud y diversos problemas personales. Finalmente, agradecer a la Escuela de Ingeniería en Informática, quienes me entregaron las diferentes herramientas para formarme como profesional, a lo cual puedo destacar que desde el primer día que ingrese a la carrera, todo el ambiente entre estudiantes y profesores siempre ha sido espectacular donde los docentes son muy cercanos, transmiten una confianza extraordinaria al momento de aclarar dudas y de enseñar de la mejor manera posible.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En nuestra sociedad están ingresando nuevos actores para convivir con nosotros, como son los robots sociales, que tienen la particularidad de realizar diversas tareas, ya sea en nuestro hogar, en las calles y en muchos otros lugares en los que nos encontremos con el fin de hacernos la vida más simple. Estos robots ahora pueden realizar una serie de tareas de forma autónoma y sin supervisión humana, tales como proporcionar información y asistencia a los clientes en tiendas, aeropuertos y otros lugares públicos. Por otro lado, pueden realizar tareas de limpieza y mantenimiento en hogares, oficinas y espacios industriales. Sin embargo, si van a ser aceptados por los usuarios humanos, existe la necesidad de centrarse en la forma de interacción humano-robot que dichos usuarios consideran aceptable [8]. Por otro lado, estos robots sociales tienen la facilidad de comunicarse con los seres humanos y obtener cierta información de nosotros para tomar ciertas decisiones. En general no existe una definición universal sobre robótica social, ya que falta consenso en comprender qué hacen estos robots y qué, específicamente, los hace sociales. Dentro del campo de la interacción humano-robot, los robots sociales asumen un papel especial y entran en la categoría de ”interacción próxima”, en la que ”los humanos y los robots interactúan como compañeros” [9]. Los robots tienen que ser espontáneos, educados y deben aprender a reaccionar de acuerdo a la carga emocional del ser humano, proporcionando un ambiente amigable. Sin la retroalimentación emocional de los humanos, será muy difícil que los robots interactúen con los humanos de forma natural.

El Deep Learning es un subconjunto del Machine Learning, que es esencialmente una Red Neuronal con tres o más capas. Estas Redes Neuronales intentan simular el comportamiento del cerebro humano, aunque lejos de igualar su capacidad, lo que le permite aprender a partir de grandes cantidades de datos [10]. Se ha logrado trabajar con técnicas Deep Learning en varias áreas como la seguridad, la salud y las interfaces hombre-máquina. Con el fin de desarrollar técnicas para interpretar y codificar expresiones, por ejemplo, faciales y extraer estas características a través de las Redes Neuronales para una mejor predicción con

respecto a las emociones presentadas [11]. En los trabajos de reconocimiento de emociones, los modelos de Deep Learning son ampliamente utilizados porque son especialmente buenos en la clasificación. El análisis de sentimiento con algoritmos de Deep Learning pueden trabajar con varios tipos de datos como por ejemplo la cara, el cuerpo, la postura y los datos de contexto [12]. La interacción humano-robot es fundamental para el futuro de la sociedad, ya que nos ayudaría a obtener una cierta cercanía a los robots a partir de un diálogo, ya sea en forma remota o presencialmente, con lo cual trae varios beneficios para diversas tareas en las oficinas, hospitales o en el mismo hogar. Esta interacción permite al robot proporcionar entretenimiento, enseñanza, comodidad, asistencia a niños o ancianos y personas discapacitadas [13].

En los últimos años, con el rápido desarrollo del reconocimiento de patrones y la inteligencia artificial, se han realizado más y más investigaciones en el campo de la interacción humano-robot [14]. El reconocimiento de emociones, como medio importante de interacción inteligente entre humanos-robot tiene un amplio historial de aplicaciones. Se ha aplicado en los campos de la medicina auxiliar, la educación a distancia, la seguridad pública [15]. El reconocimiento de expresiones faciales, por ejemplo, extrae la información que representa las características de la expresión facial de las imágenes originales a través de la imagen de la computadora [16]. Lo cual se logra bajo una Red Neuronal con una cierta arquitectura y técnicas deep learning que van a ayudar a entrenar los datos procesados para poder obtener resultados con mayor precisión. Con ello se podrá aplicar el reconocimiento de emociones a partir de robots que pueden comprender el estado mental y la intención de las personas de acuerdo con el reconocimiento de emociones y luego dar las respuestas apropiadas.

El contexto en el que se desarrolla el trabajo es la interacción entre un humano y un robot, los cuales van a detectar las diversas emociones de las personas por medio del área visual, de lo hablado, los gestos, texto, y muchas otras modalidades. El reconocimiento de emociones a partir de imágenes se ha convertido en un tema apasionante en robótica, con muchos métodos y técnicas aplicables en robots, pero no necesariamente diseñados exclusivamente para robots [17]. Para que la comunicación sea efectiva y se mejora la HRI es importante reconocer las emociones que vienen de los humanos de modo que los robots puedan adaptar su comportamiento.

- Problema

  Los datasets usados para el entrenamiento de modelos en ER no son aptos para llevarlo a un contexto de robótica social, ya sea en relación con la calidad de los datos, datasets no están balanceados, la falta de modalidades presentes en los datasets, el ambiente controlado en donde se realiza el reconocimiento de emociones y además que algunos datasets no están disponibles de forma pública en internet. En robótica social, cuando se quiere llevar el reconocimiento a un ambiente real, la situación se

vuelve más compleja, ya que se tiene un robot que a través de los sensores está realizando capturas y van ocurriendo muchos factores externos como la calidad de la imagen, la luminosidad influye, la existencia de ruido, y muchas otras cosas. Por lo que en condiciones reales cambia bastante con respecto a condiciones controladas de laboratorio [18]. Los dataset son esenciales para el entrenamiento de los modelos de Deep Learning por lo que una selección adecuada de los mismos va a impactar en la calidad de estos modelos.

- Propuesta de solución

  Aplicar ciertos criterios que permitan seleccionar los datasets adecuados para el entrenamiento de los modelos de Deep Learning tomando en cuenta la calidad de los datos, las modalidades que contiene, el entorno de desarrollo donde se llevó a cabo y adaptarlo con la arquitectura ya implementada para realizar un análisis de los datos por cada dataset a partir de una selección de métricas de evaluación como el F1 score y la exactitud donde se obtendrán resultados que serán comparados e interpretados para llegar a tomar decisiones de qué modalidad y dataset son los más recomendables para la arquitectura y para el ámbito de la robótica social [17].

- Principales contribuciones

  Se enfoca en aplicar criterios de evaluación para la selección de dataset en relación con la falta de calidad de los datos, falta de modalidades, dataset no balanceados y el ambiente en donde se trabajó (controlado o no controlado). Se configura la arquitectura para adecuar los dataset seleccionados y realizar el entrenamiento de los datos. Donde se procesan de forma individual cada modalidad y se juntan por medio del método Embracenet+ para fusionar y realizar la predicción final. Se realizó un análisis de datos para seleccionar las métricas de evaluación a los 3 modelos individuales y a todo el sistema compuesto se evaluó por separado donde las emociones a analizar se aplicaron individualmente y para tener un enfoque multimodal se lleva a cabo 4 experimentos entre cada modalidad rostro+audio, audio+texto, rostro+texto, rostro+audio+texto las cuales se mencionan de esta manera en las diferentes tablas (F+A, A+T, F+T, F+A+T) lo que se conoce como un estudio de ablación. Permite realizar varias comparaciones entre las distintas modalidades y dataset a trabajar [3]. Con lo cual tiene relevancia este trabajo en el ámbito del reconocimiento de emociones, ya que es aplicado en la arquitectura a través de distintos dataset previamente seleccionados, adecuados, analizados y comparados para obtener mejores resultados y saber qué dataset son más recomendables como también la modalidad para lograr una interacción humano robot más adecuada a la realidad. Lo que va a permitir que el estudio de los diferentes dataset puedan servir o sean un aporte y un camino para futuros trabajos investigativos.

- Estructura del documento

  Este informe se encuentra estructurado de siguiente forma: en el capítulo 1 se presenta la introducción enfatizando en el contexto y la problemática a nivel general, luego se encuentra el capítulo 2 que especifica el marco conceptual y el estado del arte, posteriormente en el capítulo 3 contiene la definición del problema, la solución propuesta, la importancia del trabajo ya sea en el área científica o social como también se mencionaría el objetivo general y el específico y también la metodología a trabajar.

  Para avanzar al capítulo 4 se explican los materiales que consisten en los dataset y casos de estudio a utilizar, como también se presentan los criterios de inclusión y exclusión. Se realiza análisis descriptivo y la interpretación de los resultados. Por otro lado, están los métodos que abarca las técnicas de análisis y la descripción de procesos.

  Se prosigue con el capítulo 5 donde se explica el proceso y las condiciones de experimentación, detalladamente a partir del software y hardware utilizado.

  El capítulo 6 se muestra la ejecución de los experimentos mediante visualizaciones, el proceso de preprocesamiento y procesamiento de la data en cada dataset.

  El siguiente capítulo 7 es de los resultados donde se lleva a cabo las evaluaciones unimodales y multimodales. Se realizó el análisis de resultados enfocándose en los dataset utilizados, se da a conocer los criterios de evaluación como la discusión de resultados.

  Para terminar se tiene el capítulo 8 donde se enfoca en la conclusión del trabajo donde se realiza una breve introducción a lo realizado, una descripción del cumplimiento de los objetivos iniciales, presentación de limitaciones y proyecciones del trabajo.

Capítulo 2

Marco Conceptual y Estado del Arte

2.1. Marco Conceptual

En este capítulo se definieron los conceptos más importantes a trabajar para la evaluación de los datasets para reconocimiento de emociones en robótica social en un orden de lo más general a lo más específico, a lo cual se abarcará el Reconocimiento de emociones, Robots sociales, Interacción Humano-Robots, Datasets, Deep Learning, Análisis de los datos y Métricas de Evaluación.

2.1.1. Reconocimiento de emociones

A través de las emociones existe una área que se dedica al reconocimiento de emociones a través de distintas modalidades, ya sea por audio, texto o simplemente por las expresiones del rostro que se van obteniendo por medio de una fuente o más fuentes de datos con la finalidad de detectar el estado emocional. El reconocimiento de emociones es un área importante de investigación para permitir la interacción humano-computadora, en donde los robots interactúen de manera más humana y empática con las personas, lo que a su vez mejora la experiencia del usuario y abre nuevas oportunidades en una variedad de aplicaciones, desde la asistencia en el hogar hasta la atención médica y la educación [19]. A menudo, el éxito de las interacciones depende de la inteligencia emocional de los involucrados [20]. El estudio de la emoción tiene como objetivo minimizar la brecha entre humanos y ordenadores, analizando textos, discursos e imágenes para predecir y clasificar los sentimientos o sensaciones de las personas con respecto a algo [21].

2.1.1.1. Reconocimiento de emociones faciales

Es una tecnología utilizada para analizar los sentimientos de diferentes fuentes como fotos y videos. Donde el robot o la computadora puede reconocer e interpretar emociones

humanas y estados afectivos donde se basan en tecnologías de Inteligencia Artificial [22].

2.1.1.2. Reconocimiento de emociones en audio

Permite llevar a cabo la tarea de reconocer los aspectos emocionales del habla, independientemente de los contenidos semánticos u oraciones por parte de la persona. Se han utilizado muchas técnicas para extraer emociones de las señales incluidas técnicas de análisis y clasificación del habla bien establecidas. Donde se dividen en dos fases conocidas como la extracción de características en distintas fuentes de datos y la fase de clasificación de características utilizando clasificadores lineales y no lineales [23]. Algunas técnicas son: Extracción de Características Acústicas, Modelos de Emoción Mel-Cepstral Coefficients (MFCC), Modelos Basados en Prosodia, Redes Neuronales Convolucionales (CNN) para Espectrograma, entre otras.

2.1.1.3. Reconocimiento de emociones en texto

Es fundamentalmente un problema de clasificación basado en el contenido, que incluye nociones de procesamiento del lenguaje natural y campos de aprendizaje profundo. Donde las técnicas a ocupar son el NLP (Natural Language Processing) que mejoran el rendimiento de los métodos basados en el aprendizaje al incorporar las características semánticas y sintácticas del texto y poder detectar las emociones humanas [24].

2.1.2. Reconocimiento de emociones multimodal

El procesamiento de emociones multimodal continúa teniendo una aplicación generalizada en la ciencia [25]. Esta expansión ayudaría a comprender mejor las emociones con la experiencia de otras modalidades relacionadas con el estudio (video, audio, rostros, etc.) Se integran muchos enfoques y estrategias diferentes para alcanzar el objetivo del estudio. Muchos de ellos usan técnicas de big data, principios semánticos y aprendizaje profundo [26].

El aprendizaje multimodal es una forma de aprendizaje mucho más eficiente que la unimodal [27]. Los estudios también intentaron integrar señales de diferentes modalidades para una mejor eficiencia y precisión, como expresiones faciales y audio, audio y texto escrito, señales fisiológicas y varias combinaciones de estas modalidades [28].

2.1.3. Robots Sociales

Los robots sociales están capacitados para comunicarse con los usuarios, para ayudar a simplificar las tareas a realizar, ya sea ofreciendo información o interactuando en entornos de atención médica, como por ejemplo en intervenciones de salud mental para niños [29]. Estos robots interactúan con los humanos y entre sí de una manera socialmente aceptable,

transmiten intenciones de una manera perceptible para los humanos y están facultados para resolver objetivos con otros agentes, ya sean humanos o robots [30].

2.1.4. Interaccion Humano-Robot

Interacción humano-robot (HRI) es la comunicación del futuro en donde se van transmitiendo información y emociones de diferentes formas lo cual permitirá que el robot las capte por distintas modalidades y tome ciertas decisiones para realizar tareas o acciones al respecto. La HRI se ocupa específicamente de los algoritmos, las técnicas, los modelos, y los marcos necesarios para construir sistemas robóticos que participen en interacciones sociales con humanos. Donde abordan desafíos como percibir a los humanos y sus actividades, generar y comprender expresiones verbales, modelar, expresar y comprender estados emocionales [31].

2.1.5. Datasets

Los Datasets son un conjunto de datos que son estructurados, es decir, son modelos de datos predefinidos, fáciles de buscar y analizar, como también puede ser no estructurados, donde la data puede venir en texto, imágenes, sonido, videos u otros formatos. Los datasets pueden contener información, como registros médicos o registros de seguros, para que los utilice un programa que se ejecuta en el sistema [32].

Los conjuntos de datos constan de imágenes, videos, audio o texto. Para ellos se mencionarán algunos ejemplos en el contexto de ER como son:

- Desafío EmotiW [33]: Contiene audio, video y metadatos. Los metadatos son compuestos por la identidad, la edad y el género del actor, los datos se recopilan de películas.

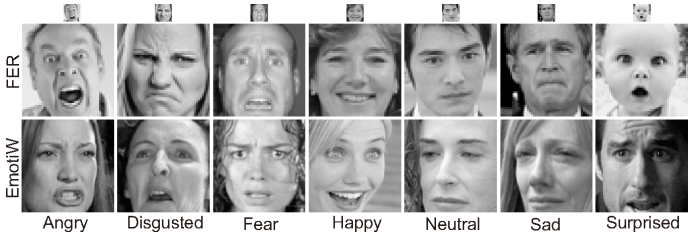


Figura 2.1: Dataset Emotiw

- CMU Multi-modal Opinion Sentiment and Emotion Intensity (MOSEI) [34]: Es el mayor conjunto de datos de análisis de sentimiento a nivel de oración y reconocimiento de emociones en videos en línea que están etiquetados en las seis emociones comunes.

Figura 2.2: Dataset Mosei

2.1.6. Deep Learning

Deep Learning consiste en varias capas de redes neuronales, que son algoritmos modelados sobre la forma en que funcionan los cerebros humanos. El entrenamiento con grandes cantidades de datos es lo que configura las neuronas en la Red Neuronal. El resultado es un modelo de Deep Learning que una vez entrenado procesa nuevos datos. Los modelos de deep learning toman información de múltiples fuentes de datos y analizan esos datos en tiempo real, sin necesidad de intervención humana [35]. Es fundamental para resolver muchas tareas relacionadas con la inteligencia artificial, reconocimiento visual de objetos o patrones, la percepción del habla y la compresión del lenguaje. Se ha demostrado que las representaciones de alto nivel aprendidas brindan resultados de calidad en muchos problemas de aprendizaje [36].

2.1.7. Análisis de datos

El análisis de datos es un conjunto de técnicas que se utilizan con fines descriptivos para entender e interpretar datos y también se hacen con fines predictivos. En donde se define también como un proceso de inspección, limpieza, transformación y modelado de datos con el objetivo de descubrir información útil, informando conclusiones y apoyando la toma de decisiones [37] [38].

2.1.8. Métricas de Evaluación

Las métricas de evaluación permiten medir el rendimiento de algunas técnicas de análisis. En particular, en este trabajo se usaron el F1 score y la exactitud, las cuales son importantes para verificar si los resultados obtenidos son de relevancia en comparación a otros trabajos de investigación, como también cuantificar el desempeño de la solución y conocer el rendimiento en términos de eficacia y precisión. El F1 score permite combinar en un solo valor tanto las cantidades que el modelo es capaz de identificar como la calidad del modelo a la hora de clasificar dichas emociones y, por otro lado, se tiene la exactitud que va a ayudar a detectar todas las predicciones acertadas por el modelo representado en porcentaje.

La evaluación de un modelo es una parte importante de la creación de un modelo de aprendizaje automática eficaz. La métrica de evaluación de clasificación más frecuente es la exactitud que permite dar un significado a los resultados obtenidos, como existen muchos otros que permiten apoyar la veracidad de las métricas [39].

2.2. Estado del Arte

En esta sección se realizó la búsqueda de documentación útil utilizando bases de datos científicas importantes para la computación como ACM Digital Library, Science Direct, Springer Link y muchas otras. Para entregar una información actualizada de que se está haciendo en el área de reconocimiento de emociones (ER) y la interacción humano-robot (HRI), a modo de comparar los dataset con diversos criterios que se denotaron en una tabla bien detallada. Las keywords utilizadas en la búsqueda son: Multimodal emotion recognition; in-the-wild datasets; Deep Learning.

2.2.1. A multimodal emotion recognition method based on facial expressions and electroencephalography

En este trabajo se propuso un método de reconocimiento de emociones multimodal para establecer un sistema HRI con un bajo sentido de desarmonía. Este método se basa en las expresiones faciales que se usaron con CNN para clasificar las expresiones faciales y la electroencefalografía (EEG) con una SVM para clasificar las características de las señales EEG. Se investigaron en dataset públicos como son el FER2013 (el cual contiene 7 tipos de expresiones faciales) y el SEED-IV (contiene datos de EEG y movimientos oculares de 15 participantes). El método de reconocimiento de emociones multimodal se combinó con el sistema HRI utilizando el método de Monte Carlo para obtener los resultados del reconocimiento multimodal, donde logró una tasa de reconocimiento del 83,33 % tanto para las expresiones faciales como para el EEG [40].

2.2.2. A Face Emotion Recognition Method Using Convolutional Neural Network and Image Edge Computing

Este trabajo consiste en un método de reconocimiento de expresiones faciales basado en redes neuronales convolucionales (CNN) y reconocimiento de bordes de imágenes. Lo cual se va normalizando la imagen de la expresión facial y extrayendo los bordes de cada capa de la imagen en el proceso de convolución. La información de borde extraída se superpone en cada imagen característica para retener la información de estructura de borde de la imagen de textura. La reducción de la dimensionalidad de las características

implícitas extraídas se procesa luego utilizando el método de agrupación máxima. Finalmente, la impresión de la imagen de la muestra de prueba se clasifica y reconoce utilizando el clasificador softmax. Para confirmar la solidez de este método de detección de expresiones faciales en entornos complejos, se diseñaron experimentos de simulación combinando científicamente la base de datos de expresiones faciales de Fer2013 con el conjunto de datos de LFW (Labeled Faces in the Wild). Los resultados experimentales muestran que el algoritmo propuesto puede lograr una tasa de detección promedio del 88,56 % con menos iteraciones, y la velocidad de entrenamiento del conjunto de entrenamiento es aproximadamente 1,5 veces más rápida que el algoritmo de contraste [16].

2.2.3. Multimodal Emotion Recognition Based on Ensemble Convolutional Neural Network

Para mejorar la precisión de la detección de emociones, se propone un modelo de red neuronal convolucional de conjunto (ECNN) que se utiliza para extraer automáticamente la correlación entre el EEG multicanal y las señales fisiológicas periféricas. Primero se diseñó cinco redes convolucionales y se utilizó la capa Global Average Pooling (GAP) en lugar de la capa totalmente conectada con el fin de abordar el problema de sobreajuste, que también mejorará la precisión y la estabilidad en función de la utilización eficaz de la información extraída por la red neuronal convolucional (CNN). Luego se usaron múltiples estrategias de votación para crear un modelo de conjunto. Finalmente, este modelo divide las emociones en cuatro categorías. Basado en una simulación del conjunto de datos DEAP [41].

2.2.4. Multimodal emotion recognition using deep canonical correlation analysis:

En este trabajo se realizó un análisis de correlación canónica profunda (DCCA) para el reconocimiento de emociones multimodal. La idea básica detrás de DCCA es transformar cada modalidad por separado y coordinar diferentes modalidades en un hiperespacio mediante el uso de restricciones de análisis de correlación canónica especificadas. Se evaluó el desempeño de DCCA en cinco conjuntos de datos multimodales: los conjuntos de datos SEED, SEEDIV, SEEDV, DEAP y DREAMER. Los resultados experimentales demuestran que DCCA logra tasas de precisión de reconocimiento de vanguardia en los cinco conjuntos de datos: 94,58 % en el conjunto de datos SEED, 87,45 % en el conjunto de datos SEEDIV, 84,33 % y 85,62 % para dos tareas de clasificación binaria y 88,51 % para una tarea de clasificación de cuatro categorías en el conjunto de datos DEAP, 83,08 % en el conjunto de datos SEEDV y 88,99 %, 90,57 % y 90,67 % para tres tareas de clasificación binaria en el conjunto de datos DREAMER. También compara la solidez del ruido de DCCA con los métodos existentes al agregar cantidades variables de ruido al conjunto de datos

de SEEDV. Los resultados experimentales muestran que DCCA es más robusto. Al visualizar la distribución de características usando tSNE y calculando la información mutua entre diferentes modalidades antes y después de usar DCCA, las características transformadas por DCCA de diferentes modalidades pueden ser más uniformes y emocionalmente distinguidas [42].

2.2.5. Facial emotion recognition in real-time and static images

En este trabajo, la detección de emociones se realizó tanto en tiempo real como en imágenes fijas. Este proyecto utilizó las bases de datos CohnKanade (CK) y Extended CohnKanade (CK+). Esta base de datos contiene una gran cantidad de imágenes fijas de 640 x 400 píxeles que se pueden usar con cámaras web en tiempo real. La expresión de destino para cada secuencia en el conjunto de datos está completamente codificada en FACS (Sistema de codificación de acción facial) y las etiquetas de emoción se verifican y validan. Por lo tanto, para detectar emociones, primero se debe usar el filtro HAAR de OpenCV para detectar rostros en imágenes fijas o videos en tiempo real. Una vez que se reconoce la cara, se puede recortar y procesar para detectar más rasgos faciales. Luego, el conjunto de datos se entrena utilizando marcadores faciales utilizando un algoritmo de aprendizaje automático (máquina de vectores de soporte) y se clasifica de acuerdo con ocho emociones. Con SVM, se logra una precisión de alrededor del 93,7 % [43].

2.2.6. A multimodal emotional communication based humans-robot interaction system

Se ha propuesto un sistema de interacción humano-robot basado en la comunicación emocional multimodal (MECHRI). Esto incluye múltiples modos de comunicación emocional, como la voz, las expresiones faciales y los gestos. Los robots del sistema MECHRI pueden reconocer las emociones de las personas y reaccionar en respuesta a ellas. En primer lugar, el sistema MECHRI recopila datos emocionales multimodales mediante Kinect, eye tracker y equipos portátiles, etc. En segundo lugar, el reconocimiento de emociones basado en la fusión de información unimodal o multimodal se realiza en la estación de trabajo. En tercer lugar, los robots NAO, los robots móviles, brindan retroalimentación emocional a los humanos. Los experimentos de cuatro escenarios muestran que el sistema MECHRI puede lograr una comunicación emocional multimodal entre humanos y robots. Además, el reconocimiento se valida basándose en la fusión de información multimodal [44].

2.2.7. Multimodal emotion recognition with evolutionary computation for human-robot interaction

En este trabajo se explora las implicaciones del uso de bases de datos estándar para la evaluación de técnicas de reconocimiento de emociones, donde ampliaron la optimización evolutiva de ANN y HMM para el desarrollo de un sistema de reconocimiento de emociones multimodal, establecieron las pautas para el desarrollo de bases de datos emocionales del habla y las expresiones faciales, se establecieron reglas para la transcripción fonética del habla mexicana, y se evaluó la capacidad del sistema multimodal dentro del contexto del diálogo hablado entre un robot humanoide y humanos. El reconocimiento de emociones depende de la estructura de los subconjuntos de bases de datos utilizados para el entrenamiento y las pruebas, y también depende del tipo de técnica utilizada para el reconocimiento cuando una emoción específica puede ser altamente reconocida por una técnica específica, la optimización de los HMM condujo a una estructura de Bakis que es más adecuada para el modelado acústico de vocales específicas de la emoción, mientras que la optimización de las RNA condujo a una estructura de ANN más adecuada para el reconocimiento de expresiones faciales, algunas emociones pueden reconocerse mejor en función de los patrones de habla en lugar de los patrones visuales, y la integración ponderada del sistema multimodal de reconocimiento de emociones optimizado con estas observaciones puede lograr una tasa de reconocimiento de hasta el 97,00 % en las pruebas de diálogo en vivo con un robot humanoide [45].

2.2.8. Emotional expressión recognition with a cross-channel convolutional neural network for human-robot interaction

Se propone un modelo de red neuronal profunda que puede reconocer expresiones emocionales espontáneas y clasificarlas positiva o negativamente. El modelo fue evaluado en dos experimentos diferentes. Primero, se entrena la red usando dos conjuntos de datos. El conjunto de datos CohnKanade con ejemplos emocionales simulados y el corpus CAM3D con ejemplos espontáneos. Donde el modelo puede reconocer expresiones de emoción tanto simuladas como espontáneas. En el segundo experimento, se despliega una red en cabezas de robots humanoides que pueden identificar las expresiones emocionales positivas/negativas del sujeto. En este escenario, la cabeza del robot puede responder a las expresiones emocionales detectadas y proporcionar retroalimentación emocional [46].

2.2.9. Adaptative Feature Selection-Based AdaBoost-KNN with Direct Optimization for Dynamic Emotion Recognition in Human-Robot Interaction.

AdaBoostKNN se propone mediante la selección de características adaptativas con optimización directa de la detección dinámica de emociones en las interacciones humano-robot, donde las emociones dinámicas en tiempo real se detectan en función de las expresiones faciales. Esto permite que el robot comprenda las emociones humanas dinámicas y facilita la interacción humano-robot. En función de los puntos clave faciales extraídos por el modelo Candide3, se adopta la selección de funciones adaptativas, es decir, se completa la selección PlusL MinusR. Puede determinar las características que más contribuyen al reconocimiento de emociones, formando así la base de la clasificación de emociones. AdaBoostKNN ajusta los pesos de los datos de manera iterativa. Además, los parámetros óptimos globales se aproximan con optimización directa hasta que la tasa de reconocimiento alcanza su valor máximo. El desempeño experimental de la propuesta es verificado por un k-validación cruzada de pliegues. Los resultados muestran que la tasa de reconocimiento del enfoque propuesto es mayor que la del AdaBoostKNN. También es más alta que la tasa alcanzada por otros métodos de reconocimiento tradicionales, como AdaBoost, KNN y SVM. Los experimentos muestran las capacidades dinámicas del robot para comprender emociones en las interacciones humano-robot [47].

2.2.10. Emotion Recognition from Speech for an Interactive Robot Agent

Se enfocan en la percepción de emociones a partir de expresiones humanas que sustentan las interacciones humano-robot. El desarrollo de un sistema de reconocimiento de emociones para agentes robóticos interactivos implica varios pasos. El primer paso es seleccionar la base de datos de Berlín, que es el conjunto de datos apropiado para entrenar y probar el modelo desarrollado. El segundo paso importante es la extracción y selección de características emocionales apropiadas. El tercer paso es crear un esquema de clasificación adecuado. Se analiza el rendimiento de cada clasificador y se realizan comparaciones entre múltiples marcos de detección de emociones. Sobre la base de los resultados de estos estudios preliminares, se ha desarrollado una aplicación prototipo que permite el reconocimiento de emociones basado en la voz en tiempo real para su uso futuro en robots interactivos. En una serie anterior de pruebas, la aplicación alcanzó un nivel de rendimiento del 81 % al 92 %. Este enfoque prevé la integración de hardware de captura de voz, software de reconocimiento de emociones, dispositivos móviles y sistemas robóticos para respaldar las interacciones entre humanos y robots [48].

2.2.11. Facial Expressions Recognition for Human-Robot Interaction Using Deep Convolutional Neural Networks with Rectified Adam Optimizer

En este trabajo se presenta la interacción entre humanos y un robot NAO utilizando redes neuronales convolucionales profundas (CNN) basándose en un método de canalización de extremo a extremo que aplica dos CNN optimizadas, una para el reconocimiento facial (FR) y otra para el reconocimiento de expresiones faciales (FER) para obtener velocidad de inferencia en tiempo real. Se consideran dos modelos diferentes para FR, uno conocido por ser muy preciso, pero tiene una velocidad de inferencia baja (red neuronal convolucional basada en regiones más rápida) y otro que no es tan preciso, pero tiene una velocidad de inferencia alta (red neuronal convolucional de detector de disparo único). Para el reconocimiento de emociones se ha utilizado el aprendizaje por transferencia y el ajuste fino de tres modelos de CNN (VGG, Inception V3 y ResNet). Los resultados generales muestran que los modelos de red neuronal convolucional con detector de disparo único (SSD CNN) y red neuronal convolucional basada en regiones más rápidas (RCNN más rápido) para la detección de rostros comparten casi la misma precisión. En términos de FER, ResNet obtuvo la mayor precisión de entrenamiento (90,14 %), mientras que la red del grupo de geometría visual (VGG) tuvo una precisión del 87 % e Inception V3 alcanzó el 81 %. Los resultados muestran una mejora de más del 10 % cuando se usan dos CNN serializadas en lugar de usar FER-CNN [49].

2.2.12. Multimodal learning for facial expression recognition

Este artículo propone el aprendizaje multimodal mediante el reconocimiento de expresiones faciales (FER). El método de aprendizaje multimodal hace el primer intento de aprender una expresión general, teniendo en cuenta la textura de la imagen facial complementaria y la modalidad del punto de referencia. Para aprender la representación de cada modalidad y las correlaciones e interacciones entre las diferentes modalidades, se usa la Regularización estructurada (SR) para aplicar y aprender el ancho y la densidad específicos de cada modalidad. La introducción de SR tiene en cuenta una amplia gama de expresiones faciales que no solo manejan expresiones faciales sutiles, sino que también funcionan para varias entradas de imágenes faciales. La red de aprendizaje multimodal propuesta hace que el aprendizaje de expresión colaborativo a partir de entradas multimodales sea más adecuado para FER. Los resultados experimentales de las bases de datos CK+ y NVIE muestran la superioridad del método propuesto [50].

2.2.13. Information-Driven Multirobot Behavior Adaptation to Emotional Intention in Human-Robot Interaction.

Se han propuesto mecanismos para adaptar varios comportamientos de robots basados en información sobre el HRI. En este mecanismo, el friendQ de aprendizaje difuso basado en información (IDFFQ) selecciona políticas de comportamiento óptimas y se utilizan expresiones faciales que contienen información de identificación para comprender la intención emocional humana. El propósito es que los robots entiendan su comportamiento y se adapten a la intención emocional humana para que el HRI pueda funcionar sin problemas. Los resultados muestran que el IDFFQ propuesto reduce 51 pasos de aprendizaje en comparación con el aprendizaje AmigoQ basado en reglas de producción difusa (FPRFQ), y el tiempo computacional es aproximadamente 1/4 del tiempo consumido por FPRFQ. Además, la precisión del reconocimiento de emociones y la comprensión de las intenciones emocionales es del 80,36 % y del 85,71 %, respectivamente [51].

2.3. Comparación de dataset

En la tabla 2.1 se presentan los diferentes trabajos y dataset relacionados con el área de reconocimiento de emociones en robótica (estado del arte), considerando aspectos como:

- **Nombre del dataset**: Indica el nombre del dataset utilizado en el trabajo.
- **Autor**: Se indica la cita a utilizar donde se encuentra en formato IEEE.
- **Arquitectura**: Se mencionan los diferentes tipos de arquitectura que se aplicaron en los diferentes trabajos.
- **Emociones**: Se mencionan las distintas emociones que se encuentran dentro del dataset para el trabajo..
- **N°de participantes**: Se indica la cifra exacta de los individuos a participar en el experimento realizado.
- **Detalles**: Se categoriza las personas por su sexo con respecto a su cantidad y su participación en el trabajo.
- **Ambiente**: Señala el lugar en donde se trabaja si es bajo condiciones controladas o no.
- **Cámara**: Se indica el tipo de cámara a utilizar dentro del trabajo.
- **Adquisición**: Se indica los objetos tecnológicos que se utilizan a lo largo del trabajo.
- **Disponible**: Se indica si el dataset está disponible o no en forma pública.

Tabla 2.1: Conjuntos de datos para el reconocimiento de emociones en robótica

Nombre del dataset	Autor	Arquitectura	Emociones	N° Partici-pantes	Detalles	Ambiente	Cámara	Adquisición	Disponible
EEG,FER2013 SEED-IV	[40]	CNN	miedo,feliz, trsite,neutral	15	8 mujeres y 7 hombres entre 20 a 24 años	Controlado	Camara 2D	Robot Pepper (Robot Humanoide)	Si
Fer2013, LFW	[16]	CNN con un clasificador de Haar y un algoritmo AdaBoost.	temeroso,enojado, triste,feliz sorprendido, disgustado,neutral	?	?	controlado	?	Intel Core i5-6500 CPU de frecuencia 3,2 GHz, 16 GB de memoria y 6 GB de memoria GPU NVIDIA GeForce GTX 1060.	si
DEAP	[41]	ECNN, GAP, CNN	relajado,Depresion, excitación, miedo	32	?	controlado	?	dispositivo de adquisición de datos para ver videos	si
SEED,SEED-IV SEED-V, DEAP, DREAMER	[42]	DCCA, CCA	Feliz, triste, neutral, miedo, disgusto	16	6 hombres y 10 mujeres	controlado	?	gafas de seguimiento ocular SMI ETG	si
CK, CK+	[43]	Filtros Haar en OpenCV y SVM	Felicidad, neutra tristeza,ira, desprecio disgusto, miedo, sorpresa	?	?	controlado	Cámara web	?	si
?	[44]	MEC-HRI y el ELM	neutral, feliz, ira, sorpresa, miedo, asco y tristeza	?	?	controlado	cámara somatosensorial 3D	3 robots NAO, 2 robots móviles y equipos de adquisición de información emocional como eye tracker, Kinect y sensores portátiles	no
MX-Expressions y MX-Speech	[45]	Standard + GA HMM, PCA+ANN+GA	Ira,alegria, Neutro, tristeza	17	3 hombres y 6 mujeres para MX-Expressions y 3 hombres y 5 mujeres para MX-Speech	no controladas	camara de profundidad	Robot Humanoide (Bioloid)	no
CK y CAM3D	[46]	CCCNN	7 emociones y 12 emociones	5	?	controlado	Cámara RGB	computadora con procesador Intel XEON CPU de 2.4Ghz y un robot humanoide.	si
JAFFE	[47]	AFS-AdaBoost-KNN-DO	Felicidad,ira, tranquilo,triste, disgusto,sorpresa miedo,neutral.	10	10 voluntarios entre 18 y 28 años	controlado	?	2 robots moviles, 1 enrutador, 1 estación de trabajo de computación emocional, 1 equipo de transmisión de datos	si
Berlín	[48]	Perceptron multicapa	enojado,asco,feliz, triste,aburrido, angustia,neutral.	13	?	controlado	?	Prototype robot emotion recognition y una API de reconocimiento de Google	si
Fer2013,CK+, JAFEE y KDEF	[49]	CNN,SSD, Faster R-CNN y ResNet	enfado,asco,miedo alegria,neutral, tristeza y neutral.	?	?	controlado	2 cámaras de video RGB con 640×Resolución y 480 a 30 fotogramas por segundo.	Robot Nao	si
CK+, NVIE	[50]	Arquitectura de aprendizaje multimodal para Fer	Enojado,disgusto, miedo, feliz triste, sorpresa	123	Entre 18 y 50 años	controlado	cámaras AG-7500	2 hardware Panasonic sincro-nizado	si
Datos de construcción propia	[51]	Mecanismo de adaptación de comportamiento basado en IDFFQ	Felicidad,sorpresa,miedo neutral,asco tristeza,ira	12	Entre 20 a 65 años	controlado	?	3 robots moviles y computadoras personales, ARK-3500 con 4 núcleos (2,7 GHz), memoria (4 GB) y sistema Windows 7.	no

Capítulo 3

Definición del Problema

En este capítulo se explica la problemática a resolver junto con la solución propuesta y la importancia del trabajo dentro del contexto de robótica social. Por otro lado, se presenta su objetivo general y específicos, culminando con la metodología empleada en el trabajo.

3.1. Formulación del Problema

En primera instancia se consideró como base una arquitectura previamente diseñada por su autor original JuanPablo Heredia un estudiante de la Universidad Católica San Pablo del Perú [3], en donde se llevaron a cabo ciertas pruebas que consisten en el procesamiento de entrada, las cuales son el reconocimiento facial, transcripción de audio y extracción de funciones MFCC. Lo siguiente es que los cuadros de video se procesan en el momento de su captura, los datos de voz se transforman en texto y funciones MFCC, luego se llevan a un mecanismo de integración donde las modalidades de texto y audio se procesan y luego se fusionan las tres modalidades con el método embracenet+ para posteriormente realizar un procedimiento para cambiar el comportamiento del robot de acuerdo con la emoción reconocida [52]. Basándose en técnicas de Deep Learning en un ambiente controlado con distintos datasets a través de diferentes modalidades, por lo que se llegaron a resultados de gran nivel en comparación a otras investigaciones, pero la idea es llevarlo a un contexto de robótica social con datasets que permitan llevar a cabo resultados lo más realista posible bajo un ambiente no controlado para mejorar la interacción entre el ser humano y el robot. Los datasets que se utilizan actualmente para hacer ese reconocimiento son datasets generalmente de laboratorio (ambiente controlado), por lo cual no son adecuados y se necesita llevar el reconocimiento de emociones a la robótica social. Debido a que en un ambiente real todo se complica porque el robot mediante sus sensores está haciendo captura y puede ocurrir que la imagen se distorsione, la luminosidad influya, ruidos y muchos agentes externos [3].

Lo que se necesita es evaluar diferentes datasets en un ambiente no controlado, para ello se debe realizar un preprocesamiento, reentrenar los modelos existentes por cada modalidad, detallar a través de diferentes métricas y analizar qué dataset, modalidad y emociones son las más recomendables para trabajar en un ambiente real.

Cabe destacar que tanto la arquitectura como las evaluaciones unimodales y multimodales del dataset Iemocap son realizadas por JuanPablo Heredia [3]. Se utiliza esta arquitectura como base con la finalidad de tener un estándar y ocupar los mismos modelos para entrenar la data y predecir las emociones por medio de diferentes modalidades y comparar los resultados de los datasets en ambiente no controlado ingresados a la arquitectura que se mencionara con más detalle en el capítulo 4.2.1.

Iemocap es un dataset de laboratorio con el cual se entrena la arquitectura y su entrenamiento permite familiarizarse con la misma y adicionalmente sirve como referencia de comparación para los siguientes datasets. Normalmente, cuando se usan datasets in-the-wild las métricas bajan considerablemente por los diferentes factores externos que se presentan.

3.2. Solución Propuesta

Como se muestra en la figura 3.1 en primera instancia se van a seleccionar los diferentes datasets bajo criterios de inclusión y exclusión para obtener la mayor información posible, pasando por un preprocesamiento en el cual se realiza un análisis descriptivo de cada dataset escogido con la finalidad de presentar cómo está estructurado dicho dataset a trabajar, las cantidades de emociones presentes representado en gráficas, recortar rostros de ciertos actores y eliminar datos duplicados. Lo siguiente es el procesamiento de los datos en donde van a pasar las imágenes, audios y texto como datos de entrada a la arquitectura para realizar el entrenamiento individual por cada modalidad, obteniendo sus métricas con el objetivo de medir el rendimiento del modelo. Luego se ingresa al embracenet+ el cual es una arquitectura para la clasificación multimodal, en donde su arquitectura se puede apreciar en la figura 4.18, para fusionar las modalidades y generar la predicción final como también se obtienen las métricas de desempeño para proceder a las evaluaciones multimodales y un análisis de los resultados para saber cuál es el más adecuado con respecto a sus modalidades, calidad de los datos, ambiente en donde se trabaja y determinar si existen algunas carencias que pudiesen ser mejoradas. Esto permite una mayor optimización y análisis de los resultados que se van a ir obteniendo [53]. Por otro lado, investigar cuáles son las mejoras que se tienen que realizar sobre el dataset para que responda correctamente a lo que se necesita.

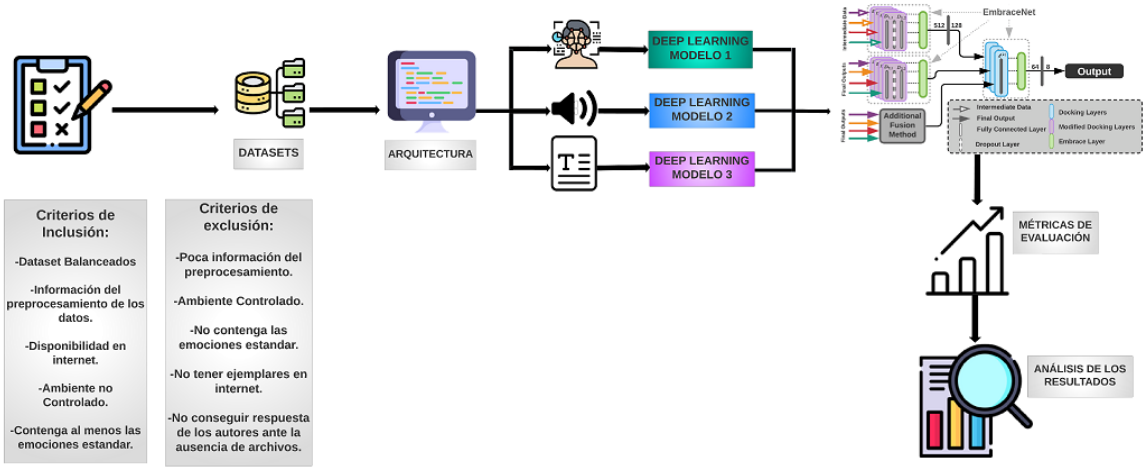


Figura 3.1: Diagrama de la Solución Propuesta

El problema es que se están realizando reconocimiento de emociones multimodal, entonces puede que existan dataset que no tenga todas las modalidades y si llegarán a tener la mayoría de las modalidades existentes, preguntarse si se puede trabajar bien con ellos. Estas emociones se van a clasificar en tristeza, neutralidad, ira y alegría para evaluar el desempeño de la arquitectura mediante las distintas modalidades [3].

Los métodos que se usarán para el reconocimiento de emociones en las diferentes modalidades están bajo un aprendizaje supervisado con la finalidad de predecir las emociones a partir del conjunto entrenado que corresponde a los datos de entrada y los resultados esperados. En este caso, como es un trabajo de título más de investigación, podemos rescatar un par de preguntas como, por ejemplo: ¿Cuáles son los dataset que se adaptan mejor a la arquitectura ya existente, según sus resultados?, ¿Se pueden incorporar mejoras a los dataset y a la arquitectura?, ¿Cuál dataset cumple de mejor manera los requisitos establecidos?, ¿Cuál es la diferencia relevante entre los distintos dataset? Y ¿Qué emoción es la más destacada por los distintos casos a trabajar?

3.3. Importancia del trabajo

En los HRI, los robots deben ser socialmente inteligentes. Deben poder responder adecuadamente a las señales afectivas y sociales humanas para participar de manera efectiva en las comunicaciones bidireccionales. La inteligencia social permitiría que un robot se relacione, comprenda, interactúe y comparta información con personas en entornos reales centrados en el ser humano [54]. El fin de todo esto es que la cercanía hacia los robots sociales será una cosa de futuro que nos ayudará para transmitirle nuestras emociones de

distintas formas, los cuales van a estar capacitados para realizar diversas tareas en el contexto de nuestra sociedad, ya sea en el ámbito del hogar, oficinas, hospitales, supermercado donde se brindaría cierta atención al ser humano [55].

El contenido de los datasets en relación con la calidad de sus datos en el conjunto de entrenamiento y pruebas son sumamente relevantes para el rendimiento del aprendizaje y del conocimiento mismo que se extrae de ellos. Por otro lado, la cantidad de investigaciones que se hacen al respecto son muy importantes para el crecimiento del área, teniendo ciertas colaboraciones, ya sea compartiendo datasets como el IEMOCAP y futuros proyectos científicos. Como resultado, un ejemplo de un sistema de reconocimiento de afecto utilizando señales fisiológicas para niños con trastorno del espectro autista [56].

Es por ello que el trabajo realizado es el camino inicial para llevar la detección de emociones a la robótica social y obtener resultados preliminares ante los datos que se van obteniendo por medio de las distintas técnicas de Deep Learning y las modalidades que se van agregando al proyecto para mejorar el reconocimiento de emociones del ser humano y tomar las decisiones correspondientes ante esto.

3.4. Objetivos

3.4.1. Objetivo General

Evaluar distintos dataset para la detección de emociones mediante técnicas de Deep Learning y modalidades de rostro, audio y texto en base a una arquitectura ya implementada en el contexto de robótica social, tomando en cuenta las distintas características físicas de los robots.

3.4.2. Objetivos Específicos

- Instalar y adecuar la arquitectura de reconocimiento de emociones basada en Deep Learning para el reconocimiento multimodal.
- Seleccionar los diferentes dataset a probar mediante criterios de inclusión y exclusión para el entrenamiento de reconocimiento de emociones.
- Evaluar y comparar con soluciones similares el desempeño de los diferentes modelos a nivel unimodal y multimodal.

3.5. Metodología

La metodología de trabajo será basándose en la investigación cuantitativa, el cual consiste en cuantificar y analizar variables para obtener resultados. Implica la utilización y el análisis de datos numéricos utilizando técnicas estadísticas específicas para responder preguntas como quién, cuánto, qué, dónde, cuándo, cuántos y cómo. También describe los métodos para explicar un problema o fenómeno mediante la recopilación de datos en forma numérica. La investigación cuantitativa requiere la reducción de los fenómenos a valores numéricos para poder realizar el análisis estadístico [57].

Para iniciar una investigación siempre se necesita una idea para tener un primer acercamiento a la realidad objetiva, la cual proviene de una necesidad de resolver una problemática. Luego se pasa a la etapa del planteamiento del problema que es el centro de la investigación donde se define los métodos, se establece los objetivos de investigación y se desarrollan las preguntas de investigación. Lo siguiente es revisar la literatura, detectar la literatura pertinente, extraer y recopilar la información de interés y construir el marco teórico. Para dar el paso a visualizar que alcance tendrá la investigación para establecer sus límites conceptuales y metodológicos. Posteriormente, se elabora la hipótesis de la investigación y definir conceptualmente y operacionalmente sus variables. Para dar paso a definir y precisar el diseño específico para la investigación para luego seleccionar diferentes datasets apropiados para la investigación donde se definieron los métodos de selección y modelos a trabajar. En la próxima fase se prepararon y adecuaron los datasets a la arquitectura y se ejecuta el código para prepararlo para su análisis. En el análisis de los datos se explorarán los datos obtenidos en la recolección, interpretarlos mediante pruebas estadísticas, las hipótesis planteadas, comparar las métricas de evaluación por modalidad y preparar los resultados para presentarlos. Por último, se elabora el reporte o informe de resultados [1].

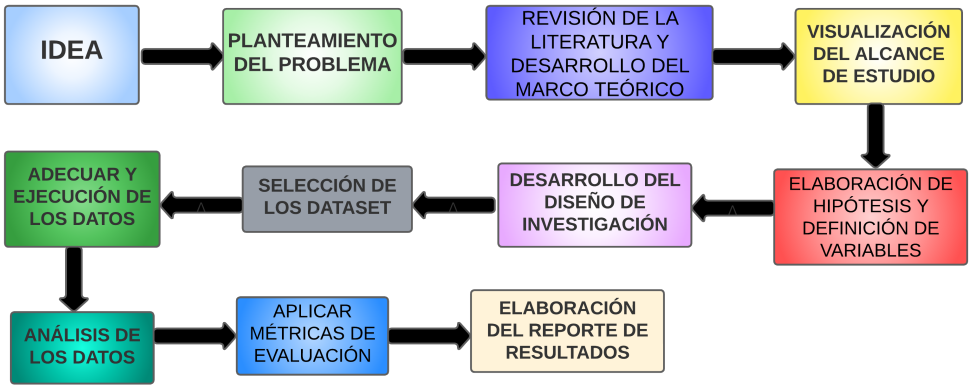


Figura 3.2: Proceso de investigación cuantitativa [1]

Capítulo 4

Materiales y Métodos

En este capítulo se van a mencionar los materiales que corresponden a todos los datasets a trabajar con su análisis descriptivo, mencionando la estructura, los criterios de inclusión y exclusión para escoger cada uno de ellos. Por otro lado, se tienen los métodos que abarcan las técnicas de análisis, explicando cada reconocimiento por modalidad detalladamente, como también los modelos a utilizar y la descripción de los procesos principales a lo largo del trabajo.

4.1. Materiales

4.1.1. Datasets y/o Casos de estudio

En la tabla 4.1 contiene los datasets que pueden o no ser seleccionados para realizar su preprocesamiento, procesamiento y sus evaluaciones unimodales y multimodales respectivas. Para realizar el proceso de investigación de datasets se van a tomar en cuenta los siguientes items que se muestran a continuación:

- **Nombre del dataset**: Indica el nombre del dataset a utilizar en el trabajo que se llevó a cabo. Es importante para poder investigar acerca de trabajos relacionados y obtener mayor información.
- **Autor**: Se indica la cita a utilizar donde se encuentra el dataset o el paper en formato IEEE. Es relevante para poder comunicarse vía correo electrónico con el fin de solicitar ciertos archivos o redactar cualquier imprevisto.
- **Emociones**: Se mencionan las distintas emociones que se encuentran dentro del dataset para el trabajo. Las emociones presentes dentro del dataset son muy relevantes a la hora de decidir cuál elegir, ya que se están buscando las mismas emociones que se trabajaron en IEMOCAP que son (angry, happy, disgust, neutral, sad, fear, surprise) con el fin de compararlas y obtener sus métricas.

- **Modalidad**: Se mencionan las distintas modalidades que se encuentren dentro del dataset a trabajar. Ayudará a informar si el dataset contiene la cantidad requerida a estudiar como son las 3 modalidades principales rostro, audio y texto que son las que se trabajaron en IEMOCAP.
- **N°de participantes**: Se indica la cifra exacta de la gente que participó en la construcción del dataset. Cabe destacar que ante la ausencia de información se va a utilizar el símbolo _, ya que no se pudieron obtener dichos datos. Es importante el número de participantes para obtener un estimado de las dimensiones de los conjuntos de train y test al momento de elegir un dataset.
- **Ambiente**: Se menciona el lugar en donde se trabaja si es bajo condiciones controladas o no. Es destacable al momento de seleccionar los datasets, ya que se da prioridad a los que se trabajaron en un ambiente no controlado por el hecho de que es lo más cercano a la realidad donde existe ruido y poca luminosidad.
- **Disponible**: Se indica si el dataset está disponible o no en forma pública en internet. Es fundamental para encontrar la data de una manera más rápida y evitar contactarse con el autor.

Nombre del dataset	Autor	Emociones	Modalidad	N°de Participantes	Ambiente	Disponible
Fer2013	[58] [59]	neutro, feliz, enojado, asustado, asqueado, triste, sorprendido	Rostro	70	Controlado	Si
Kdef	[60] [61]	enojado, disgusto, miedo, feliz, triste, sorpresa, neutro	Rostro	_	Controlado	Si
Mosei	[62]	felicidad, tristeza, neutral, enojado	Idioma, Vocal y audio.	1000	No Controlado	Si
Moseas	[62]	Alegria, tristeza, ira, miedo, asco y sorpresa	Idioma, vocal y audio	1645	No Controlado	Si
Heu Emotion	[63]	Ira, aburrido, confundido, decepcionado, disgusto, miedo, feliz, neutral, triste, sorpresa.	Rostro, habla y postura	8984 y 967	No Controlado	No
Meld	[64] y [65]	ira, disgusto, tristeza, alegría, neutro, sorpresa y miedo	Rostro, habla y texto	_	No Controlado	Si
Afew	[66]	Enojado, disgusto, miedo, feliz, triste, neutro y sorpresa	Rostro, habla y postura	330	No Controlado	No
Emotiw	[67]	enojado, asco, miedo,feliz, neutro, triste y sorpresa	Rostro, audio y texto	_	No Controlado	Si
Sfew	[68]	enojo, asco, miedo, felicidad, triste y sorpresa	Rostro, postura, audio	_	No Controlado	Si
Cheavd	[69]	Enojado, feliz, triste, neutro, sorpresa y disgusto	Rostro y audio	238	No Controlado	No
The Enterface05	[70]	ira, asco, miedo, felicidad, tristeza, sorpresa, neutro.	Rostro y habla.	42	Controlado	No
Aff-Wild2	[71] y [72]	ira, disgusto, miedo, felicidad, tristeza, sorpresa, neutral y otros.	Rostro	460	No Controlado	Si

Tabla 4.1: Datasets a investigar

4.1.2. Casos de estudio

Se tomaron en cuenta diversos dataset a investigar, los cuales se analizaron previamente para llevarlo a proceso de selección de los casos de estudio, en donde cada uno de ellos se le aplicó un criterio de evaluación con respecto a la falta de modalidades, emociones, la disponibilidad del dataset y el ambiente en el cual se trabaja. Al verificar la mayoría

de estos puntos se puede llevar a cabo el preprocesamiento del conjunto de datos que conlleva la limpieza de datos, el balanceo de los datos, realizar el análisis descriptivo y adecuar la entrada de datos a la arquitectura ya implementada que se desea evaluar.

Afew: Es un conjunto de datos en donde participan 330 personas en un ambiente no controlado con 1809 videos de diferentes películas. En las cuales existen diferentes modalidades como facial, audio y postura. Estos datos forman parte del reconocimiento emotion in the Wild challenge 2018. Las carpetas contienen subcarpetas específicas de la emoción donde están etiquetadas las 7 emociones diferentes como son: Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise. La etiqueta del archivo .avi es su nombre de carpeta correspondiente donde existen videos de corta duración de ciertas películas [66] [73].

Sfew: Es un conjunto de datos para el reconocimiento de expresiones faciales. Se creó seleccionando fotogramas estáticos de la base de datos AFEW mediante el cálculo de fotogramas clave basados en la agrupación de puntos faciales. Se ha dividido en tres conjuntos: Train (958 muestras), Val (436 muestras) y Test (372 muestras). Cada una de las imágenes se asigna a una de las siete categorías de expresión, es decir, ira, disgusto, miedo, neutralidad, felicidad, tristeza y sorpresa [74].

Dataset Meld: El conjunto de datos multimodal de EmotionLines (MELD) se creó mejorando y ampliando el conjunto de datos de EmotionLines. MELD contiene las mismas instancias de diálogo disponibles en EmotionLines, pero también abarca la modalidad de audio y visual junto con el texto. MELD tiene más de 1400 diálogos y 13000 expresiones de la serie Friends TV. Múltiples oradores participaron en los diálogos. Cada expresión en un diálogo ha sido etiquetada por cualquiera de estas siete emociones: ira, disgusto, tristeza, alegría, neutralidad, sorpresa y miedo. MELD también tiene una anotación de opinión (positiva, negativa y neutral) para cada expresión. Se ha desarrollado bajo un ambiente no controlado [64].

4.1.3. Análisis descriptivo

El dataset **Iemocap** es trabajado en un ambiente controlado, el cual es unos de los principales puntos de referencia para el reconocimiento de emociones multimodales que está disponible sin problemas. Está compuesto varios tipos de datos, como información de rostros, de captura de movimiento, voz, videos y transcripciones de diálogos. Los datos disponibles de Iemocap son 7532 muestras en donde se han anotado 10 categorías de emociones, pero el número de muestras está desequilibrada. Es importante recalcar que de las 10 emociones solo se consideraron la felicidad, neutralidad, tristeza e ira para todo el proceso de entrenamiento y evaluación, con el fin de comparar las mismas emociones estándar con los otros dataset a utilizar.

Lo siguiente que se realizó es un análisis descriptivo en este dataset para tener una idea de la organización de carpetas y de que está compuesta a rasgos generales este conjunto

de datos. En la figura 4.1 los primeros 4 archivos csv son para el conjunto de entrenamiento y el último archivo es para las pruebas.

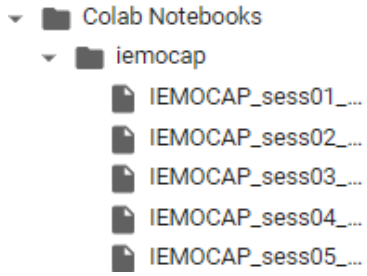


Figura 4.1: Archivos csv a utilizar en Iemocap

Algunos dataset contienen datos categóricos como clases y otros se expresa la emoción en forma cuantitativa en bases a la valencia, activación y dominancia.

En cada archivo .csv dado a conocer en la figura 4.2 se encuentra una columna start_time y end_time que representa el tiempo de los diferentes frames extraídos de un cierto archivo, como también existe una columna llamada file_name que da a conocer el nombre como tal del archivo trabajado, lo siguiente es la columna de emotion que es donde se alojan las emociones de la data con su respectiva etiqueta y los valores de val (valencia), act (activación) y dom (dominancia) que corresponden al comportamiento y a las características para distinguir entre las diferentes emociones.

Donde la valencia es la dimensión principal sobre la cual se construye la experiencia emocional, es el componente motivacional de la emoción. La activación es el proceso psicológico que se inicia cuando en el entorno se producen cambios no deseados o estresantes. La dominancia hace referencia al grado de control que la persona percibe sobre su respuesta emocional.

	start_time	end_time	file_name	emotion	val	act	dom
0	6.2100	9.3200	Ses01F_script01_1_F000	fru	2.0000	2.3333	2.3333
1	9.3500	12.8955	Ses01F_script01_1_F001	xxx	2.0000	2.3333	1.6667
2	14.3063	19.5526	Ses01F_script01_1_F002	sur	2.3333	2.3333	2.6667
3	22.3200	24.6667	Ses01F_script01_1_F003	xxx	3.0000	3.0000	2.6667
4	35.3799	39.0900	Ses01F_script01_1_F004	xxx	2.0000	2.0000	2.0000
...	...	...	...	...	...	...	...
10034	258.3600	260.1200	Ses05F_impro03_M064	hap	4.0000	3.0000	3.0000
10035	260.1500	263.9800	Ses05F_impro03_M065	hap	4.5000	4.5000	4.5000
10036	264.0000	265.5500	Ses05F_impro03_M066	hap	4.0000	3.5000	3.5000
10037	267.0700	269.2300	Ses05F_impro03_M067	hap	4.0000	3.0000	3.5000
10038	269.2700	271.5900	Ses05F_impro03_M068	hap	4.0000	3.5000	4.0000

10039 rows × 7 columns

Figura 4.2: Contenido de los csv en Iemocap

Lo importante también es saber la cantidad de emociones a analizar dentro del dataset que están categorizadas cada una de ellas para facilitar el preprocesamiento de los datos dado a conocer en la figura 4.3.

```
p['emotion'].value_counts()
```

```
xxx    2507
fru    1849
neu    1708
ang    1103
sad    1084
exc    1041
hap     595
sur     107
fea      40
oth       3
dis       2
Name: emotion, dtype: int64
```

Figura 4.3: Cantidad de emociones en Iemocap

El dataset **Afew** se caracteriza por ser capturado en un ambiente no controlado (salvaje) donde la calidad de la imagen, ruido y la luz afectarían a los videos asociados. A lo cual se desarrolló con 330 personas y con modalidades como la postura, rostro y audio. Por otro lado, cada emoción registrada se etiquetó en carpetas, por lo que los videos de las películas se fueron clasificando cada una de ellas y guardando en las carpetas asociadas. El contenido de ellas está compuesta por varios videos recortados por frame de diferentes películas donde aparecen distintos actores realizando la misma emoción detectada pero en distintas situaciones.

En la siguiente figura 4.4 se presenta la cantidad de emociones que están categorizadas por cada video dentro del dataset donde se consideraran las emociones estándar como son angry, happy, neutral y sad. De otra manera, para visualizar estos datos se realizó un gráfico de barras que permite conocer el nivel de complejidad para llevar a cabo el preprocesamiento y la limpieza de datos basándose en la cantidad de archivos a trabajar.

```
emotion:  Angry
number of audios:  197
-----
emotion:  Disgust
number of audios:  113
-----
emotion:  Fear
number of audios:  127
-----
emotion:  Happy
number of audios:  213
-----
emotion:  Neutral
number of audios:  207
-----
emotion:  Sad
number of audios:  177
-----
emotion:  Surprise
number of audios:  120
```

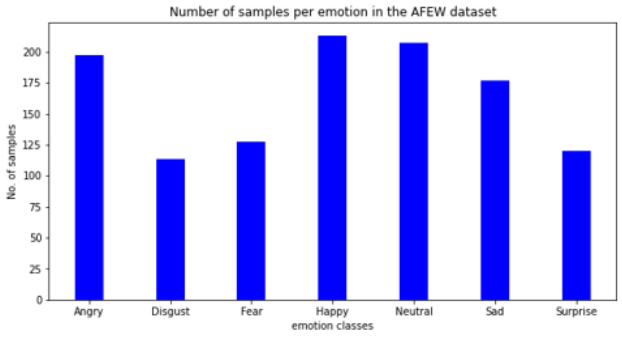


Figura 4.4: Gráfico emotion vs samples en Afew

Como se mencionó anteriormente, cada carpeta etiquetada por emoción guarda diferentes videos cortados por frame a lo cual se aprecia diferentes escenas con mejor calidad de imagen que otras al trabajar en un ambiente no controlado donde existe problemas con la calidad de la imagen. En la siguiente figura4.5 se detectan 2 rostros con una emoción enojado y disgustado.

Figura 4.5: Registro de emoción enojado y disgustado en Afew

El dataset **Sfew** se divide en 3 conjuntos como son el train,val,test los cuales cada uno de ellos tiene diversas muestras de expresiones faciales que provienen de cierta parte del dataset afew. La carpeta de entrenamiento contiene las 7 emociones etiquetadas comprimidas independientemente cada una de ellas.

En primera instancia se muestra en detalle el contenido del zip de la emoción angry y el formato de cada frames categorizadas por cada emoción, las cuales se presentan en el anexo A.1. Algunos ejemplos de las imágenes que se encuentran en la etiqueta de la emoción angry, ciertos actores no se aprecia la expresión facial de la mejor manera, ya que se trabaja en un ambiente real, a lo cual ciertas imágenes no corresponden totalmente a la emoción detectada. Lo mencionado anteriormente se refleja en la figura 4.6.

Figura 4.6: Imágenes de Angry en Sfew

Para tener una mejor representación de las diferentes imágenes y sus emociones que se alojan dentro del dataset Sfew se muestra en la figura 4.7 en forma de grilla para abarcar lo más posible el contenido de cada carpeta.

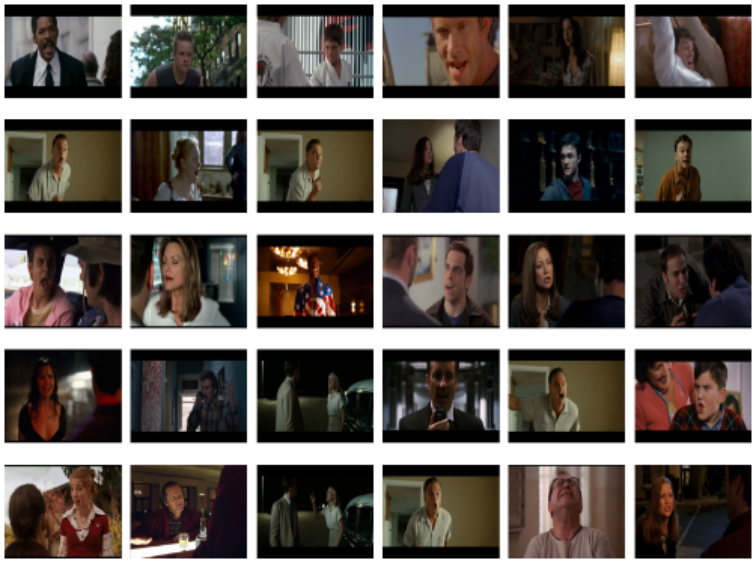


Figura 4.7: Frames originales del conjunto de datos Sfew

Lo siguiente fue recorrer la carpeta train y las subcarpetas etiquetadas en donde se obtendrán las cantidades de ejemplos de las 7 emociones del dataset y para tener una representación más formal se generó un gráfico de barras para visualizar los datos con respecto a las clases (emociones) que contiene el dataset originalmente y sus cantidades en el conjunto de train como se muestra en la figura 4.8 en donde existe un desbalanceo considerable con respecto a las cantidades de disgust, fear y surprise en comparación a las otras emociones estándar que se van a utilizar dentro del entrenamiento.

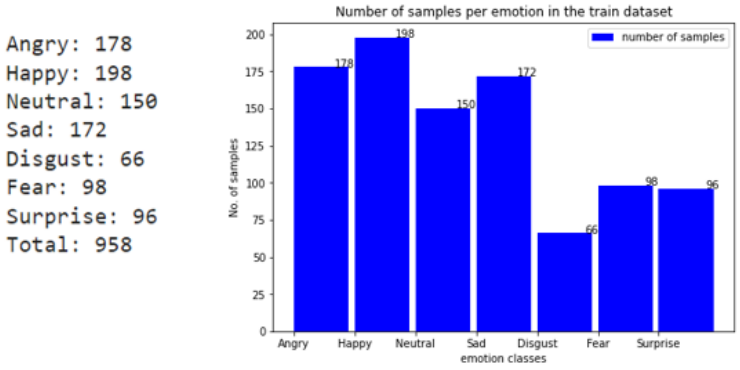


Figura 4.8: Gráfico de emociones del conjunto train en Sfew

El mismo procedimiento se llevó a cabo para el conjunto de test en la cual se puede apreciar en la figura 4.9 donde existe menos cantidad de ejemplares por cada emoción

a diferencia del conjunto de train y sigue existiendo un desbalanceo con respecto a las emociones disgust, fear y surprise que no se tomarán en cuenta para todo el proceso del entrenamiento con la finalidad de seguir el estándar de emociones a evaluar dentro del trabajo de investigación por cada dataset.

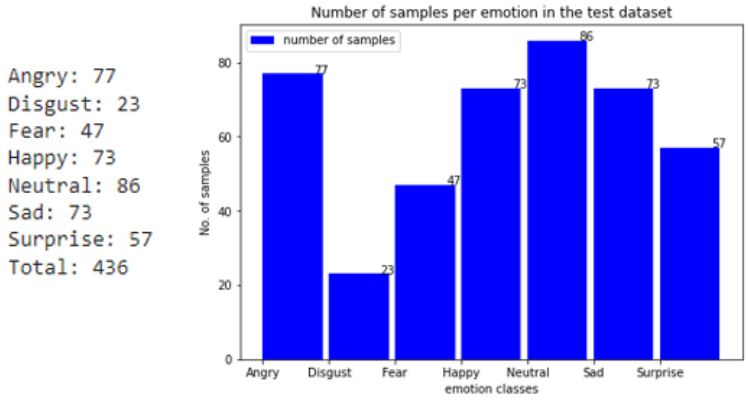


Figura 4.9: Visualización de los ejemplares del conjunto test en Sfew

Se trabaja con el dataset **Meld** para obtener el contenido de su conjunto de datos de entrenamiento y pruebas con el fin de tener una mayor claridad de cómo está estructurado el conjunto de datos. Cada uno de ellos contiene un csv particular como se muestra en la figura 4.10 con diferentes columnas que representan información importante dentro del dataset que se detalla en la tabla 4.2.

	Sr No.	Utterance	Speaker	Emotion	Sentiment	Dialogue_ID	Utterance_ID	Season	Episode	StartTime	EndTime
0	1	also I was the point person on my company's tr...	Chandler	neutral	neutral	0	0	8	21	00:16:16,059	00:16:21,731
1	2	You must've had your hands full.	The Interviewer	neutral	neutral	0	1	8	21	00:16:21,940	00:16:23,442
2	3	That I did. That I did.	Chandler	neutral	neutral	0	2	8	21	00:16:23,442	00:16:26,389
3	4	So let's talk a little bit about your duties.	The Interviewer	neutral	neutral	0	3	8	21	00:16:26,820	00:16:29,572
4	5	My duties? All right.	Chandler	surprise	positive	0	4	8	21	00:16:34,452	00:16:40,917
...	...	...	...	...	...	...	...	...	...	...	...
9984	10474	You or me?	Chandler	neutral	neutral	1038	13	2	3	00:00:48,173	00:00:50,799
9985	10475	I got it. Uh, Joey, women don't have Adam's ap...	Ross	neutral	neutral	1038	14	2	3	00:00:51,009	00:00:53,594
9986	10476	You guys are messing with me, right?	Joey	surprise	positive	1038	15	2	3	00:01:00,518	00:01:03,520
9987	10477	Yeah.	All	neutral	neutral	1038	16	2	3	00:01:05,398	00:01:07,274
9988	10478	That was a good one. For a second there, I was...	Joey	joy	positive	1038	17	2	3	00:01:08,401	00:01:12,071

9989 rows × 11 columns

Figura 4.10: Contenido del conjunto de entrenamiento de Meld

Nombre de la columna	Descripción
Sr.No.	Numeros de serie de los enunciados principalmente para hacer referencia a los enunciados en caso de diferentes versiones.
Utterance	Expresiones individuales de una persona en forma de cadena.
Speaker	Nombre del hablante asociado con el enunciado
Emotion	La emoción (neutral, joy, sadness, anger, surprise, fear, disgust) expresada por el hablante en el texto.
Sentiment	El sentimiento (positivo,neutral y negativo) expresado por el hablante en el texto.
Dialogue_ID	El índice del dialogo a partir de 0
Utterance_ID	El índice de la expresion particular en el dialogo a partir de 0
Season	El numero de la temporada de Friends Tv al que pertenece una expresión particular.
Episode	El numero de episodio de Friends Tv en una temporada en particular a la que pertenece el enunciado.
StartTime	La hora de inicio del enunciado en el episodio dado en el formato 'hh:mm:ss, ms'
EndTime	La hora de finalización del enunciado en el episodio dado en el formato 'hh:mm:ss, ms'.

Tabla 4.2: Tabla con el detalle de cada columna del dataset de Meld

En la siguiente figura 4.11 se dan a conocer la cantidad de emociones que existen dentro del conjunto de datos donde predomina la emoción neutral por mayoría y son datos que no se encuentran balanceados, es decir, la diferencia entre cantidades de videos por emoción es amplia. Para representarlo de otra manera se realizó una gráfica que muestra la cantidad de emociones junto con la cantidad de sentimientos por cada una de ellas a partir de los videos originales del dataset. Cabe destacar que las emociones a trabajar son las estándar anger, neutral, sadness y happy, pero en este dataset en particular al ausentarse la emoción happy se considerara como joy y surprise por sad debido a problemas con los videos originales y por la poca cantidad de ejemplares que existen.

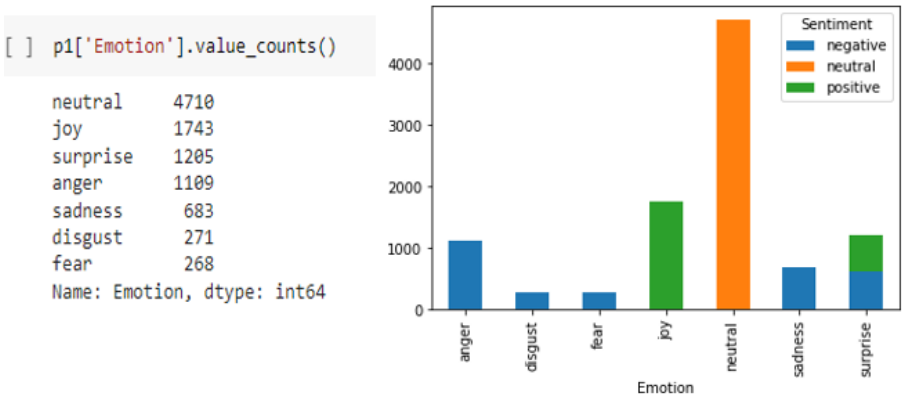


Figura 4.11: Gráfico de Emociones y Sentimientos de Meld

La siguiente figura 4.12 muestra el frames de un video en específico donde se aprecian muchas personas realizando diversas emociones, las cuales no están recortadas los rostros por hablante y tampoco identificados.

Figura 4.12: Frame de un video en específico de Meld

4.1.4. Criterios de inclusión y exclusión de datasets

Es recomendable tener muy claro los criterios de inclusión y exclusión para poder escoger dataset de la mejor manera posible como también evitar su elección. Para ello se presentan en la siguiente tabla 4.3:

CRITERIO DE INCLUSIÓN	CRITERIO DE EXCLUSIÓN
-Los dataset se encuentran muy bien balanceados.	**-El tener poca información al respecto del dataset ya sea del preprocesamiento facial, audio y texto.**
-Se encuentre toda la información posible para el preprocesamiento de los datos.	**-No tener ejemplares dentro de internet para tener una referencia.**
-Que esten disponibles en internet para poder comunicarse con los autores ante cualquier imprevisto.	**-No poder conseguir respuesta de los autores ante la ausencia de algunos archivos.**
-Se encuentren desarrollados en un ambiente no controlado.	**-Se encuentren desarrollados en un ambiente controlado.**
-Contenga al menos las emociones estandar que son angry, neutral, sad y happy.	**-No contenga las emociones que se requieren.**

Tabla 4.3: Criterio de Inclusión y Exclusión

4.1.5. Dataset excluidos

Se dejan excluidos por diversos temas que se explicarán uno por uno a continuación y por consecuencia no serán considerados en el estudio:

Tenemos por ejemplo el dataset **the enterface05** que se maneja en un ambiente controlado y, por otro lado, los datos procesados no han sido publicados. A lo cual se solicitó al autor vía correo electrónico para obtener los datos requeridos, pero no hubo respuesta.

Como también el dataset **heu-emotion y cheavd** son los dataset más recomendables a utilizar por la cantidad de modalidades, ambiente en el cual se trabaja, pero el único problema es que no son dataset públicos lo cual se solicitó un correo a los autores de dichos trabajos para lograr tener los datos procesados, pero surgió el mismo problema mencionado anteriormente que el autor no respondió la solicitud vía correo.

El conjunto de datos **engagement wild** fue solicitado vía correo electrónico con destino a los autores de dicho dataset, donde el gran problema que se encontró fue que el conjunto de videos no contemplaban audios de los participantes, simplemente eran videos de postura y gestos al momento de reaccionar a varias escenas de distintas películas. Por otro lado, no contenían información relevante para facilitar el preprocesamiento de las modalidades.

4.1.6. Interpretación de resultados

El dataset Iemocap se puede apreciar que la información relevante las contiene en diferentes csv con las emociones categorizadas y con los frames respectivo de cada video. A los cuales 4 csv son para el conjunto de train y el último para el conjunto de pruebas. Donde los datos no se encuentran balanceados en primera instancia y los gráficos para representar las emociones que contiene el dataset se puede concluir que a partir de los valores act,val y dom están muy bien agrupados y representados. Donde siempre existen algunas excepciones que están un poco más excluidas, que tienen características similares a otras emociones o tienen un comportamiento parecido como puede ser la emoción de felicidad y la excitación. Los videos originales del dataset ya vienen recortados los rostros de los hablantes de cada escena y un identificador, lo cual simplifica el trabajo a realizar.

Para el dataset Sfew es simplemente una base de datos de imágenes que está relacionada con el dataset Afew donde las imágenes están categorizadas por emociones y están divididas por frames. Por lo que contiene solo una modalidad y no existen videos en el dataset original.

El dataset Afew es desarrollado en un ambiente no controlado donde sus videos son recopilaciones de varias películas que tienen diferentes calidades de imagen. El conjunto de train y test contienen una gran cantidad de videos que están categorizadas cada una de ellas donde se consideraran las emociones estándar, pero existe un problema que se repiten una gran cantidad de videos en los dos conjuntos, a lo que se realiza una limpieza posteriormente. La idea es trabajar con los videos y sacar los audios para luego transcribirlo a texto.

El dataset Meld está organizado de una forma distinta donde la mayoría de la información se encuentra dentro de un csv para el conjunto de entrenamiento y otro csv para el conjunto de pruebas que presentan los textos de los videos originales con su hablante y las emociones que está representando en cierto frame del video. Estos videos son desarrollados en un ambiente no controlado, dado que son extraídos de una serie de TV, donde a diferencia de los demás dataset se aprecian muchas personas representando diversas emociones en los videos a los cuales no están recortadas los rostros y tampoco identificados los que están realizando tal emoción. Por otro lado, se trató de comunicar con el autor donde no hubo respuestas y no existen trabajos relacionados con la modalidad facial donde se complica más aún el preprocesamiento de dicha modalidad. La cantidad de emociones dentro

de este dataset están desbalanceado por el lado de la emoción neutral, a lo cual se realizará una limpieza de datos posteriormente, como también algunos formatos diferentes en los nombres de archivos en el conjunto de pruebas.

4.2. Métodos

4.2.1. Técnicas de análisis

En esta sección se van a describir las arquitecturas de Deep Learning utilizadas en cada modalidad para realizar el reconocimiento de emociones [3].

4.2.1.1. Reconocimiento facial

Se usó una arquitectura VGG19 debido a su profundidad con 19 capas, lo que le permite aprender representaciones complejas de las imágenes y para capturar características sutiles en las expresiones faciales que son importantes para el reconocimiento de emociones. Por otro lado, las imágenes se reduce a 48×48, que también reducen el número de características extraídas de las capas convolucionales; por lo tanto, las capas lineales al final de la red también tienen menos parámetros. Este modelo VGG19 tiene cinco bloques compuestos por capas convolucionales y capas de agrupación máxima, como se muestra en la Figura 4.13. Esos bloques convolucionales tienen la siguiente configuración: (i) dos convoluciones de 64 filtros; (ii) dos convoluciones de 128 filtros; (iii) cuatro convoluciones de 256 filtros; (iv) cuatro convoluciones de 512 filtros; y (v) cuatro convoluciones de 512 filtros. Después de los bloques de convolución, se aplica una capa de agrupación promedio, lo que da como resultado un vector de 512 características después de aplanar los resultados de las convoluciones. Finalmente, una capa lineal procesa las 512 características para obtener la predicción final [3].

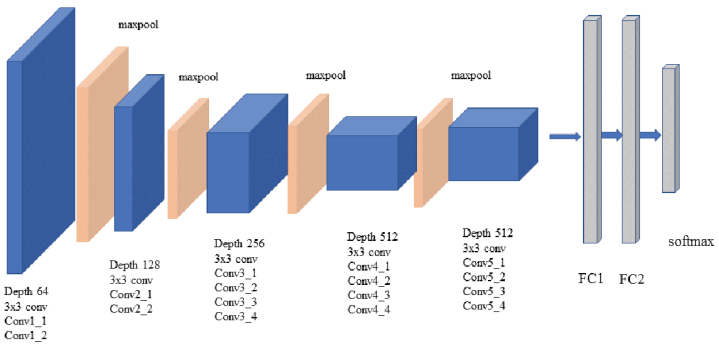


Figura 4.13: VGG19 arquitectura definida por Roland Hewage [2].

4.2.1.2. Reconocimiento de audio

Para la detección de emociones a partir del habla, se aplica un modelo convolucional utilizado por Venkataramanan y Rajamohan [75] como se muestra en la figura 4.14.

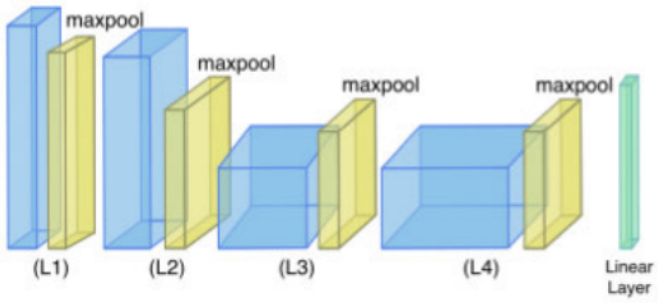


Figura 4.14: Arquitectura modelo de audio definida por Venkataramanan y Rajamohan [3]

Para este modelo, los audios de voz se procesaron como MFCC como se muestra en la figura 4.15. MFCC es un método de extracción de características de los coeficientes cepstrales de frecuencia mel en donde se enfoca en la modalidad del habla, lo cual permite extraer el contraste de frecuencia vocal, lo cual es relevante para la detección de emociones [76] y se utilizan como datos 2D; y como son vectores 2D, podrían ser procesados como imágenes para redimensionarlos y uniformar los datos de entrada [3]. Los espectrogramas mel capturan características relevantes del sonido que son perceptualmente significativas para los humanos. La escala mel se adapta mejor a cómo percibimos las diferencias de frecuencia en el oído humano, lo que permite capturar características importantes de manera más efectiva. También tienden a enfocarse en elementos del sonido que son relevantes para la expresión emocional, como cambios en la frecuencia y la energía. Esto hace que sean efectivos para identificar patrones relacionados con las emociones.

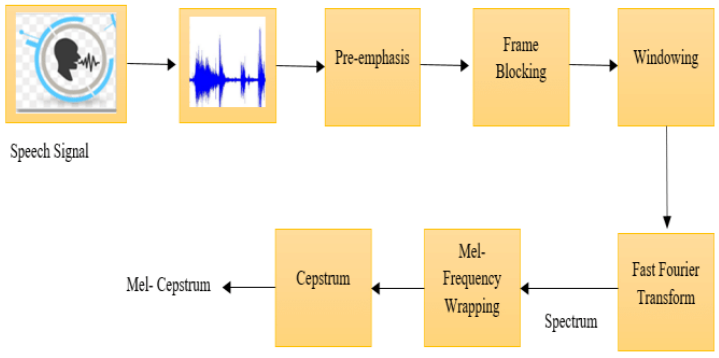


Figura 4.15: MFCC arquitectura definida por J. Kharibam y A. Devi [4]

Este tamaño de entrada se establece en 128×259, lo que significa 128 funciones de audio y 259 unidades de tiempo. La arquitectura del modelo tiene 4 bloques, donde cada uno tiene una capa de convolución, una normalización por lotes, una función de activación, una capa de agrupación máxima y una capa de abandono. Las capas de convolución están configuradas como L1:128, L2:128, L3:64, L4:64; las capas de agrupación como L1:2×2; y los demás como L2, L3, L4: 4×4. Las capas de abandono se establecen en 0,5 para el primer bloque (L1) y 0,25 para los demás (L2, L3, L4). Finalmente, la función de activación utilizada es la función Unidad Lineal Exponencial (ELU), que permite algunos valores negativos y ayuda al proceso de aprendizaje; sin embargo, el tiempo de ejecución podría aumentar debido a la operación exponencial agregada. Sin embargo, durante el entrenamiento del modelo, ELU permite una convergencia más rápida que ReLU.

Para la arquitectura ya implementada el habla se transcribe a texto como se muestra en la figura 4.16 en donde se ocupa esta técnica que consiste en el procesamiento del lenguaje natural, es el campo de la Inteligencia artificial que se ocupa de como las computadoras analizan, comprende e interpretan el lenguaje humano.

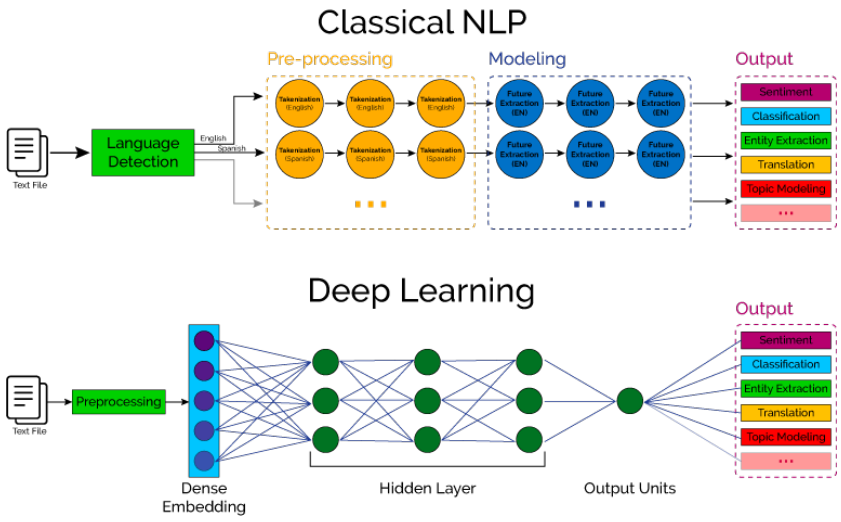


Figura 4.16: NLP arquitectura definida por Matthewbdineen [5]

4.2.1.3. Reconocimiento de texto

Para esta modalidad, usamos una implementación PyTorch de DialogXL [77], que es un modelo especializado para el Reconocimiento de Emociones en Conversación (ERC) basado en XLNet [78]. La versión original de XLNet es un modelo que ha alcanzado resultados de vanguardia en muchas aplicaciones. Consiste en un modelo de lenguaje autorregresivo basado en la arquitectura del transformador que, dada una secuencia, genera la probabilidad de la secuencia de palabras a seguir. DialogXL mejora XLNet mediante

el uso de una memoria mejorada para almacenar un contexto histórico más largo durante el diálogo y la autoatención de Dialog-Aware para realizar un seguimiento de los diferentes oradores en una conversación. Cada oración de un hablante (expresión) pasa a través de una capa de incrustación que convierte la oración en una secuencia de vectores como se muestra en la Figura 4.17. Luego, esta representación pasa a través de una pila de redes neuronales, donde cada capa contiene un componente de autoatención consciente del diálogo y un componente de recurrencia del enunciado; cada una de estas capas genera un vector que se pasa a la siguiente capa. Al final de la última capa, el estado oculto del token de clasificación y el contexto histórico pasan a través de una red neuronal de avance para obtener la emoción predicha [3].

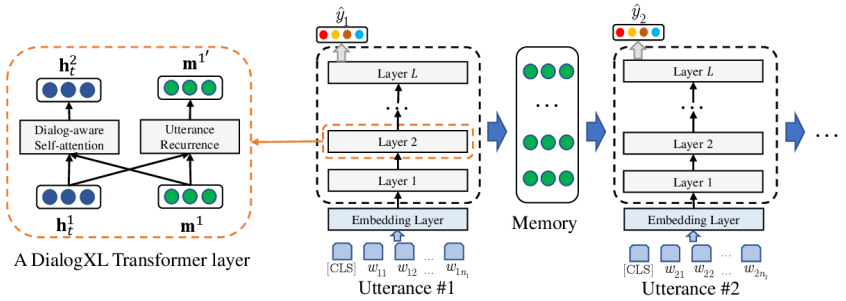


Figura 4.17: Dialogxl arquitectura definida por Weizhou Shen y Junqing Chen [6]

4.2.2. Metodo de Fusión

Se utilizó el método EmbraceNet+ para fusionar las modalidades y generar la predicción final[7]. EmbraceNet+ es una extensión de EmbraceNet que mejora algunos aspectos, y se compone de tres EmbraceNets y dos métodos de fusión adicionales. Un EmbraceNet básico tiene capas de acoplamiento que ajustan las salidas de modalidad al mismo tamaño mientras aprende sobre la correlación entre ellos; y una capa que selecciona características de las salidas de modalidades siguiendo la distribución multinomial. A diferencia de EmbraceNet+, dos EmbraceNet básicos tienen modificaciones en las capas de acoplamiento al agregar una capa lineal y una capa de abandono en cada capa de acoplamiento. Hay tantas capas de acoplamiento como modalidades y solo una capa de abrazo en la EmbraceNet básica, aunque la capa de abrazo necesita tantas probabilidades y valores de disponibilidad como modalidades para funcionar correctamente. En el sistema propuesto, los modelos individuales utilizados solo dan la clasificación final, no los vectores de características intermedias; por lo tanto, en el método de fusión EmbraceNet+, se elimina la EmbraceNet inicial, que procesa los datos intermedios, como se muestra en la figura 4.18. Cada capa de acoplamiento alterada está formada por una capa lineal de 32 neuronas, una capa de abandono con 0,5 de probabilidad de decaimiento y otra capa lineal de 16 neuronas. Como métodos de fusión adicionales, se usó la suma ponderada, cuya salida es un vector de n

probabilidades (n=número de categorías de emociones), y una concatenación, cuya salida es un vector de 3n por la cantidad de modalidades. Así, la otra EmbraceNet recibe tres vectores de 16n, y 3n valores (que funcionan como modalidades), y se manejan acoplando capas de una capa lineal de 16 neuronas cada una, lo que lleva a añadir una capa extra lineal de neuronas, que genera la predicción final. EmbraceNet y EmbraceNet+ toleran la falta de datos al multiplicar las probabilidades de la distribución multinomial con un vector binario de disponibilidad. Este vector de disponibilidad contiene 1 si los datos de la modalidad respectiva son correctos, de lo contrario, contiene 0. Por ejemplo, si los datos de la cara no son claros o son erróneos, el vector de disponibilidad será [0,1,1] y las probabilidades [0.0,0.5,0.5]; asumiendo que el vector representa [cara, audio, texto]. Esas probabilidades se utilizan para seleccionar características siguiendo la distribución multinomial.

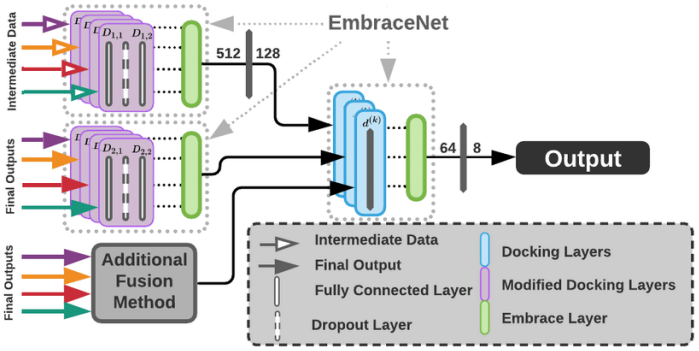


Figura 4.18: Embracenet+ arquitectura definida por JuanPablo Heredia [7]

Se eligió el conjunto de datos Iemocap para entrenar y probar los métodos individuales y de fusión. Los tipos de datos son de información de rostro, voz, videos, información de movimiento, ángulo de la cabeza y transcripciones de diálogos [3]. Para llevar a cabo toda la tarea de preprocesamiento, cada cuadro se cortó para contener solo a la persona anotada y descartar los datos de ruido. El iemocap se usa mucho para entrenar y probar modelos de reconocimiento de emociones, para la detección de rostros se utilizó el modelo retinaface [79] por su precisión y facilidad de uso. El cual es un detector de rostro robusto de una sola etapa que realiza la localización de rostros basada en pixeles.

Al cortar los videos en los subvideos anotados, se pierden varios segmentos porque las herramientas utilizadas no funcionaban correctamente, provocando que la modalidad presencial tenga mucho menos número de muestras para entrenamiento y pruebas. Esto también provoca que no todas las muestras tengan las tres modalidades, pero lo cierto es que todas tienen al menos una. Sin embargo, este hecho permite simular pérdida de datos por fusión de modalidad, lo que ayuda en la robustez del sistema para su aplicación con datos de ambientes no controlados [3].

4.2.3. Formación de modelos

Iemocap tiene sus datos distribuidos en 5 sesiones, de manera que los datos de las primeras 4 sesiones se utilizaron para los modelos de entrenamiento, quedando la quinta sesión sólo para pruebas. Para la modalidad facial hay 19384 imágenes, obtenidas después del preprocesamiento (2423 muestras × 8 fotogramas cada uno), de los cuales el 79 % fueron para capacitación, y el resto (21.21 %) para pruebas. Para la modalidad de audio se cuenta con 5531 muestras, donde el 56 % se utilizó para entrenamiento y el 22.44 % para pruebas [3].

El modelo de texto DialogXL, que se usó para analizar texto, está preentrenado en el conjunto de datos de entrenamiento Iemocap, al igual que los otros modelos y usa 6 emociones (felicidad, neutralidad, tristeza, ira, emoción y miedo). Durante el entrenamiento original, se usó el optimizador AdamW y los hiperparámetros ajustables fueron la tasa de aprendizaje, la cantidad de cabezales para los cuatro tipos de atenciones utilizadas en el componente de autoatención consciente del diálogo, la longitud máxima de la memoria y la tasa de abandono. Pero al trabajar con solo cuatro emociones, fue necesario adaptar el modelo de texto. Por lo tanto, se fusionó la felicidad con la excitación para equilibrar la cantidad de muestras en el conjunto de datos y las predicciones relacionadas con el miedo se eliminaron para el entrenamiento y para una mayor fusión. Así, la modalidad de texto cuenta con 5188 muestras, de las cuales el 76 % se utilizó para entrenamiento y el 24 % se utilizó para pruebas [3].

Los modelos faciales y de audio fueron entrenados desde cero con las cuatro emociones consideradas. Para el entrenamiento, ambos modelos usan la función de pérdida de entropía cruzada y el optimizador de Adam con una tasa de aprendizaje base establecida en 0.001, pero para el modelo de audio también se utilizó un planificador para reducir la tasa de aprendizaje hasta 0.000001. El modelo de rostro presentó sobreajuste, expuesto por el aumento de los valores de pérdida de validación; por lo tanto, el entrenamiento se detuvo en la época 22 para evitar la propagación de errores. El modelo de audio fue entrenado en 60 épocas. La gran ventaja en este caso fue el bajo nivel de procesamiento al usar una red neuronal con solo 3 capas de convolución y los datos se convirtieron de audio a imágenes con preprocesamiento [3].

Finalmente, se entrenó el método de fusión EmbraceNet+ también con el optimizador Adam y utilizando la función de pérdida de entropía cruzada. Este componente toma todas las muestras disponibles de tres modalidades y las muestras no coincidentes se tomaron como incompletas, colocando la modalidad ausente. Siguiendo el número de emociones y los parámetros requeridos, el EmbraceNet interno que emite la predicción final que fue configurada como [16,4,12] para la primera EmbraceNet modificada, suma ponderada y concatenación, respectivamente, y una capa lineal con cuatro neuronas.

4.2.4. Descripción de procesos

- Recolección: Se realizó un criterio de selección de los dataset a evaluar donde primero se tiene que considerar que el dataset esté balanceado, la cantidad de modalidades existentes, la calidad de los datos y el ambiente de trabajo en el cual se llevó a cabo para posteriormente llevarlo a la arquitectura implementada y adecuarlo a ella [18].

- Almacenamiento: Todos los datos que provienen de los diferentes dataset se van a ir adecuando a la arquitectura ya implementada con técnicas Deep Learning dentro de una red neuronal, las cuales se van a procesar las distintas modalidades de una forma independiente para luego fusionarlo con el método Embracenet+ y aplicar ciertas métricas de evaluación en el siguiente proceso [18].

- Análisis: Consiste en seleccionar las métricas de evaluación que son el F1 score y la exactitud en donde se llevarán a cabo evaluaciones para las tres modalidades individuales (audio, texto y rostro) y todo el sistema compuesto se evalúan por separado donde las emociones a analizar se aplicarán sus métricas individualmente [17]. Posteriormente, para tener un enfoque multimodal se lleva a cabo 4 experimentos que consisten en la evaluación de todas las modalidades, evaluación de cara y texto, evaluación de audio y texto y evaluación de cara y audio [18].

- Visualización: Aquí se representará los datos de forma gráfica y tablas comparativas para proporcionar una manera accesible de ver y comprender de una forma más sencilla los datos importantes. Donde se van a haber reflejado en un gráfico de comparaciones de las métricas de evaluación y un gráfico de comparación de las precisiones (exactitud) reportadas con diferentes modalidades. En donde se llevará a cabo en los distintos dataset a trabajar [18].

Capítulo 5

Experimentación

En este capítulo se presenta el diseño de la etapa de experimentación de forma detallada, se van a explicar las condiciones de experimentación que abarca los software utilizados, herramientas de almacenamiento y algoritmos existentes. Por último se describe el hardware a utilizar en donde se encuentra la máquina local y remota.

5.1. Diseño de Experimentos

En este trabajo de investigación se establecen las siguientes preguntas:

- ¿Cuáles son los dataset que se adaptan mejor a la arquitectura ya existente según sus resultados?
- ¿Se pueden incorporar mejoras a los dataset y a la arquitectura?
- ¿Cuál dataset cumple de mejor manera los requisitos establecidos?
- ¿Cuál es la diferencia relevante que existe entre los distintos dataset?
- ¿Qué emoción es la más destacada por los distintos casos a trabajar?

Para contestar a las dos primeras preguntas se realizó primero que nada un análisis descriptivo de cada dataset para verificar el contenido de estos y aplicar los criterios de evaluación (dataset balanceado, ambiente de trabajo, disponibilidad en internet, cantidad de emociones), luego se aplicó el preprocesamiento realizando una limpieza de los datos con el fin de que la entrada de los datos a la arquitectura sea de la mejor manera posible al reentrenar los modelos y obtener las métricas. A continuación, se podrá responder a las siguientes preguntas de investigación planteadas desde el diseño de la etapa de experimentación, como se muestra en la figura 5.1 va a iniciarse al momento que se realizó la evaluación individual por cada modalidad en la cual se aplican las métricas de evaluación correspondientes como el F1 score y la exactitud. En donde se visualizan los datos

a través de un gráfico que muestra las métricas y sus resultados v/s su modalidad. Como también una tabla con las emociones categorizadas y un resumen de los resultados por cada modalidad con respecto a las métricas de evaluación [18] como se muestra en el anexo A.2.

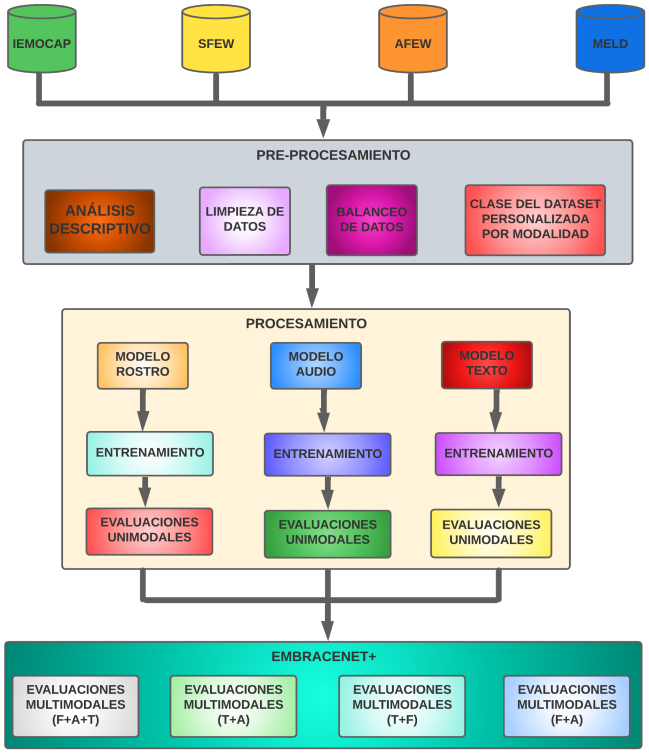


Figura 5.1: Etapa de experimentación

Luego en el anexo A.3 se presenta un enfoque multimodal donde se realizó 4 casos de experimentos relacionados con la ausencia de una de las modalidades, primero se realizó la evaluación de las tres modalidades, luego la del texto+audio, luego la del texto+rostro y por último el rostro+audio. En todos los casos se utilizó el método Embracenet+ que a su vez detecta automáticamente la ausencia de modalidades y ajusta las probabilidades. Todas estas evaluaciones se van a visualizar por medio de un gráfico de comparación entre las modalidades y sus métricas de evaluación. Paralelamente, obteniendo una tabla con las distintas emociones, métricas de evaluación por cada caso y comparando el dataset a utilizar con el dataset Iemocap [18] y Aff-Wild2.

La idea de la figura 5.1 es proporcionar una evaluación de diferentes dataset a través de la misma arquitectura, realizando un preprocesamiento previo al ser entrenado cada modalidad de forma individual con sus modelos y obtener sus evaluaciones unimodales, como también fusionar los diferentes vectores resultantes con la data y sus etiquetas para

obtener evaluaciones multimodales y conocer qué modalidad, dataset y emoción son las más adecuadas para llevarlo a una interacción humano-robot.

5.2. Condiciones de Experimentación

5.2.1. Software utilizado

Pandas: Es una librería que se utilizó para crear dataframes a partir de diferentes archivos csv de cada conjunto de datos y visualizar, manipular o recorrer las diferentes filas y columnas de la mejor manera. La definición como tal de dicha librería se aprecia en el anexo A.4

SciPy: Es una librería que se ocupó para manipular las señales de audio, ya sea con respecto a su amplitud, frecuencia y los filtros asociados. La definición de esta librería se encuentra en el anexo A.4.

Numpy: Es una librería que cumplió la función de convertir varios datos en un formato de matriz. Por ejemplo, apilar arreglos en secuencia horizontal para visualizar una muestra de frames, como también expandir la forma de una matriz de entrada que se la a pasar en la modalidad facial, devolver el promedio de los elementos de la matriz en el caso de los audios. Se encuentra más información en el anexo A.4.

Matplotlib: Es una biblioteca que permite mostrar gráficamente el comportamiento del entrenamiento de los diferentes modelos. Puede crear una imagen a partir de una matriz numpy bidimensional como también crear una figura y una cuadrícula de subparcelas con una sola llamada. La definición de dicha biblioteca se encuentra en el anexo A.4.

Pytorch: Es una biblioteca que se empleó en su totalidad en la arquitectura implementada, partiendo desde la preparación de los dataset, las técnicas de Deep Learning con los datos que van siendo entrenados y realizando el proceso de pruebas con las métricas de evaluación correspondientes [80]. Para más detalles se encuentran en el anexo A.4.

AnyDesk: Permitió conectar el equipo personal con el servidor de la universidad de manera remota para trabajar de mejor forma con sus componentes. En el anexo A.4 se encuentran más detalles.

5.2.2. Herramientas de Almacenamiento

Colab: Es un ambiente de trabajo el cual permite que cualquier persona escriba y ejecute código en Python o PyTorch a través del navegador, y es especialmente adecuado para el aprendizaje automático, el análisis de datos y la educación. Colab es un servicio de notebook Jupyter alojado que no requiere configuración para su uso, al tiempo que brinda acceso gratuito a los recursos informáticos, incluidas las GPU [81].

Drive: Es un ambiente de trabajo que permite almacenar archivos de forma segura y abrirlo o editarlos desde cualquier dispositivo. Se otorgan 15 GB de espacio en la unidad donde se pueden cargar archivos o crear archivos dentro del Google Drive, como también compartir y organizar archivos[82].

5.2.3. Algoritmos existentes

VGG19: Es una red neuronal convolucional con 19 capas de profundidad. Se ocupó en la modalidad de rostro en el cual el tamaño de las imágenes se redujeron a 48x48 como también el número de características extraídas de las capas convolucionales, por lo tanto, las capas lineales al final de la red neuronal también tienen menos parámetros [18].

MFCC: Es un método que permite la extracción de características de los coeficientes cepstrales de frecuencia mel en donde se enfoca en la modalidad del habla, lo cual permite extraer la frecuencia vocal [76].

Dialogxl: Es un método que permite guardar diálogos para detectar las emociones de una conversación y hacer un seguimiento de los diferentes participantes, todo enfocado en la modalidad de texto [18].

NLP: Es una técnica utilizada en el procesamiento del lenguaje natural para convertir la modalidad del habla en texto [83].

Embracenet+: Es una arquitectura que ayuda a fusionar las distintas modalidades a trabajar y generar una predicción final con respecto a las emociones analizadas [7].

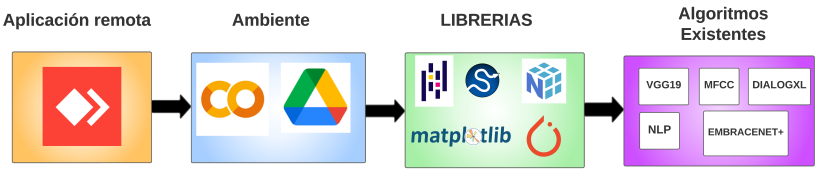


Figura 5.2: Esquema del software utilizado.

5.2.4. Hardware utilizado

En esta sección se especifica el hardware utilizado para el preprocesamiento, entrenamiento y análisis de los resultados asociado a los diferentes datasets.

5.2.4.1. Máquina Local

El hardware de nuestro equipo personal se utilizó para trabajar de la mejor manera en cada uno de los datasets y poder realizar las evaluaciones correspondientes. Las características de esta máquina son las siguientes:

- Procesador: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz
- Ram: 8,00 GB
- Almacenamiento: 256GB
- Sistema Operativo: Windows 10, 64 bits, procesador x64

La máquina local presentó ciertas limitaciones en el almacenamiento del disco duro, ya que al momento de descargar una serie de conjunto de datos que contenían un tamaño de 50 GB donde no se encontraba el espacio suficiente para trabajar con la máquina local. Por otro lado, para el proceso de implantación ocurre exactamente lo mismo, solamente que varía la capacidad de almacenamiento dentro del colab donde se realizó una conexión (montaje) con el drive que contenga cada persona. Como caso personal se tiene un drive de la universidad con un almacenamiento de 1TB para poder guardar todos los archivos y dataset a utilizar y dentro del mismo colab se otorgan una ram de 12,68GB y un disco duro que ofrecen de 107,72GB.

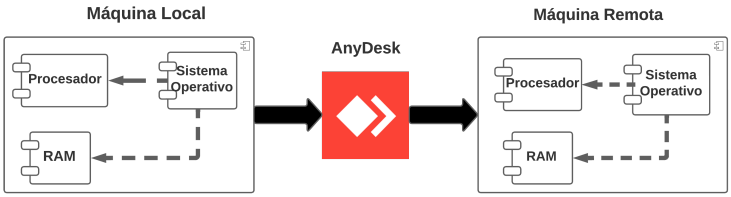


Figura 5.3: Esquema del Hardware utilizado

5.2.4.2. Máquina remota

Para resolver el problema del almacenamiento en disco duro mencionado anteriormente se utiliza el hardware de forma remota alojado en la Facultad de Ingeniería Civil Informática de la Universidad de Valparaíso para realizar el entrenamiento con dataset de mayor cantidad de datos que requiere una cantidad importante de recursos para el procesamiento. En donde este hardware presenta las siguientes características para abordar la problemática:

- Procesador: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz 3.19 GHz
- Ram: 64,0 GB
- Almacenamiento: 3TB
- Sistema Operativo: Windows 10, 64 bits, procesador x64

Capítulo 6

Ejecución de Experimentos

Esta sección presenta toda la etapa del preprocesamiento y el procesamiento de datos en los diferentes dataset de forma unimodal y multimodal.

6.1. Visualizaciones

El primer dataset a trabajar es el Sfew que contiene una serie de imágenes por frame del dataset Afew las cuales están categorizadas en diferentes carpetas con sus emociones, en la cual se dividen en un conjunto de entrenamiento y para pruebas. Lo primero que se realizó fue el preprocesamiento de los datos del dataset el cual consiste en mostrar el organigrama de carpetas y como está estructurado el dataset con respecto a sus emociones y la calidad de imágenes que se encuentran. Luego se procedió a adecuar la entrada de datos donde se realizó una clase exclusivamente para el dataset con el fin de capturar los datos de train y test con sus etiquetas y llevar a cabo el entrenamiento como se muestra en las figuras 6.1 y 6.2. Cabe destacar que en este dataset en particular, al tener una sola modalidad que es la facial, no se aplicó una evaluación multimodal.

```
classs={'Angry':0, 'Happy':1, 'Neutral':2, 'Sad':3}

class SFEW(Dataset):
  def __init__(self, root_dir='', categories={}, transform=None):
        super(SFEW, self).__init__()
        self.DataRoot = root_dir
        self.Categories = categories
        self.Transform = transform
        self.load_data()

  def load_data(self):
        self.Data = {}
        for cat in self.Categories.keys():
          for img in os.listdir(join(self.DataRoot,cat)):
              self.Data[img]=self.Categories[cat]
        self.DataKeys=list(self.Data.keys())
        print (self.Data)
```

Figura 6.1: Clase del dataset Sfew

```
def __len__(self):
      return len(self.DataKeys)

def make_shuffle(self):
      random.shuffle(self.DataKeys)

def __getitem__(self, idx):
      if torch.is_tensor(idx):
          idx = idx.tolist()

      lb= self.Data[self.DataKeys[idx]]
      dt= cv2.imread(join(self.DataRoot,list(self.Categories.keys())[lb],self.DataKeys[idx]))
      sample=  {'data': dt, 'label': lb,  'name': self.DataKeys[idx]}

      if self.Transform:
        sample= self.Transform(sample)
      return sample
```

Figura 6.2: Continuación clase del dataset Sfew

Al tener el dataset customizado con los datos de entrada ya dispuestos para entrenar en la arquitectura, se procedió a separar los conjuntos de datos de entrenamiento como se muestra en la figura 6.3 con el fin de tener en todos los archivos sus nombres correspondientes y etiquetas.

```
train_dataset = SFEW(root_dir='/content/drive/MyDrive/SFEW2.0/Train', categories=classs, transform=FaceTransform((48,48)))
len(train_dataset)
```

```
{'HarryPotter_GobletOfFire_004924334_00000004.png': 0, 'Descendants_003858520_00000022.png': 0, 'MarotAtTheWedding_011010680_00000043.png': 0
698
```

```
test_dataset = SFEW(root_dir='/content/drive/MyDrive/SFEW2.0/Val', categories=classs, transform=FaceTransform((48,48)))
test_dataset.make_shuffle()
len(test_dataset)
```

```
{'Saw3D_001444527_00000011.png': 0, 'Saw3D_001444527_00000082.png': 0, 'Saw3D_004056050_00000032.png': 0, 'Saw3D_004056050_00000003.png': 0,
309
```

Figura 6.3: Conjunto de Train y Test en Sfew

En la siguiente figura 6.4 se da a conocer el comportamiento que va teniendo el conjunto de train y test a lo largo del entrenamiento. Por lo cual en el eje x se alojan las cantidades de épocas que se realizaron para train y test con respecto a su exactitud o datos perdidos que se denotan en el eje y. Donde la línea azul representa los datos de las pruebas y la naranja del entrenamiento. Se puede extraer cierta información del gráfico en el cual el conjunto de train como el de test no están realizando un aprendizaje a lo largo de las diferentes épocas, lo que implica que el modelo es de baja calidad y una precisión muy baja para identificar las emociones a partir de las imágenes entrenadas. Para obtener más información del entrenamiento se encuentra en el anexo A.5.

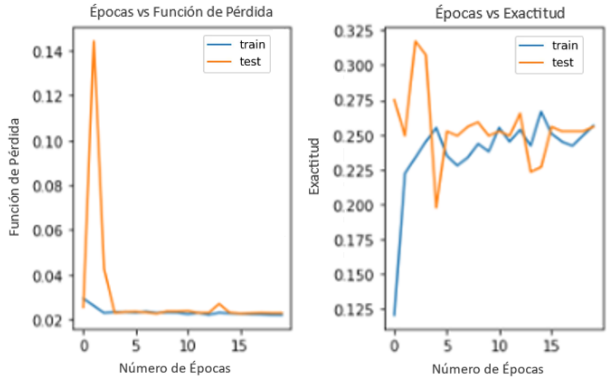


Figura 6.4: Comportamiento Train y Test Modalidad Facial en Sfew

En la figura 6.5 se calculó el promedio de la exactitud de las pruebas donde se recorre cada ejemplo por su data y con su etiqueta correspondiente. El resultado no es lo más esperado, ya que 0.29 es muy impreciso con respecto a la cantidad de datos que se entrenaron, por lo que la calidad del modelo trabajado es de baja calidad para detectar si ciertas imágenes son de una determinada emoción o no y lo que trae problemas para la interpretación de los resultados.

```
Tacacc = 0
Tcont = 0
yt , yp = [], []
with torch.no_grad():
    for sample in tqdm(test_dataloader):
        data = sample['data'].float().cuda()
        label = sample['label'].flatten()
        out = model(data).to('cpu')
        Tacacc += (torch.argmax(out, dim=1) == label).float().sum()
        yt += label.tolist()
        yp += torch.argmax(out, dim=1).tolist()
        Tcont += data.shape[0]
print('TT test_acc: %.4f'%(Tacacc/Tcont))
```

```
100% 5/5 [05:08<00:00, 58.17s/it]
TT test_acc: 0.2945
```

Figura 6.5: Promedio exactitud en Sfew

El dataset afew contiene 3 modalidades presentes como son el rostro, audio y texto. Lo cual va a permitir realizar evaluaciones multimodales, para ello se tienen que realizar los mismos procedimientos que en el dataset anterior. Partiendo por un preprocesamiento de los datos en donde se obtuvieron la cantidad de emociones que se encuentran dentro del dataset, conocer la distribución de carpetas categorizadas por emociones dentro del dataset con sus respectivos videos y limpiar ciertos archivos que están dañados o que están repetidos en el conjunto de test. En este dataset en particular se encuentran videos asociados a las emociones, por lo que se van a generar imágenes por frames de cada emoción en una carpeta específica como se muestran en la figura 6.6

Se van recorriendo las emociones presentes dentro del dataset como también la ruta en donde se alojan los videos originales a los cuales se van a guardar en una variable llamada val_images que se va a pasar por un ciclo iterativo con la finalidad de obtener una carpeta por cada emoción. Se van recortando los videos originales categorizados del dataset para el conjunto de train y test en formato .png obteniendo diferentes frames y guardándolos con su id correspondiente en la carpeta Afew_images_videos con sus emociones correspondientes.

```
def tidy_SFEW(origin_dir,destination_dir,emo_video_labels):

  for emo in emotions_list:
    if not os.path.exists(destination_dir+emo):
      os.makedirs(destination_dir+emo)
  val_images=glob.glob(origin_dir+"/**/*.avi", recursive = True)
  i=0
  for image_path in val_images:
    image_video_id=image_path.split("/")[-1].split(".")[-2]
    image_id=image_path.split("/")[-1].split(".")[-2]
    sfew_emo=image_path.split("/")[-2]

    try:
      afew_emo=emo_video_labels[image_video_id]
      destination=destination_dir+afew_emo

      if not os.path.exists(destination):
        os.makedirs(destination)
      cap = cv2.VideoCapture(image_path)
      count=0
      while True:
        ret, frame = cap.read()
        if not ret:
          break
        cv2.imwrite(destination+"/"+ image_id + str(count)+ ".png", frame)
        count+=1
```

Figura 6.6: Preprocesamiento en Afew

Para obtener más información sobre la distribución de los frames en las diversas carpetas y el contenido de ellas se encuentra en el anexo A.6, A.7 y A.8.

Ya obteniendo las imágenes recortadas por frame de cada video en particular y etiquetadas cada una de ellas, se lleva a cabo la clase personalizada para el dataset como se muestra en el anexo A.9 con la particularidad de recorrer todo el conjunto de Train y de Test para prepararlo para el proceso de entrenamiento en la red neuronal.

En la figura 6.7 se llama a la clase del dataset especificando los parámetros importantes que son la ruta hacia donde están alojados los frames del conjunto de Train y Test con sus etiquetas correspondientes y nombre de cada archivo para el entrenamiento de ambos en la arquitectura. La función transform es importante para gestionar el tamaño de cada imagen, como también el shuffle que se utiliza para mezclar las imágenes con diferentes personajes y posturas, es decir, desordena los ejemplos para entrenamiento, evitando que todo el batch sea de una misma clase o emoción. Cabe recordar que la cantidad de archivos se redujo a consecuencia de los videos duplicados encontrados en el dataset original.

Los datos se pasan al entrenamiento en un formato de tensor con el contenido propio de la imagen (data) como también la etiqueta a la que corresponde. El tamaño de las entradas que se cargan para ingresar al entrenamiento por parte del conjunto de train es de 152 elementos y para el test 79 elementos.

```
train_dataset = AFEW(root_dir= "/content/drive/MyDrive/Afew2/Train/Afew_images_videos/",
categories=classs, transform=FaceTransform((48,48)))
train_dataset.make_shuffle()
len(train_dataset)
```

```
{'01240400028.png': 0, '01240400029.png': 0, '01240400030.png': 0, '01240400031.png': 0,
9721
```

```
test_dataset = AFEW(root_dir= "/content/drive/MyDrive/Afew2/Val/Afew_images_videos/",
categories=classs, transform=FaceTransform((48,48)))
test_dataset.make_shuffle()
len(test_dataset)
```

```
{'0023384140.png': 0, '0023384141.png': 0, '0023384142.png': 0, '0023380140.png': 0, '
5024
```

Figura 6.7: Conjunto de Train y Test en Afew

Al tener entrenado el modelo con la data que se ha querido trabajar, se realizan las métricas de evaluación y un gráfico de comparación entre la cantidad de épocas que se llevaron a cabo en el entrenamiento y los datos perdidos, como también la precisión que logró el modelo en el conjunto de entrenamiento y de pruebas. En la figura 6.8 se puede apreciar que el conjunto de train denotado con las líneas naranja y la de test con la línea azul no están aprendiendo de la misma forma debido a los problemas de calidad de las imágenes, formato de los archivos y la cantidad de conjuntos de train y test. El comportamiento de estos dos conjuntos representa una inestabilidad significativa en el modelo entorno a la calidad y precisión a la hora de predecir las emociones a través de imágenes. En el anexo A.10 y A.11 se muestra en más detalle el entrenamiento de la modalidad facial.

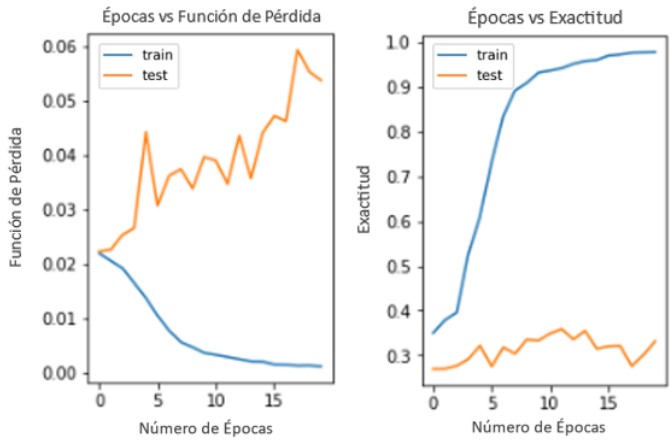


Figura 6.8: Comportamiento Train y Test Modalidad Facial en Afew

Por último, en la figura 6.9 se obtiene el promedio de la exactitud que tiene el conjunto de test a la hora de ser entrenado con el modelo de rostro. Lo cual representa el porcentaje de casos que el modelo ha acertado para encontrar emociones correctas a las que están representando las personas en las imágenes.

```
from sklearn.metrics import classification_report
Tacacc = 0
Tcont = 0
yt,yp=[],[]
with torch.no_grad():
    for sample in tqdm(test_dataloader):
        data = sample['data'].float().cuda()
        label = sample['label'].flatten()
        out = model(data).to('cpu')
        yt += label.tolist()
        yp += torch.argmax(out, dim=1).tolist()
        Tacacc += (torch.argmax(out, dim=1) == label).float().sum()
        Tcont += data.shape[0]
print('TT test_acc: %.4f'%(Tacacc/Tcont))
```

```
100% 79/79 [1:03:58<00:00, 37.78s/it]
TT test_acc: 0.3153
```

Figura 6.9: Promedio Test-exactitud en Afew

Lo siguiente se trabajó con la modalidad de audio en este dataset afew que contiene la misma distribución de carpetas con las emociones categorizadas y se ocupan los mismos videos de donde se extrajeron las imágenes por frame. Lo siguiente es convertir el formato de audio de .avi a .waw con el fin de trabajar mejor las frecuencias mel en la arquitectura y posteriormente tener un formato de audio validó para adquirir la conversión a texto. Como se muestra en la figura 6.10 se convierte el formato de audio de todos los videos que están categorizados por emociones y se van generando una nueva carpeta para train y test que contengan la misma estructura de carpetas que la original.

```
def convert_to_audio_ne(set_path,set_name):
  f = open("convert_to_audio.sh", "w")
  path_afew_set=set_path
  urls_afew = glob.glob(pathname=path_afew_set + '/*')

  for url_emo in urls_afew:
    urls_emotion = glob.glob(pathname=url_emo + '/*')
    emo=os.path.basename(url_emo)
    new_dir=set_name+'_audio_ne/'+ emo + '/'
    if not os.path.exists(new_dir):
      os.makedirs(new_dir)

    for video_file in urls_emotion:
      file_name=os.path.splitext(os.path.basename(video_file))
      f.write("ffmpeg -i "+video_file+" -vn -ac 1 "+new_dir+file_name[0]+".wav"+'\n')
      dict_xy_afew[file_name[0]]=emo
  f.close()
```

Figura 6.10: Conversión Formato audio en Afew

Se calcula la cantidad de emociones que existen en cada carpeta del dataset con el fin de tener un estimado de lo que se va a procesar en la arquitectura, tomando en cuenta que las emociones principales para el estudio son angry, happy, neutral y sad para seguir con el estándar de lo trabajado en el dataset iemocap. Para tener una representación gráfica de las cantidades de audios con respecto a las emociones a utilizar, se puede mencionar que este dataset contiene una serie de ejemplos balanceados (no existe tanta diferencia entre las cantidades de emociones) con respecto a las emociones a trabajar como se muestra en la figura 6.11 por lo cual no es necesario realizar un balanceo de datos con las emociones.

```
emotion:  Angry
number of audios:  197
-----
emotion:  Disgust
number of audios:  113
-----
emotion:  Fear
number of audios:  127
-----
emotion:  Happy
number of audios:  213
-----
emotion:  Neutral
number of audios:  207
-----
emotion:  Sad
number of audios:  177
-----
emotion:  Surprise
number of audios:  120
```

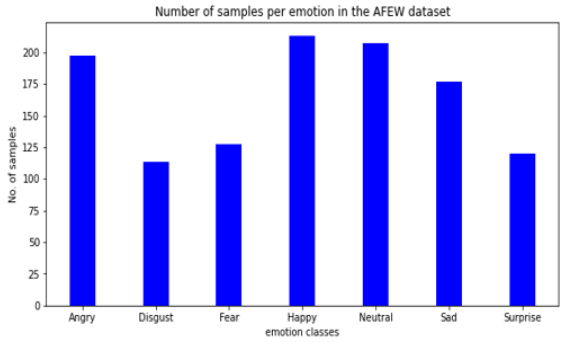


Figura 6.11: Gráfico de emociones en Afew

Tomando como base el preprocesamiento que viene con el dataset original para la modalidad de audio, lo destacable es la función que se muestra en la figura 6.12 donde los diferentes audios con un largo determinado se van a estandarizar la duración al máximo relleno de audio len con 0s para que todos tengan las mismas dimensiones. Paralelamente, se realiza la conversión del audio a espectrogramas para entrenarlo en la arquitectura con las diferentes características y se guardan en vectores para representarlo de la mejor manera. Por último se va a calcular el log-mel-spectrogram ocupando las diferentes librerías de librosa y se va a normalizar los niveles del audio si se requiere.

La principal diferencia entre el log-mel spectrogram y otros espectrogramas radica en su enfoque en imitar la percepción auditiva humana y en resaltar las características relevantes del sonido de una manera que sea más útil para tareas específicas de procesamiento de audio y análisis.

```python
def preprocess_audio(audio_file,desired_audio_len=137063,frame_size=0.025,stride=0.01,normalize=False,preemphasis=True,norm_waveform=False):

  if norm_waveform==True:
    audio_file = (audio_file  - audio_file.mean()) / audio_file.std() + 1e-10
  y,sr = librosa.load(audio_file)
  if preemphasis==True:
    y=librosa.effects.preemphasis(y, coef=0.97)
  frame_size=int(round(frame_size*sr))
  stride_size= int(round(stride*sr))
  mel_spectrogram = librosa.feature.melspectrogram(y,sr=sr,n_fft=frame_size,window=scipy.signal.hamming,hop_length=stride_size,n_mels=320)
  log_mel_spectrogram = librosa.power_to_db(mel_spectrogram)
  if normalize==True:
    mean = np.mean(log_mel_spectrogram, axis=0)
    std = np.std(log_mel_spectrogram, axis=0)
    log_mel_spectrogram = (log_mel_spectrogram - mean) / std + 1e-10
  return log_mel_spectrogram.T,sr
```

Figura 6.12: Preprocesamiento audio en Afew

Posteriormente, se recorre la carpeta donde se encuentran los audios de train y test para llamar a la función que realiza todo el procesamiento del audio y convertirlo a espectrogramas, seguido de esto se crea una nueva carpeta llamada mel que contendrá todos los

espectrogramas con la misma distribución de carpeta que la original y guardando en un formato npy los espectrogramas con sus etiquetas correspondientes como se muestra en la figura 6.13

```
def create_set(url_list, dict_set, desired_audio_len=137063, normalize=False, preemphasis=True, mode='Train' ):
  x_data,y_data=[],[]
  mean=[]; spec_values=[]
  i=0
  for url_list2 in os.listdir(url_list):
    for audio_file in glob.glob(url_list+'/'+url_list2+'/*'):
      file_id=os.path.splitext(os.path.basename(audio_file))[0]
      melSpec_dB,sr = preprocess_audio(audio_file,desired_audio_len=137063,frame_size=0.025,stride=0.01,
      normalize=normalize,preemphasis=preemphasis)
      mean.extend(np.mean(melSpec_dB,axis=0))
      spec_values.extend(melSpec_dB)
      if not os.path.exists(os.path.join('mel',mode,url_list2)):
        os.makedirs(os.path.join('mel',mode,url_list2))
      np.save(os.path.join('mel',mode,url_list2, file_id +'.npy'), melSpec_dB)
      label= emo_dict[url_list2]#get_categorical_label(file_id)
      dict_set[audio_file]=label
      x_data.append(melSpec_dB)
      y_data.append(label)
  return x_data,y_data,mean,spec_values
```

Figura 6.13: Creación Espectogramas en Afew

En el anexo A.12 se visualiza la carpeta mel con el contenido de las carpetas. Luego se llamará a la función create_set con los parámetros correspondientes como la ruta donde están alojados los audios, tamaño del audio, normalización y el preemphasis con el fin de guardar en dos variables distintas el conjunto de train y test con los espectrogramas ya listos para el entrenamiento como se muestra en la figura 6.14.

```
normalize=False; preemphasis=True; desired_audio_len=137063
num_classes=7
train_data={}
test_data={}
```

Figura 6.14: Preparación de la data en Afew

Lo siguiente es crear la clase propia para el audio que se encuentra en el anexo A.13 y A.14. Al tener el conjunto de train y de test ya por separado y bien definido con los archivos y sus etiquetas, se procede a entrar a la arquitectura para entrenar los datos.

```
Audio_trainset = AudioDataset(root_dir='mel/Train/', categories=classs, transform=AudioTransform((259,128)))
Audio_trainset.make_shuffle()
len(Audio_trainset)
```

```
{'Angry': 0, 'Happy': 1, 'Neutral': 2, 'Sad': 3}
557
```

```
Audio_testset = AudioDataset('mel/Test/', categories=classs, transform=AudioTransform((259,128)))
Audio_testset.make_shuffle()
len(Audio_testset)
```

```
{'Angry': 0, 'Happy': 1, 'Neutral': 2, 'Sad': 3}
237
```

Figura 6.15: Conjunto de Train y Test en Afew

En el anexo A.15, A.16 y A.17 se da a conocer el entrenamiento del audio y sus métricas correspondientes a lo largo de las épocas establecidas. Luego se realizó un gráfico para representar el comportamiento del conjunto de train y test dentro de la arquitectura al momento de ser entrenada a través de 40 épocas, tomando en cuenta sus métricas correspondientes como se muestra en la figura 6.16. Cabe destacar que en el primer gráfico se da a conocer en el eje x las épocas establecidas para el entrenamiento y en el eje y la función de pérdida, donde en cada conjunto (train y test) no existe una variación muy grande con respecto a lo que se ha equivocado el modelo para predecir las emociones. Por otro lado, el segundo gráfico representa en el eje x la cantidad de épocas y en el eje y la exactitud para cada conjunto. Lo importante es que a partir de la época 25 existe un sobreajuste en donde la red deja de aprender debido a la calidad de los datos presentes en el dataset original.

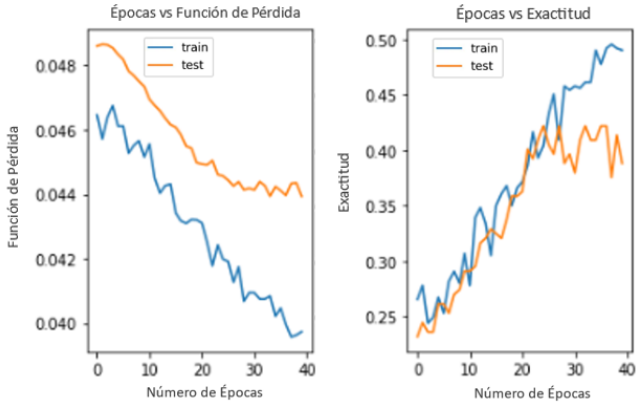


Figura 6.16: Comportamiento Train y Test Modalidad Audio en Afew

Luego se tomó en cuenta el conjunto de test con su data, etiqueta y la salida del modelo para calcular el promedio de las pruebas con respecto a las métricas de la exactitud como se muestra en la figura 6.17 como también guardar la lista de emociones predichas y las emociones reales para realizar un clasification report que consiste en medir la calidad de las predicciones de un algoritmo de clasificación y apreciar cuántas predicciones son verdaderas y cuántas son falsas. Donde más adelante se representaran a partir de las emociones utilizadas y con las métricas correspondientes.

```
from sklearn.metrics import classification_report
Tacacc = 0
Tcont = 0
yt,yp=[],[]
with torch.no_grad():
    for sample in tqdm(test_dataloader):
        data = sample['data'].float().cuda()
        label = sample['label'].flatten()
        out = model(data).to('cpu')
        yt += label.tolist()
        yp += torch.argmax(out, dim=1).tolist()
        Tacacc += (torch.argmax(out, dim=1) == label).float().sum()
        Tcont += data.shape[0]
print('TT test_acc: %.4f'%(Tacacc/Tcont))
```

```
100% 8/8 [00:00<00:00, 12.88it/s]
TT test_acc: 0.3797
```

Figura 6.17: Promedio exactitud Test en Afew

Por último, se trabajó con la modalidad de texto que implica una serie de modificaciones en los archivos que se tienen hasta el momento, como son los audios de entrenamiento y de pruebas. Pero en general el procedimiento a realizar es similar a las demás modalidades.

En la figura 6.18 se va a extraer el audio de entrenamiento y realizar ciertas excepciones de algunos que no serán reconocidos, ya sea por la calidad del audio o por su duración. Donde se van listando en una carpeta llamada txt con el mismo organigrama de antes, es decir, con su carpeta de entrenamiento y categorizado por la emoción y luego se abre dicho archivo para escribir lo traducido por la función. Esto mismo se realizó para los audios de pruebas, que la única diferencia que se van a guardar en una carpeta distinta llamada test con sus etiquetas correspondientes y sus archivos txt con el contenido asociado al audio trabajado.

```
def extract_audio(file_id):
    r = sr.Recognizer()
    with sr.AudioFile(file_id) as source:
        audio = r.record(source)
    try:
        s = r.recognize_google(audio)
        return s
    except Exception as e:
        print("Exception: " + str(e))
        return 'ERROR'

def extract_audio2(url_list, mode='Train'):
  i=0
  for url_list2 in os.listdir(url_list):
      for audio_file in glob.glob(url_list+'/'+url_list2+'/*'):
        file_id=os.path.splitext(os.path.basename(audio_file))[0]
        archivo=extract_audio(audio_file)

        if not os.path.exists(os.path.join('txt',mode,url_list2)):
          os.makedirs(os.path.join('txt',mode,url_list2))
        file=open(os.path.join('txt',mode,url_list2, file_id +'.txt'),'w')
        file.write(archivo)
```

Figura 6.18: Extracción de audios a texto en Afew

Al tener estos archivos de texto generados se realizó una función como se muestra en la figura 6.19 que permitió recorrer los archivos txt tanto para los de entrenamiento como para el de las pruebas con su texto asociado y su etiqueta para agregarlos un dataframe

sin considerar los archivos erróneos. Además de guardar el dataframe como un csv con la finalidad de convertirlos a un vector e ingresarlos como datos de entrada al modelo.

```
def extract_audio3(set):
    textos=[]
    labels=[]

    for element in set:
      texto=element['data']
      label=element['label']

      if texto=='ERROR':
        continue
      textos.append(texto)
      labels.append(label)
    return pd.DataFrame({'text':textos, 'target':labels})

T=extract_audio3(Text_trainset)
T.to_csv('text_train.csv',index=False)
```

Figura 6.19: Conversión txt a Dataframe en Afew

En la siguiente figura 6.20 se muestra el dataframe del conjunto de entrenamiento donde se destaca el texto en una columna y por la otra las etiquetas con sus emociones correspondientes donde cada una tiene un significado [0: Happy, 1: Neutral, 2: Sad, 3: Angry]. Como también, este procedimiento se realizó para el conjunto de pruebas con el fin de entrar a la arquitectura de la mejor manera.

	text	target
0	you didn't do your job you weren't counting yo...	0
1	we're gathered here to	1
2	see	1
3	Rochester Mayo	2
4	the fearful of each other	2
...	...	...
304	puppies	1
305	I cannot actually allow you to accompany the b...	2
306	I don't know what I've done but I am so sorry	0
307	what we saw on the monitor wasn't actually hap...	2
308	fair I surely hope not what about the information	2

309 rows × 2 columns

Figura 6.20: Dataframe Train en Afew

Al tener los archivos .csv de train y test se realizó el entrenamiento sobre el modelo, las cuales contienen una MLP para permitir el procesamiento de los datos con el fin de encontrar las métricas de desempeño como se muestra en la figura 6.21. Al tener el entrenamiento del texto con sus etiquetas en los dataframe de train y test se obtuvieron las métricas de evaluación por cada emoción como son el F1 score y la exactitud. Dando a conocer el promedio alcanzado durante el entrenamiento, donde las evaluaciones unimodales de la modalidad de texto se van a mostrar en la tabla 7.4 con mayor detalle.

```
models = ['roberta-large-nli-stsb-mean-tokens', 'distilbert-base-nli-mean-tokens', 'bert-large-nli-stsb-mean-tokens']
train_embeddings = []
test_embeddings = []
for model_name in models :
    model = SentenceTransformer(model_name)
    train_df_embedded = model.encode(T["text"].to_list())
    test_df_embedded = model.encode(T2["text"].to_list())
    clf = MLPClassifier(hidden_layer_sizes=(500, 300), random_state=1, early_stopping = True)
    clf.fit(train_df_embedded, T["target"].tolist())
    pickle.dump(clf,open('/content/text.clf'.format(model_name),'wb'))
    train_embeddings.append(clf.predict(train_df_embedded))
    test_embeddings.append(clf.predict(test_df_embedded))
```

Figura 6.21: Entrenamiento en Afew

Lo siguiente es crear la clase del dataset para la parte multimodal en donde se encuentra en gran detalle en el anexo A.18

Como se muestra en la figura 6.22 se va llamando a la clase del dataset con los parámetros mencionados anteriormente con la finalidad de verificar la cantidad del conjunto de train y test que va a hacer entrenado dentro de la arquitectura del embracenet+.

```
train_dataset = DatasetAFEW(Classes, pred_data, audi_data, text_data, T42,
                            'average', transform=FusionTransformer(''))
test_dataset = DatasetAFEW(Classes, pred_data, audi_data, text_data1, T42,
                           'average', mode = 'Test', transform=FusionTransformer(''))

len(train_dataset), len(test_dataset)
```

Figura 6.22: Conjunto de Train y Test multimodal Afew

Al tener los conjuntos de datos de train y test ya formados se procede a realizar el entrenamiento multimodal en donde se muestra en el anexo A.19, A.20 y A.21.

En la figura 6.23 representa el comportamiento del conjunto de train y test al ser entrenado bajo la arquitectura del embracenet. Lo importante a resaltar es que se presentan dos gráficos con una línea azul y naranja que significan el conjunto de train y test específicamente, en los cuales partiendo por el de la izquierda en el eje x representa las épocas que se han establecido para el entrenamiento de los conjuntos de datos y en el eje y se encuentra la función de pérdida como tal. Luego en el gráfico de la derecha se encuentra en el eje x las épocas y en el eje y la exactitud de los conjuntos de datos. Cabe destacar que en esta oportunidad se encuentra la concatenación de todas las modalidades presentes dentro del dataset donde se aprecia en la gráfica de la izquierda una función de pérdida más notoria por el lado del conjunto de pruebas y en el gráfico de la derecha se puede apreciar que el conjunto de pruebas contiene la exactitud menor que el conjunto de train es decir el modelo está más impreciso a la hora de predecir las emociones en el conjunto de pruebas debido a la calidad de los datos.

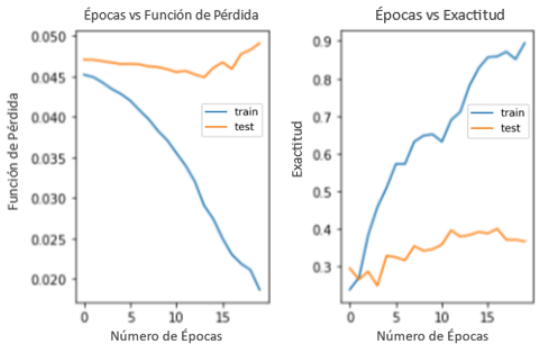


Figura 6.23: Comportamiento Train y Test Multimodal Afew

Para finalizar se tiene el dataset Meld donde se realizó el mismo procedimiento que los anteriores conjuntos de datos realizando un análisis descriptivo para conocer la distribución de carpetas, cuantas emociones existen en el conjunto de train y test o los archivos csv que contiene las etiquetas correspondientes como también los textos de los diversos videos. Esto se puede apreciar en las figuras 4.10 y 4.4 respectivamente. Primero que nada, en la modalidad facial existe una particularidad que en los diferentes videos se encuentran muchas personas en las escenas que están demostrando emociones y al realizar el preprocesamiento de estos videos recortando las escenas por frames se puede apreciar lo dicho y no están identificados las personas. Paralelamente, se buscó diversos trabajos asociados al preprocesamiento de la modalidad facial de dicho dataset y no se encontraron fuentes en la web.

La data original de train y test son videos grupales como se muestra en la figura 4.12 que tienen un formato en específico, que se encuentra en los csv correspondientes. Los csv de train contienen 9988 videos con las 7 emociones y el test 2747 videos.

En la siguiente figura 6.24 se realiza un recorte de los diferentes videos con la librería cv2 y el método cv2.videocapture() en donde se generan 5 frames en formato png por cada video sin categorizarlas por emoción solamente identificándolo por su id de formato y si pertenece al conjunto train o test. Para más detalles se pueden encontrar en el anexo A.22.

```
destination= '/content/drive/MyDrive/Meld/MELD.Raw/Frames_Train'

for i in os.listdir('/content/drive/MyDrive/Meld/MELD.Raw/train_splits'):
  image_path= '/content/drive/MyDrive/Meld/MELD.Raw/train_splits/'+ i
  cap = cv2.VideoCapture(image_path)
  count=0
  cont=0
  while True:
    ret, frame = cap.read()
    if cont > 5:
      break
    if not ret:
      break
    cv2.imwrite(destination+"/"+ i[:-3] + str(count)+ ".png", frame)
    count+=1
    cont+=1
```

Figura 6.24: Recorte de frames en el conjunto de train en Meld

Por otro lado, para solucionar el problema de los videos grupales y de los actores que no están identificados, se crea un dataset personalizado a partir de una grabación de la serie Friends en HBO temporada 1 capítulo 1 y temporada 2 capítulo 2. Se cargan los modelos de detección de rostros en la siguiente figura 6.25 y con la ayuda de la librería cv2 se van recortando los videos por frames y se guardan en una carpeta llamada Data. Los modelos de detección de rostros que se usaron son res10_300x300_ssd_iter_140000.caffemodel es un archivo que contiene los parámetros entrenados de un modelo de detección de objetos SSD basado en la arquitectura ResNet-10, que fue entrenado en imágenes de 300x300 píxeles durante 140.000 iteraciones. Este archivo se puede utilizar para realizar detecciones de objetos en imágenes en tiempo real. Deploy.prototxt.txt es un archivo de configuración en el marco Caffe que define la arquitectura de una red neuronal preentrenada para inferencia o evaluación, sin contener los valores de los pesos entrenados.

Caffe es un framework, que se utilizó por su eficiencia en tiempo real para la detección de imágenes, como también para usar modelos preentrenados y la transferencia de aprendizaje, provee detectores de rostro, la cual puede procesar más de un millón de imágenes en un solo día con la GPU de medios estándar que es de milisegundos por imagen. Desde su biblioteca de código abierto, una gran cantidad de investigaciones son impulsadas por Caffe y cada día algo nuevo está saliendo de ella [84].

```
prototxtPath = r"/content/drive/MyDrive/Videos-Grupos/Archivos/deploy.prototxt"
weightsPath = r"/content/drive/MyDrive/Videos-Grupos/Archivos/res10_300x300_ssd_iter_140000.caffemodel"
net = cv2.dnn.readNet(prototxtPath, weightsPath)
```

Figura 6.25: Cargando modelo de detección de rostros en Meld

En la figura 6.26 se da a conocer el procedimiento para la captura de rostros a partir del video grabado en HBO de la serie friends donde se capturan los frames y se va tomando el bounding box de la detección escalado de acuerdo a las dimensiones de la imagen las cuales se validan, se extrae el rostro captado y se convierte BGR a GRAY para finalmente redimensionar el tamaño de los frames a 224 x 224 y guardarlo en la carpeta de destino. Se encuentra más detalles de la distribución de carpetas en el anexo A.23 y A.24.

```
for i in range(0, out.shape[2]):
    if out[0, 0, i, 2] > 0.3:
        box = out[0, 0, i, 3:7] * np.array([frame.shape[1], frame.shape[0], frame.shape[1], frame.shape[0]])
        (Xi, Yi, Xf, Yf) = box.astype("int")

        if Xi < 0: Xi = 0
        if Yi < 0: Yi = 0
        face = frame[Yi:Yf, Xi:Xf]
        face = cv2.resize(face, (224, 224), interpolation = cv2.INTER_LINEAR)
        cv2.imwrite(DataPath + '/friends_{}.jpg'.format(count), face)
        cv2.rectangle(frame, (Xi, Yi),(Xf, Yf), (0,0,255),1)
        count = count + 1
cv2.imshow("Frame", frame)
```

Figura 6.26: Captura de Rostros en Meld

Luego en la figura 6.27 se entrena una red neuronal Vggface que en realidad es una Vgg de 16 capas que ha sido entrenado con la data de Vggface que es la data para reconocer actores de 2600 actores aproximadamente, la cual contiene por actores unas 10 mil imágenes y entrena con 2,6 millones de imágenes en total. Con la cual se trabaja con las 6 clases que vienen siendo los nombres de los actores principales (Chandler, Ross, Joey, Rachel, Mónica y Phoebe) y lo que ingresa a la capa de entrada son imágenes de 224 x 224 las cuales van pasando por diversas capas ocultas que se destaca la extracción de características compuesta por la convolución y la reducción de dimensiones y finalmente las capas de salidas que es donde se encuentra la parte de la clasificación.

```
nb_class = 6; hidden_dim = 4096
inputs = tf.keras.Input(shape=(224, 224, 3))
custom_vgg_model=tf.keras.applications.vgg16.VGG16(include_top=False, weights='imagenet',input_shape=(224, 224, 3))

last_layer = custom_vgg_model.get_layer('block5_pool').output
x = tf.keras.layers.Flatten(name='flatten')(last_layer)
x = tf.keras.layers.Dense(hidden_dim, activation=tf.nn.relu, name='fc6')(x)
x = tf.keras.layers.Dropout(.3)(x)
x = tf.keras.layers.Dense(hidden_dim, activation=tf.nn.relu, name='fc7')(x)
x = tf.keras.layers.Dropout(.3)(x)
x = tf.keras.layers.Dense(nb_class, activation=tf.nn.softmax, name='fc8')(x)
custom_vgg_model = tf.keras.Model(custom_vgg_model.input, x)
custom_vgg_model.summary()
```

Figura 6.27: Red Neuronal vgg16 en Meld

Se van a entrenar los frames de cada actor con un optimizador adam, un tamaño del batch de 64 durante 30 épocas para detectar la función de pérdida de ambos conjuntos de train y test como también la exactitud del propio modelo. Los resultados obtenidos muestran la exactitud igual a 0.9987 y una función de pérdida igual 0.0093.

```
custom_vgg_model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
hist = custom_vgg_model.fit(X, y, batch_size=64, validation_split=0.20, epochs=30)
```

```
Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/30
41/41 [==============================] - 25s 454ms/step - loss: 3.2352 - accuracy: 0.5218 - val_loss: 12.3582 - val_accuracy: 0.0653
Epoch 2/30
41/41 [==============================] - 16s 394ms/step - loss: 0.3240 - accuracy: 0.9389 - val_loss: 15.8867 - val_accuracy: 0.0778
Epoch 3/30
41/41 [==============================] - 16s 399ms/step - loss: 0.1494 - accuracy: 0.9790 - val_loss: 17.3470 - val_accuracy: 0.0731
```

Figura 6.28: Entrenamiento en Meld

Al tener ya la red entrenada con los actores de Friends se realiza el mismo procedimiento con los frames del dataset meld, cargando el modelo de la Vgg16 y los modelos de detección de rostros.

Esta red que fue mencionada en la figura 6.27 tiene una parte de la entrada que es el preprocesamiento de la imagen con un tamaño de 224x224 donde se van a leer las imágenes desde una carpeta, se van a ir detectando los rostros y reconociendo a los actores, lo siguientes son las capas ocultas donde se encuentran la extracción de características y finalmente

las capas de salida en donde está la parte de clasificación. En la parte de extracción de características está compuesta por la convolución y por la reducción de dimensiones. Donde la convolución tiene la imagen y se lleva a cabo un filtro que recorre toda la imagen con la finalidad de resaltar ciertas características, por ejemplo en un rostro cuando se pasa un filtro x el resultado arrojará los bordes del rostro, luego se pasa por otro filtro que es resaltar las líneas verticales del rostro con tal de tener n características. Al tener la nueva imagen se va a reducir la dimensión que es el maxpooling que va a permitir mantener la información que se necesita (manteniendo los bordes, las líneas verticales y horizontales del rostro) para que en el siguiente paso se pueda buscar una característica general de toda la imagen. Al pasar por todas las convoluciones y maxpooling al final se obtendrá una matriz que va a representar las características de la imagen, pero en forma general. Esta matriz se convierte a vector en donde estará almacenado todas las características de la imagen. Finalmente, está la etapa de clasificación a la que se pasa el vector, la cual tiene valores en cada posición y se van realizando multiplicaciones por un peso determinado y sumando esos resultados de cada posición donde se busca resaltar los pesos adecuados para cada valor. En el entrenamiento se busca resaltar ciertas posiciones del vector como la posición inicial y la final, donde se le da un peso de 0.9, ya que ayudan a que la clasificación sea correcta.

Se van leyendo los frames con cv2.imread y se identifican especialmente los rostros de los actores como también se redimensionan a $244 \cdot 244$. Por otro lado, a partir de la precisión que tiene cada imagen, se van considerando y categorizando por carpeta en su conjunto determinado. A continuación en la figura 6.29 se da a conocer todo lo mencionado anteriormente.

```
CATEGORIES = ["Chandler","Joey","Monica","Phoebe","Rachel","Ross"]
for img in imglist:
  j+=1
  if img.endswith('.png'):
    image_path = img_path + "/" + img
    frame = cv2.imread(image_path)
    blob = cv2.dnn.blobFromImage(frame, 1, (224,224))
    net.setInput(blob)
    out = net.forward()
    for i in range(0, out.shape[2]):
      if out[0, 0, i, 2] > 0.5:
        box = out[0, 0, i, 3:7] * np.array([frame.shape[1], frame.shape[0], frame.shape[1], frame.shape[0]])
        (Xi, Yi, Xf, Yf) = box.astype("int")
        if Xi < 0: Xi = 0
        if Yi < 0: Yi = 0
        face = frame[Yi:Yf, Xi:Xf]
```

Figura 6.29: Recorte Frames y Reconocimiento de actores en Meld

El error comienza cuando ingresan rostros que son extras (personas de poca relevancia), en donde la red clasifica a esos extras como los actores principales, esto se debe a que en la data de entrenamiento no se encuentran muchas clases en la cual son 6 para cada actor. Luego se realiza una limpieza del dataset de forma manual para solucionar este problema en donde cada actor contiene una carpeta en específico con sus frames y los extras se van

separando a otra carpeta para no considerarlos dentro del entrenamiento en la arquitectura como se da a conocer en más detalle en el anexo A.25 y A.26.

Para evidenciar los frames recortados del actor Chandler extraídos de los diferentes videos dentro del dataset se muestran en la siguiente figura 6.30

Figura 6.30: Frames recortados de Chandler en Meld

Las cantidades que se obtiene mediante la separación manual de extras y actores es de 10260 para train y 10261 para test tomando en cuenta las 7 emociones.

Para tener una partición de 70 % para train y un 30 % de test se recorta el csv de test para tener una cantidad asociada a ese porcentaje.

Siguiendo el mismo procedimiento que en los demas dataset se procede a crear la clase de dataset en el anexo A.27 y A.28.

Se llama a la clase del dataset con sus parámetros como son el csv, el diccionario en donde se encuentran las emociones a utilizar, el límite del balanceo hacia la emoción neutral y la redimensión de las imágenes. En la siguiente figura 6.31 se encuentran 3098 imágenes para train y 1203 para test.

```
train_dataset = Meld(roots='/content/drive/MyDrive/Meld/MELD.Raw/Train cropped images/',
csv_s='/content/drive/MyDrive/Meld/MELD.Raw/train_sent_emo.csv', categories=classs,
limit=930, transform=FaceTransform((48,48)))
train_dataset.make_shuffle()
len(train_dataset)
```

```
{'Chandler/result_dia2_utt0.0.png': 'neutral', 'Monica/result_dia2_utt5.0.png': 'neutral'
3098
```

```
test_dataset = Meld(roots='/content/drive/MyDrive/Meld/MELD.Raw/Test cropped images/',
csv_s='/content/drive/MyDrive/Meld/MELD.Raw/test_sent_emo_final.csv',
categories=classs, limit=146, transform=FaceTransform((48,48)))
test_dataset.make_shuffle()
len(test_dataset)
```

```
{'Chandler/result_dia156_utt5.0.png': 'neutral', 'Chandler/result_dia156_utt5.1.png': 'neutral'
1203
```

Figura 6.31: Conjunto de Train y Test Modalidad Facial en Meld

Se realizó el entrenamiento de la modalidad facial a partir del anexo A.29. Para tener una mejor referencia de cómo se van comportando tanto el conjunto de train y test se realizan dos gráficos para representar el número de épocas vs la función de pérdida y el siguiente el número de épocas vs su exactitud. En la figura 6.32 se puede apreciar que en la época 10 empieza a tener un sobreajuste el modelo entrenado debido a la separación de actores manualmente, por lo que existe un problema de clasificación en el preprocesamiento y también debido a las cantidades de imágenes a entrenar dentro del dataset creado personalmente durante el preprocesamiento.

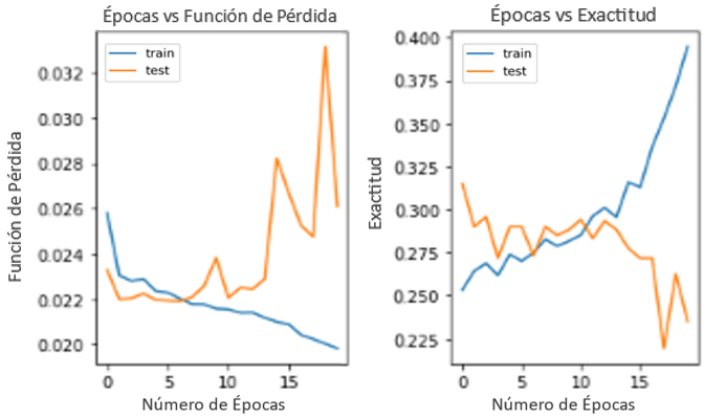


Figura 6.32: Comportamiento Train y Test Modalidad Facial en Meld

En la figura 6.33 se representa la data, etiqueta y la salida del modelo para calcular el promedio del conjunto de test con respecto a las métricas de la exactitud. Por otro lado, se va guardando la lista de emociones predichas y las emociones reales para realizar un clasification report como en todos los dataset trabajados.

```
from sklearn.metrics import classification_report
Tacacc = 0; Tcont = 0
yt,yp=[],[]
with torch.no_grad():
    for sample in tqdm(test_dataloader):
        data = sample['data'].float().cuda()
        label = sample['label'].flatten()
        out = model(data).to('cpu')
        yt += label.tolist()
        yp += torch.argmax(out, dim=1).tolist()
        Tacacc += (torch.argmax(out, dim=1) == label).float().sum()
        Tcont += data.shape[0]
print('TT test_acc: %.4f'%(Tacacc/Tcont))
```

```
100% 19/19 [00:14<00:00, 3.92it/s]
TT test_acc: 0.2594
```

Figura 6.33: Promedio de la exactitud en el Conjunto de Test en Meld

Luego se trabajó con la modalidad de audio realizando un preprocesamiento, donde cabe destacar que la única variación con los demás dataset es que en este caso se trabajó con un csv que contiene la mayor cantidad de información. Por lo que los videos originales del dataset con extensión .avi se van a convertir directamente en espectrogramas ocupando el mismo método como se muestra en las figuras 6.12 y 6.13 para tratarlos de mejor manera para la adecuación de la entrada de datos en la arquitectura. En el anexo A.30 se encuentra la clase personalizada del dataset para la modalidad de audio.

En la siguiente figura 6.34 se tiene la cantidad de datos por parte del entrenamiento en la cual se recorrió el csv correspondiente con las etiquetas establecidas en el diccionario y aplicando el balanceo a la emoción neutral debido a que tiene una mayor cantidad de ejemplos dentro del dataset en comparación a las demás.

```
Audio_trainset = AudioDataset(roots='/content/mel/Train',
csv_s='/content/drive/MyDrive/Meld/MELD.Raw/train_sent_emo.csv', categories=classs, limit=2000,
transform=AudioTransform((259,128)))
Audio_trainset.make_shuffle()
len(Audio_trainset)
```

```
{'neutral': 0, 'joy': 1, 'anger': 2, 'surprise': 3}
{'/content/mel/Train/dia0_utt0.mp4.npy': 'neutral', '/content/mel/Train/dia0_utt1.mp4.npy': 'neutral'
['/content/mel/Train/dia0_utt0.mp4.npy', '/content/mel/Train/dia0_utt1.mp4.npy', '/content/mel/Train/
6058
```

Figura 6.34: Conjunto entrenamiento Meld

A partir de la figura 6.35 se toma en cuenta el conjunto de pruebas donde se realizó la misma estrategia a partir de su csv en particular donde se llama a la clase del dataset con los diferentes parámetros para recorrer las columnas del csv y en donde se realizó un balanceo de menor escala para la emoción neutral debido a que la cantidad de datos en el conjunto de pruebas es menor que en la del entrenamiento.

```
Audio_testset = AudioDataset(roots='/content/mel/Test',
csv_s='/content/drive/MyDrive/Meld/MELD.Raw/test_sent_emo.csv', categories=classs,
limit=400, transform=AudioTransform((259,128)))
Audio_testset.make_shuffle()
len(Audio_testset)
```

```
{'neutral': 0, 'joy': 1, 'anger': 2, 'surprise': 3}
{'/content/mel/Test/dia0_utt0.mp4.npy': 'surprise', '/content/mel/Test/dia0_utt1.mp4.npy': 'anger'
['/content/mel/Test/dia0_utt0.mp4.npy', '/content/mel/Test/dia0_utt1.mp4.npy', '/content/mel/Test/
1429
```

Figura 6.35: Conjunto pruebas Meld

El siguiente paso fue entrenar el conjunto de train y test de la misma forma en como se muestra en las figuras 6.28 y A.16 con la finalidad de obtener las métricas de interés en esta modalidad con el F1 score, exactitud, exhaustividad.

En la figura 6.36 se representa el comportamiento del conjunto de train y test al momento de ser entrenada en las 20 épocas bajo el modelo de audio, tomando en cuenta sus métricas correspondientes. Que permite decir que el conjunto de train tiene mejor rendimiento dentro del modelo para predecir dichas emociones, contiene una mejor exactitud y una función de pérdida baja, por lo que se puede decir que el conjunto de train está mejor capacitado para acertar las emociones utilizadas dentro del entrenamiento y está compuesto por una data de mejor calidad que el de pruebas.

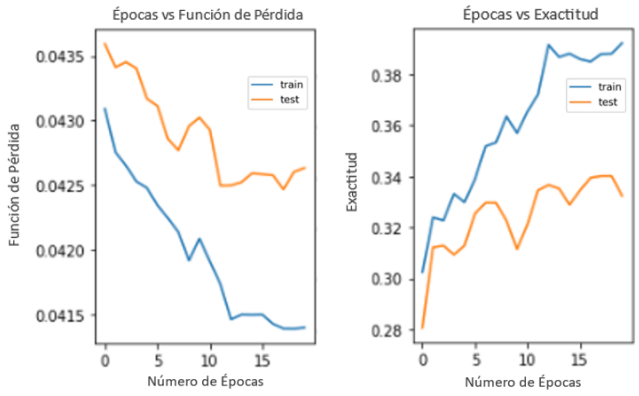


Figura 6.36: Comportamiento Train y Test Modalidad Audio en Meld

En primera instancia para la modalidad de texto se empieza creando la clase personalizada del dataset la cual se encuentran en el anexo A.31. En la figura 6.37 se llevó a cabo un llamado a la clase del dataset con los parámetros que son el csv como tal, las emociones a utilizar almacenadas en un diccionario y el balanceo de la emoción neutral. Todo esto guardándolo en una variable para verificar la cantidad del conjunto de train y test que están etiquetados y balanceados a los cuales se llevaran al entrenamiento dentro de la arquitectura.

```
Text_trainset = TextDataset(csv_s='/content/drive/MyDrive/Meld/MELD.Raw/train_sent_emo.csv', categories=classs, limit=2000)
Text_trainset.make_shuffle()
len(Text_trainset)

{'dia0_utt0': ('also I was the point person on my company\x92s transition from the KL-5 to GR-6 system.', 'neutral'), 'dia0
6058

Text_testset = TextDataset(csv_s='/content/drive/MyDrive/Meld/MELD.Raw/test_sent_emo.csv', categories=classs, limit=400)
Text_testset.make_shuffle()
len(Text_testset)

{'dia0_utt0': ('Why do all you\x92re coffee mugs have numbers on the bottom?', 'surprise'), 'dia0_utt1': ('Oh. That\x92s
1429
```

Figura 6.37: Conjunto de Train y Test en Meld

A partir de la figura 6.38 se realizó una función que va a recorrer data que viene siendo el texto como tal y las etiquetas para alojarlas dentro de un dataframe y luego convertirlo a un archivo csv para trabajarlo de mejor manera en el entrenamiento.

```python
def extract_audio3(set):
    textos=[]
    labels=[]

    for element in set:
      texto=element['data']
      label=element['label']
      textos.append(texto)
      labels.append(label)
    return pd.DataFrame({'text':textos, 'target':labels})

T=extract_audio3(Text_trainset)
T.to_csv('text_train_meld.csv',index=False)
```

Figura 6.38: Función Dataframe en Meld

Al tener el dataframe con las columnas de interés como son el texto y las emociones o target en un formato de índice, al cual se procede a convertir en un archivo csv de forma particular para el conjunto de entrenamiento y de pruebas, como se muestra en la figura 6.39

	text	target
0	Okay, in we go.	1
1	And I mean-I□m having a lot of fun.	1
2	Yeah, you made me feel really guilty about goi...	2
3	He hooked up! He hooked up with someone.	3
4	Wow! That was good. That was... Tweezers?	3
...	...	...
5396	Like-like, hand modeling!	1
5397	Ok, I'm sensing that this is some kind of word...	0
5398	I can't believe you put that on my alumni page!	2
5399	You mean with Casey.	0
5400	I rode a bike!	1

5401 rows × 2 columns

Figura 6.39: Dataframe Train en Meld

El procedimiento del entrenamiento es exactamente el mismo que en el dataset afew dado a conocer en la figura 6.21 en donde a partir de los 2 csv que contienen el texto y las etiquetas del conjunto de train y de test se pasan por el modelo para codificar la entrada de datos específicamente el texto y luego se ingresan por una MLP donde se van a entrenar el modelo codificado con el texto y las emociones para ir almacenando las predicciones de ambos conjuntos de datos y obtener las métricas como el F1 score y la exactitud para verificar el comportamiento del conjunto de datos a partir de esta modalidad individual. Estas métricas de evaluación se van a formalizar en la tabla 7.6.

En el anexo A.32 se muestra la clase personalizada del dataset para el método de fusión. En la figura 6.40 se realiza una llamada a la clase del dataset con sus respectivos parámetros para ir verificando la cantidad de data para el conjunto de train y test que va a hacer entrenada.

```
train_dataset = DatasetMELD(Classes, face_data, audi_data, text_data, csv_s='/content/drive/MyDrive/Meld/MELD.Raw/train_sent_emo.csv',
                            transform=FusionTransformer(''))
test_dataset = DatasetMELD(Classes, face_data, audi_data, text_data1, csv_s='/content/drive/MyDrive/Meld/MELD.Raw/test_sent_emo.csv',
                            transform=FusionTransformer(''))

len(train_dataset), len(test_dataset)
```

```
(6058, 1946)
```

Figura 6.40: Conjunto de Train y Test Multimodal Meld

Para tener una representación más concreta de las métricas se lleva a cabo las mismas gráficas en la figura 6.41 que en casos anteriores donde la diferencia se encuentra que ahora están todas las modalidades juntas y se interpreta que el conjunto de test está sufriendo un sobreajuste en el modelo debido a que la función de pérdida está aumentando por la baja calidad de los datos y la poca cantidad de ejemplares por cada emoción en las distintas modalidades. Lo cual da a decir que la eficiencia del modelo es baja en relación con las predicciones realizadas y los valores reales durante el aprendizaje. El conjunto de train tiene un promedio de la exactitud en 0,81 y test un 0,53.

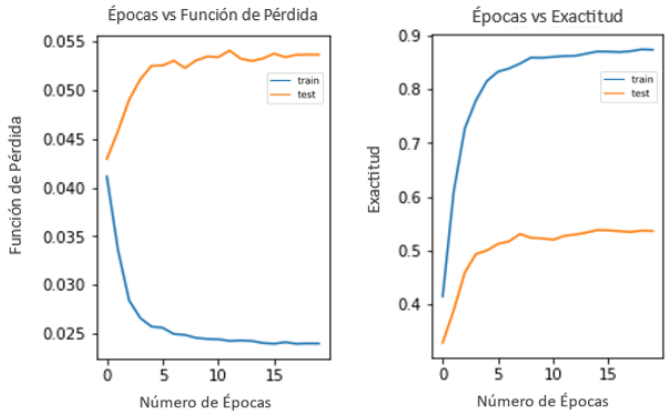


Figura 6.41: Comportamiento Train y Test Multimodal Meld

Capítulo 7

Pruebas y Resultados

En este capítulo se da importancia a las diversas pruebas unimodales y multimodales dentro de cada dataset para realizar un análisis de los resultados de forma detallada a través de tablas comparativas, como también verificar si los dataset cumplen o no los criterios de evaluación y una discusión de los resultados con la finalidad de resolver las preguntas de investigación planteadas.

7.1. Evaluaciones Unimodales y Multimodales

Lo primero es que se da a conocer el resumen de las 3 modalidades presentes en el dataset Iemocap con sus evaluaciones unimodales a partir de las métricas F1 score y su exactitud con sus emociones categorizadas para compararlo con los otros conjuntos de datos a trabajar. Cabe recordar que este dataset controlado fue trabajado por un estudiante de la Universidad Católica San Pablo del Perú [3] realizando el entrenamiento y obteniendo estas métricas que se aprecian en las tablas 7.1 y 7.2.

Emociónes	Modalidad Facial		Modalidad de Audio		Modalidad de texto	
	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)
Happy	70,0	66,8	44,0	32,1	85,0	84,9
Neutral	23,0	16,2	55,0	66,1	78,0	82,4
Sad	31,0	21,7	62,0	69,8	86,0	81,7
Angry	32,0	71,4	64,0	65,3	89,0	84,5
Promedio	39,0	44,0	56,2	58,3	84,5	83,5

Tabla 7.1: Unimodal Iemocap [3]

Para realizar un enfoque multimodal se realizaron varias evaluaciones con las respectivas modalidades en conjunto o con ausencia de alguna en especial con el fin de aplicar las métricas de desempeño como el F1 score que permite combinar en un solo valor tanto las cantidades que el modelo es capaz de identificar como la calidad del modelo a la hora de

clasificar dichas emociones y, por otro lado, se tiene la exactitud que va a ayudar a detectar todas las predicciones acertadas por el modelo representado en porcentaje. A continuación en la tabla 7.2 se presentan lo mencionado anteriormente con las emociones trabajadas en el dataset Iemocap donde previamente están entrenadas, como también las combinaciones de modalidades con sus métricas y el promedio de cada una de ellas.

Emociónes	F+A+T		T+A		T+F		F+A	
	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)
Happy	82,0	80,1	80,0	79,4	84,0	85,6	55,0	41,9
Neutral	75,0	79,2	73,0	77,9	75,0	76,5	58,0	69,0
Sad	78,0	76,3	78,0	78,0	80,0	73,5	62,0	64,1
Angry	82,0	74,7	81,0	76,5	76,0	78,3	65,0	64,7
Promedio	79,0	77,5	78,0	77,9	78,0	78,4	60,0	59,9

Tabla 7.2: Evaluación Multimodal Iemocap [3]

Cabe destacar que las métricas para evaluar el rendimiento de algún modelo en los dataset controlados como el Iemocap en este caso y en la literatura en general son mejores que en los dataset no controlados, ya que en ambiente real (no controlado) existen factores externos como el ruido o silencios, la calidad de la imagen por consecuencia de la poca o mucha luminosidad que va afectando a la hora de obtener las diferentes emociones por cada modalidad, en cambio, en un ambiente de laboratorio (controlado) todos estos factores no van a influir, ya que de alguna manera se evitan con la finalidad de obtener mejores métricas para el reconocimiento de emociones.

Lo siguiente fue trabajar con el dataset Sfew a la cual se le realizó una evaluación unimodal a la modalidad de rostro como se da a conocer en la tabla 7.3. Que es la única modalidad que se encuentra dentro de este conjunto de datos, por lo que se puede comparar con respecto a sus imágenes categorizadas.

Emociónes	Modalidad Facial		Modalidad de Audio		Modalidad de texto	
	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)
Happy	28,0	32,4	x	x	x	x
Neutral	38,0	49,3	x	x	x	x
Sad	12,0	6,9	x	x	x	x
Angry	33,0	32,8	x	x	x	x
Promedio	27,7	30,3	x	x	x	x

Tabla 7.3: Evaluación Unimodal facial (Sfew)

Como el dataset Sfew no contiene más modalidades, se procedió a trabajar con el dataset Afew para realizar el mismo procedimiento que consiste en evaluaciones unimodales para rostro, audio y texto con sus emociones trabajadas y métricas de desempeño como se muestra en la tabla 7.4

Emociónes	Modalidad Facial		Modalidad de Audio		Modalidad de texto	
	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)
Happy	25,0	24,5	51,0	44,0	37,0	35,4
Neutral	23,0	19,0	34,0	34,9	32,0	31,3
Sad	24,0	23,6	36,0	38,7	40,0	55,0
Angry	48,0	56,7	32,0	33,9	6,0	3,1
Promedio	30,0	30,9	38,2	37,8	28,7	31,2

Tabla 7.4: Evaluación Unimodal Afew

Luego se realizó el enfoque multimodal en la tabla 7.5 donde se consideran las mismas emociones obteniendo las métricas de desempeño a partir de las 3 modalidades juntas que ingresan de forma individual por el método del embracenet+ y luego se fusionan.

Emociónes	F+A+T		T+A		T+F		F+A	
	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)
Happy	45,0	49,1	54,0	55,9	36,0	38,9	47,0	52,5
Neutral	35,0	36,5	35,0	36,5	33,0	34,9	35,0	39,6
Sad	33,0	29,0	35,0	37,0	25,0	20,9	22,0	17,7
Angry	37,0	35,8	22,0	18,8	39,0	39,6	33,0	32,0
Promedio	37,5	37,6	36,5	37,0	33,2	33,5	34,2	35,4

Tabla 7.5: Evaluación Multimodal Afew

Para finalizar en la tabla 7.6 se tiene el dataset Meld que contiene 3 modalidades, por tanto, se realiza las evaluaciones unimodales entrenando la data previamente y categorizando las emociones trabajadas con la finalidad de obtener las métricas de desempeño que permite detectar la calidad del modelo y el porcentaje de aciertos a la hora de predecir las emociones.

Emociónes	Modalidad Facial		Modalidad de Audio		Modalidad de texto	
	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)
Neutral	29,0	24,9	43,0	57,3	66,0	72,0
joy	15,0	11,2	37,0	41,7	64,0	73,3
Angry	30,0	36,5	32,0	23,1	42,0	32,2
Surprise	26,0	33,6	16,0	10,3	59,0	58,2
Promedio	25,0	26,5	32,0	33,1	57,7	58,9

Tabla 7.6: Evaluación Unimodal Meld

En la tabla 7.7 se realizó el enfoque multimodal con las 4 evaluaciones posibles, enfocándose en las emociones trabajadas con la finalidad de obtener las métricas de desempeño tanto de F1 score y la exactitud.

Emociónes	F+A+T		T+A		T+F		F+A	
	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)
Neutral	61,0	50,7	63,0	53,5	63,0	54,4	41,0	32,4
Joy	53,0	70,8	55,0	71,6	56,0	73,1	24,0	34,7
Anger	38,0	35,3	38,0	34,4	37,0	33,0	19,0	17,9
Surprise	49,0	57,6	51,0	58,0	51,0	58,7	17,0	22,0
Promedio	50,2	53,6	51,7	54,3	51,7	54,8	25,2	26,7

Tabla 7.7: Evaluación Multimodal Meld

7.2. Análisis de Resultados

En esta sección se explican en detalle los resultados de cada dataset por medio de las evaluaciones unimodales y multimodales que se concretan a través de las métricas de evaluación importantes con las que se trabajaron y se vieron reflejadas en las tablas anteriormente como son la exactitud y el F1 score.

7.2.1. Iemocap

Particularmente para el dataset iemocap en sus evaluaciones unimodales, como se muestra en la tabla 7.1 a modo resumen, lo principal a destacar es que la modalidad de texto se lleva los mejores resultados en cuanto al porcentaje de aciertos a dicha predicción con un promedio de 83,5 % con base en todas las emociones analizadas. En cambio, para la modalidad de rostro existe un mayor desequilibrio con respecto al F1 score y la exactitud donde van variando los resultados por cada emoción de una forma exagerada, lo cual deja en evidencia que su promedio de precisión es muy bajo para predecir emociones con su modelo VGG19. Por último, está la modalidad de audio donde la emoción con mejores métricas se la lleva la tristeza junto con el enojo, pero a diferencia con la modalidad de rostro contiene un promedio de mejor precisión y menos aleatoriedad en los datos debido a la calidad de los audios y al modelo en el que se trabajó.

La evaluación multimodal que se realizó en el dataset Iemocap es exactamente la misma para todos los dataset donde se utiliza el embracenet+ para fusionar las modalidades que a su vez detecta la ausencia de alguna modalidad e ir ajustando las probabilidades. La primera evaluación es la del F+A+T donde se puede destacar que es la que contiene mejor resultados a la hora de predecir las emociones debido a que están presentes todas las modalidades y permiten complementarse unas con otras para ir obteniendo la predicción con las mejores métricas ponderadas.

En la evaluación de texto+audio se presenta el reporte de las métricas de evaluación con sus emociones categorizadas, en donde la precisión representa la calidad del modelo para detectar predicciones positivas y la exactitud del modelo con la intención de encontrar el porcentaje de aciertos que tiene un promedio del 78,0 % y una exactitud del 77,9 %.

En la evaluación de texto+rostro contiene un mejor rendimiento a partir de los datos presentados en las métricas de evaluación donde presentan valores en su gran mayoría sobre el 70,0 % por lo que significa que el modelo predice en estas modalidades de una forma correcta y tiene una precisión considerable en todas sus emociones.

En la modalidad de rostro+audio no es la que tiene los mejores datos con respecto a los casos anteriores, ya que las métricas de evaluación, como el promedio de la precisión que demuestra la calidad del modelo a la hora de predecir las emociones, se encuentra en

un 62,0 % lo cual nos permite decir que el modelo se está equivocando un 38,0 % de las veces al momento de predecir las emociones y el 62,0 % acertadamente. Por otro lado, la exhaustividad contiene un promedio del 59,0 % que representa la cantidad de emociones correctas que el modelo es capaz de identificar y por último el promedio de la exactitud que se encuentran en un 59,9 % donde demuestra que el modelo ha acertado casi un 60,0 % de las veces para predecir las emociones establecidas.

7.2.2. Sfew

En el dataset Sfew se puede apreciar en la figura 6.3 que no contiene una data amplia para su única modalidad, ya sea para el conjunto de entrenamiento y para las pruebas. Lo cual afectara directamente a la calidad del entrenamiento que se realizó dentro de la arquitectura. Donde se procedió a iterar en 20 épocas de una forma estándar para mejorar las métricas de desempeño dentro del entrenamiento, como se puede apreciar en la figura A.5. Las cuales no mejoraron bastante y el promedio de la exactitud de los datos fue de 30,3 % lo que significa que es un modelo impreciso para reconocer emociones de forma facial.

En la evaluación unimodal se representa las métricas correspondientes al entrenamiento realizado de una forma más resumida en donde se presenta por cada emoción, en particular el F1 score y la exactitud. La emoción con mayor desempeño tanto para predecir los casos correctamente identificados por el modelo es neutralidad con un 49,3 % de exactitud y un 38,0 % de F1 score. Por otro lado, la emoción, tristeza es la más baja e imprecisa, lo cual trae como consecuencia que el promedio de la exactitud sea muy bajo como se muestra en la figura 7.3.

7.2.3. Afew

En cambio, al analizar los resultados del dataset Afew que contiene las 3 modalidades se obtiene más información al respecto, es decir, existe una mayor cantidad de datos para entrenar y comparar sus resultados con otros datasets. Para la modalidad de rostro, como se muestra en la figura 6.7 el conjunto de train y de test es mucho más extenso debido a que cada video tiene una cierta cantidad de frames (imágenes estáticas) por lo que la entrada a la arquitectura y su entrenamiento será de un tiempo más prolongado. Al iterar 20 épocas en el entrenamiento, las métricas de desempeño son muchos mejores que las del dataset anterior debido a la gran cantidad de imágenes que se van procesando y el preprocesamiento previo dentro de los conjuntos de datos. Los resultados que se presentan en la figura 7.4 muestran las 4 emociones de forma individual con sus 2 métricas de evaluación, las cuales resultan en la emoción angry que tiene una exactitud del 56,7 % lo cual implica decir que el modelo está identificando las emociones en un casi 57,0 % de forma correcta y el otro 43,0 % serían predicciones erróneas por parte del modelo. Estos resultados son concordantes con

respecto a lo previsto, ya que las métricas al paso del entrenamiento fueron mejorando con base en la función de pérdida, donde el modelo fue reconociendo las imágenes disponibles ya sea en la carpeta train o test y se fueron categorizando dependiendo de su etiqueta, pero al reducir la cantidad de videos originales del dataset debido a la duplicación del mismo, afectó considerablemente, lo que permite decir que es un dataset poco viable para el reconocimiento facial de emociones con grandes cantidades de datos al menos para las 4 emociones estándar.

En la modalidad de audio se encuentra el reporte del conjunto de pruebas que contiene 237 ejemplos con un promedio de la exactitud de 37,8 % lo cual es un porcentaje apto para identificar emociones dentro del entrenamiento como se muestra en la figura A.5. La valoración de la exactitud y de las métricas en general se basa bajo la comparación con otro dataset de la competencia, como se muestra en la figura 7.8 y simultáneamente con los dataset trabajados como el Sfew y el Meld bajo ambientes no controlados. La emoción happy es la que contiene el mayor F1 score de todas las emociones con un 51,0 % el cual representa la calidad del modelo trabajado, como también su métrica de exhaustividad es la más alta entre las demás emociones, lo que permite deducir que es un modelo capaz para identificar las emociones en la modalidad de audio.

En la modalidad de texto se realizó el mismo procedimiento que las otras modalidades para tener un resumen en relación con el desempeño que lleva a cabo el modelo con sus archivos de texto al ser entrenadas. Los resultados van a depender del audio, ya que los textos que se fueron procesando dentro de la arquitectura fueron extraídos de los archivos .avi en donde lo importante es la calidad, la cantidad de ejemplos y la capacidad del modelo para predecir ciertas emociones. Para ello se lleva a cabo el informe de clasificación donde se deja como parámetro las etiquetas de las emociones y las predicciones de las mismas. Al tener muy baja cantidad de ejemplos para entrenar debido a que la extracción de audios a texto, se encontraron en muchos casos que no existía un audio legible de calidad para llevarlo al texto, por lo que se omitieron para entrenarlas. Lo importante de este reporte es que la exactitud promedio es de 31,2 % en las 4 emociones, es decir, es relativamente bajo con base en comparaciones con otros dataset de la competencia como el Aff-wild2 que se muestra en la figura 7.8. Pero cabe mencionar que el modelo se destaca a la hora de encontrar la emoción sad que contiene un 55,0 % de exactitud, permite decir que el modelo está acertando de mejor manera tal emoción y que es un modelo estable con una capacidad para predecir emociones de manera promedio considerando que es un dataset de origen no controlado y todos los problemas asociados con la cantidad de archivos y la calidad de los mismos.

Es por ello que las métricas como la exactitud, la precisión y la exhaustividad son buenas formas de evaluar los modelos de clasificación para conjuntos de datos equilibrados.

Por otro lado, se tiene el enfoque multimodal de este dataset donde se aprecia que en las 4 evaluaciones el F1 score promedio y la exactitud se mantiene en un rango estable

que por lo general no son muy buenos resultados para predecir ciertas emociones. La mejor evaluación que se puede destacar es en donde se encuentra las 3 modalidades juntas con un 37,5 % de F1 score y una exactitud de 37,6 % y la peor es la evaluación de texto+rostro que contiene un 33,2 % de F1 score y un 33,5 % en su exactitud para detectar estas emociones.

7.2.4. Meld

Este dataset contiene 3 modalidades de las cuales se analizó la modalidad de rostro, audio y de texto. Al pasar este conjunto de datos por las diferentes etapas de análisis descriptivo, preprocesamiento, adecuar la entrada de datos a la arquitectura, entrenamiento y obtener las diversas métricas, se puede apreciar en la siguiente tabla 7.6 que las evaluaciones unimodales que se realizaron por cada modalidad con las emociones estándar existe una mejora desde la primera a la última con respecto al momento de predecir las emociones.

En la modalidad de rostro se encuentra un 26,5 % por parte de la exactitud y un 25,0 % en el F1 score donde son métricas bajas debido a problemas con el etiquetado manual de los actores principales y extras en donde existe error humano al separar dichas imágenes que posteriormente fueron entrenadas. Debido a que el dataset original no contenía una identificación de sus actores y los videos en los csv se presentaban por sus id como por sus capítulos, temporadas y actores que realizaban cierta conversación, pero en dichos videos se encontraba mucha gente en los alrededores que actuaban como extras sin etiquetar.

Se puede destacar que en la modalidad de audio se utilizaron los videos originales del dataset en su totalidad para llevar a cabo el entrenamiento, el cual no se desarrolló de la mejor manera debido a ciertos ruidos leves adentro de los videos originales, ya que está manejando en un ambiente no controlado donde existen factores externos que perjudican las métricas de cierta manera y algunos audios no son legibles para poder ser entrenados. Permitiendo mostrar un F1 score de 32,0 % y una exactitud promedio de 33,1 % a lo largo del entrenamiento.

En cambio, la modalidad de texto al tener las conversaciones de cada actor en los respectivos csv se permite identificar de una forma más legible, por lo tanto, contiene métricas de mejor rendimiento en comparación al dataset anterior, es decir, el modelo está aprendiendo de una forma regular con un promedio de F1 score 57,7 % y una exactitud del 58,9 %. Por último, el enfoque multimodal en dicho dataset se encuentra en la tabla 7.7 donde la evaluación mejor catalogada al pasar por el embracenet+ para mejorar el rendimiento de las mismas es el texto+rostro con un 51,7 % de F1 score y un 54.8 % de exactitud.

7.2.5. Resultados de la Competencia

En la siguiente figura 7.8 se llevó a cabo una comparación de los resultados obtenidos en el dataset Meld y los de la competencia que se encuentra en un ambiente no controlado llamado Aff-wild2 [71]. Con la finalidad de saber si nuestras métricas están a la altura de alcanzar predicciones competitivas para algunas categorías de emociones en ambientes no controlados. Donde se puede destacar que en general las métricas son comparables por cada emoción y modalidad, las cuales están en un rango similar basándose en la calidad de los modelos y la capacidad para identificar las emociones. Sobre todo cada dataset tiene sus complicaciones, como es en el caso del Meld en la parte facial con la etiquetación de los actores y de su separación manual ya mencionada anteriormente, como también el Aff-wild2 presenta problemas en sus métricas en las modalidades de audio y texto debido a que la mayor cantidad de datos son videos de personas reaccionando a películas y que están etiquetados, pero no para las demás modalidades como audio y texto. Es posible que el segmento de audio no coincida con la emoción etiquetada presentada, ya que estos fueron anotados por sus expresiones faciales.

Se encuentran muchos audios que no corresponden a la emoción que aparece anotada para rostros. Lo mismo con interacciones con otras personas o los videos de reacciones que no coinciden con la emoción que se ve en el rostro. Por otro lado, el texto, es porque como el audio también no está bien filtrado por su forma de adquisición, entonces cualquier modelo de extracción de habla falla y se requiere transcribir.

Dataset	Meld						AFF-wild2					
Emociónes	Rostro		Audio		Texto		Rostro		Audio		Texto	
	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)
Neutral	29,0	24,9	43,0	57,3	66,0	72,0	70,0	74,3	63,0	60,2	75,0	96,9
Happy	15,0	11,2	37,0	41,7	64,0	73,3	55,0	60,4	30,0	38,8	12,0	6,9
Angry	30,0	36,5	32,0	23,1	42,0	32,2	9,0	6,2	40,0	3,4	0,0	0,0
Surprise	26,0	33,6	16,0	10,3	59,0	58,2	32,0	40,8	23,0	21,4	0,0	0,0
Promedio	25,0	26,5	32,0	33,1	57,7	58,9	41,5	45,4	39,0	30,9	21,7	25,9

Tabla 7.8: Tabla comparativa entre el Meld y el Aff-wild2

7.3. Criterios de Evaluación

En esta sección la finalidad es verificar el dataset que cumpla con los criterios de evaluación ya mencionados en el capítulo 3, se muestra a continuación una tabla a modo resumen con el fin de dar cuenta que dataset es el más adecuado para trabajar en un contexto de robótica social basándose en los criterios previamente establecidos. Paralelamente, esta información va a ayudar a responder las preguntas de investigación que necesariamente son importantes para la discusión de resultados.

	Iemocap	Sfew	Afew	Meld	Aff-Wild2
Cantidad de Modalidades	✓	✗	✓	✗	✓
Emociones Estandar	✓	✓	✓	✗	✓
Enfoque Multimodal	✓	✗	✓	✓	✓
Datos Balanceados	✗	✓	✓	✗	✗
Ambiente No controlado	✗	✓	✓	✓	✓
Calidad de train y test	✓	✗	✗	✗	✗
Métricas de desempeño	✓	✗	✗	✓	✗

Figura 7.1: Criterios de Evaluación

7.4. Discusión de Resultados

¿Cuáles son los dataset que se adaptan mejor a la arquitectura ya existente según sus resultados?

El dataset Meld es el que se adapta de mejor manera a la arquitectura, donde contiene las mejores métricas de desempeño, ya sea en las evaluaciones unimodales como en las evaluaciones multimodales, comparándolo con el dataset Sfew, Afew y Aff-Wild2. Debido a que la clave se encuentra en las transcripciones de audio a texto, donde no existieron problemas en perdidas de datos importantes, ya que previamente han sido pre procesadas por profesionales y están en bruto dentro de los csv como etiquetadas. En donde se obtienen por cada modalidad un F1 score que va a representar el rendimiento del modelo acorde a la cantidad de emociones que logra identificar como también la calidad del modelo a la hora de predecir las emociones en donde va oscilando en un rango del 25,0 % al 57,7 %. Por otro lado, se encuentra la exactitud que nos presenta el porcentaje de casos que el modelo ha acertado con respecto a las predicciones de las emociones, las cuales oscilan entre el 26,5 % y el 58,9 %.

¿Se pueden incorporar mejoras a los dataset y a la arquitectura?

Para el dataset Iemocap es recomendable utilizar diferentes modelos para la modalidad de rostro y de audio, ya que son las más afectadas al momento de realizar el entrenamiento de los datos.

El dataset Sfew se deberían incorporar una mayor cantidad de datos para la modalidad de rostro y obtener la modalidad de audio y de texto para poder aplicar un enfoque multimodal, con el fin de entrenar varias modalidades y llevar a cabo métricas que correspondan a la realidad. En otras palabras, independizarse del dataset Afew y crear su propia

data en un ambiente no controlado con diferentes modalidades y modelos que permitan entrenar de la mejor manera los datos.

El dataset Afew con sus archivos originales en el conjunto de test deberían estar categorizados para evitar traer complicaciones al momento de realizar el preprocesamiento de los datos, también se debería mejorar en cuanto a la arquitectura y al entrenamiento de los datos, es en la modalidad de texto, particularmente en el funcionamiento del modelo, ya que es justamente donde debería tener mayor rendimiento el entrenamiento en cuanto a las métricas de desempeño según varias fuentes en esta área. Con la finalidad de competir con otros dataset en el contexto de robótica social y probando diversos modelos en la modalidad de texto que se adapten de mejor manera a la arquitectura.

En el dataset Meld se necesita mayor información acerca de trabajos relacionados con la modalidad facial o un preprocesamiento de base con las imágenes categorizadas para ser más fácil el reconocimiento facial de la persona que está realizando tal emoción. Como también en los conjuntos de entrenamiento y de pruebas, aplicar un balanceo previamente en el dataset original para evitar tener grandes cantidades de emociones, por una parte, y muy pocas por otra.

¿Cuál dataset cumple de mejor manera los requisitos establecidos?

El dataset Afew es el único que cumple con la mayoría de los requisitos que se pueden apreciar en la figura 7.1, donde los datos estén balanceados, se encuentren todas las modalidades como también contenga las emociones estándar, se trabaje en un ambiente no controlado, se lleve a cabo enfoque unimodal y multimodal. Pero en lo único que falla es en la calidad del conjunto de train y test lo que causa problemas para las métricas de desempeño.

¿Cuál es la diferencia relevante entre los distintos dataset?

Según el documento [3] donde se trabajó con el dataset Iemocap es realizado en un ambiente controlado de laboratorio en las cuales no existe problemas con la calidad de la imagen, tampoco existe ruido en los audios y los csv se van a dividir en 4 para el conjunto de train y 1 para el de test los cuales contienen el nombre de los videos, las emociones y los valores de dominancia, activación y valencia. Por consecuencia, al entrenar los diferentes conjuntos de datos a través de los modelos se puede apreciar que las métricas son mejores para predecir e identificar las emociones representadas.

El dataset Sfew solo contiene 1 modalidad que es la de rostro, por lo que no es necesario obtener ciertas evaluaciones multimodales dentro del método embracenet+. En donde las imágenes fueron extraídas del dataset Afew en diferentes frames a lo que depende de este dataset en particular, es decir, actúa como una base de datos de expresiones faciales del dataset Afew sin restricciones como poses de cabezas variadas, amplio rango de edad y poca luminosidad.

El dataset Afew es más completo no en cuanto a obtener los mejores resultados, sino en los requisitos establecidos para llevarlo a un contexto de robótica social, ya que en sus diferentes evaluaciones unimodales y en el enfoque multimodal se dan a conocer métricas que son más o menos aptas para este contexto, a lo cual cabe destacar que es un dataset no controlado a partir de videos de actores de diferentes películas que están categorizados por sus emociones percibidas que facilitan el trabajo del mismo.

En el dataset Meld la gran diferencia es que las emociones están etiquetadas dentro de un csv específico para el conjunto de entrenamiento, como también se encuentran los textos asociados a las personas que participan en los videos originales y otro csv para el conjunto de pruebas donde el nombre de cada video tiene un formato determinado con el id del diálogo y el id del texto propiamente tal.

¿Qué emoción es la más destacada por los distintos casos a trabajar?

En las evaluaciones unimodales por parte del dataset Iemocap la emoción angry en la modalidad de texto es considerablemente la que se adaptó de mejor manera a la arquitectura con su modelo para predecir la emoción y evitar tener tantos archivos perdidos con un F1 score de 85,0 % y una exactitud del 84,9 %

En las evaluaciones multimodales, en el caso de texto+rostro se aprecia que la emoción más destacada es la de felicidad con una exactitud del 85,6 % y un F1 score de 84,0 % por lo cual tiene valores muy sólidos que permiten decir que el método de fusión está funcionando correctamente para predecir las emociones con una exhaustividad en grandes cantidades y con una precisión que otorga buenas sensaciones. Por otro lado, se puede decir que la modalidad de texto es sumamente importante para el dataset para reconocer las emociones a partir de la ausencia del audio y con la ayuda del rostro de cierta manera.

En el dataset Sfew se puede destacar la emoción de neutralidad que contiene una exactitud en su modelo de 49,3 % lo cual es un valor muy por sobre la media esperada y un F1 score de 38,0 %. Por lo general se encuentran valores muy poco competitivos a diferencia de los demás conjuntos de datos por diversas razones, ya sea la calidad del modelo a utilizar, la forma en que se están procesando los datos en la arquitectura, como también las cantidades de ejemplos a utilizar en la red neuronal.

En Afew se encuentra la emoción de angry en el caso de la modalidad facial, con unas métricas correspondientes a 56,7 % en exactitud y un F1 score de 48,0 % que vendrían siendo el mejor caso posible dentro del entrenamiento de un dataset.

En su evaluación multimodal se puede destacar como la emoción más influyente es happy con un 54,0 % de F1 score y una exactitud del 55,9 % en la fusión de modalidades de texto+audio lo cual permite decir que son métricas aceptables para el contexto en el cual está involucrado en donde afectan varios factores externos.

En el dataset Meld por parte de la evaluación unimodal, se aprecia que la emoción neutral en la modalidad de texto es la mejor en cuanto a sus métricas, ya que el F1 score

tiene un 66,0 % y la exactitud un 72,0 % que significa que el modelo está bien entrenado y capacitado para predecir este tipo de emociones con una precisión considerable en comparación a sus competencias.

En la evaluación multimodal la emoción más destacada es joy en la modalidad de texto+rostro con métricas de desempeño para el F1 score del 56,0 % y el modelo está acertando a dicha emoción un 73,1 % en su exactitud.

Cabe recalcar el hecho de que tengan un buen desempeño en una emoción (una gran diferencia) en comparación con las demás, es señal de un sobreajuste en el modelo por un cierto desbalance en los ejemplares.

Capítulo 8

Conclusiones

En este último capítulo se da a conocer una breve descripción del trabajo realizado y sus resultados. Luego una descripción de cumplimientos de objetivos iniciales, presentación de dificultades y limitaciones del trabajo, como también las proyecciones a futuro.

8.1. Conclusiones

El contexto en el que se desarrolló este trabajo es en el área de reconocimiento de emociones dentro de la robótica social, por lo cual se llevó a cabo ciertos criterios de evaluación para efectuar la selección de los distintos dataset donde se aplicaron diferentes análisis descriptivo para adecuar la entrada de datos a la arquitectura ya implementada con técnicas Deep Learning y obtener ciertas métricas de evaluación por cada dataset insertado en la arquitectura y comparar los resultados a través de diversas visualizaciones de datos.

Como resumen del trabajo realizado se puede mencionar que se fue definiendo los objetivos iniciales, instalar y adecuar la arquitectura de reconocimientos de emociones basada en Deep Leaning, como también seleccionar los diferentes dataset a probar para la arquitectura en la que se trabajó y se identificó las métricas de evaluación. Se llevó a cabo un preprocesamiento por cada modalidad, adecuando la entrada de datos a los modelos correspondientes y entrenando los conjuntos de datos. Para desarrollar un enfoque unimodal y multimodal con el fin de obtener un resumen de las métricas y visualizar los resultados.

Los resultados muestran que los diferentes modelos pueden identificar emociones utilizando imágenes recortadas, audios y transcripciones de lo que se dice. Sin embargo, los conjuntos de datos utilizados no han sido diseñados para tareas multimodales y es por eso que en este contexto de robótica social se dan a conocer que son muy inestables y de muy bajo rendimiento para llevarlo a esta realidad al momento de evaluar con diferentes modalidades ya sea por la calidad de los datos, ruido externo, procesamiento de imágenes

o simplemente los modelos a utilizar dentro de la arquitectura. Cabe destacar que los resultados del dataset AffWild2 fueron tomados de un trabajo en colaboración con el fin de comparar este dataset bajo los criterios a evaluar junto con los demás conjuntos de datos trabajados [85]

Las dificultades que se presentaron durante el trabajo fue entender y ordenar los distintos colab que se tenían en el proyecto con respecto a la arquitectura, a lo cual había que solicitar ciertos archivos que faltaban y ejecutar el código con la arquitectura bien ordenada. Por otro lado, se encontraron ciertos problemas de sintaxis con el lenguaje PyTorch y en un tiempo determinado un poco limitado con la capacidad del hardware del equipo, como también el límite de la GPU al momento de ejecutar el entrenamiento de las distintas modalidades. Por lo que se trabajó remotamente con servidores de la universidad con el fin de tener mayor almacenamiento y evitar tener problemas al momento de entrenar grandes cantidades de datos. Por último, la compleja situación de encontrar dataset que estuvieran en un ambiente no controlado y que se encontraran disponibles para el público, ya que al momento de solicitar ciertos dataset no se encontraban respuesta de los autores para adquirirlos.

Las limitaciones del trabajo es que en un contexto de robótica social los resultados que se obtienen son para decidir que dataset son los más recomendables para aplicarlo en un ambiente no controlado. Por lo que la investigación llega hasta el punto de obtener ciertos resultados y todo lo que tiene que ver con la capacidad sensorial del robot para detectar las emociones de las personas y obtener una interacción humano robot, lo más eficiente y de calidad posible es parte de ir trabajando con otras investigaciones para ampliar lo llevado a cabo.

Los próximos pasos en esta investigación incluyen la configuración de un nuevo conjunto de datos en estado salvaje. Mejorar los procesos de preprocesamiento de cada fuente de datos y sus etiquetados considerando las tareas multimodales. Reentrenar en el conjunto de redes tomando en consideración algunas técnicas de optimización en busca de lograr un rendimiento unimodal y multimodal.

Las proyecciones para el futuro es avanzar en esta área para que los enfoques multimodales de reconocimiento de emociones sean apropiados para el entrenamiento de los robots sociales y que se puedan adaptar de acuerdo con la capacidad sensorial de los robots y la calidad de los datos. Por ello, los resultados de esta evaluación de los distintos dataset van a otorgar una gran contribución para esta área, como también la búsqueda de nuevos dataset que permitan mejorar los resultados obtenidos a través de las distintas modalidades con el fin de complementar la investigación.

Bibliografía

[1] R. H. Sampieri, C. F. Collado, and M. P. B. Lucio, *Metodologia de la investigacion*, 6th ed. New York, NY: McGraw-Hill, 2014.

[2] "Building vgg19 with keras," 2022. [Online]. Available: https://saicharanars.medium.com/building-vgg19-with-keras-f516101c24cf

[3] J. Heredia, E. Lopes-Silva, Y. Cardinale, J. Diaz-Amado, I. Dongo, W. Graterol, and A. Aguilera, "Adaptive multimodal emotion detection architecture for social robots," *IEEE Access*, vol. 10, pp. 20 727–20 744, 2022. [Online]. Available: https://doi.org/10.1109/access.2022.3149214

[4] J. Kharibam and A. Devi, "Automatic speaker recognition using mfcc and artificial neural network," *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, pp. 39–42, 12 2019.

[5] "Overview of artificial intelligence and role of natural language processing in big data - datasciencecentral.com," 2022. [Online]. Available: https://www.datasciencecentral.com/overview-of-artificial-intelligence-and-role-of-natural-language/

[6] W. Shen, J. Chen, X. Quan, and Z. Xie, "Dialogxl: All-in-one xlnet for multi-party conversation emotion recognition," in *AAAI*, 2021.

[7] J. Heredia, Y. Cardinale, I. Dongo, and J. Díaz-Amado, "A multi-modal visual emotion recognition method to instantiate an ontology," in *Proceedings of the 16th International Conference on Software Technologies*. SCITEPRESS - Science and Technology Publications, 2021. [Online]. Available: https://doi.org/10.5220/0010516104530464

[8] A. Ruiz-Garcia, M. Elshaw, A. Altahhan, and V. Palade, "A hybrid deep learning neural approach for emotion recognition from facial expressions for socially assistive robots," *Neural Computing and Applications*, vol. 29, no. 7, pp. 359–373, Feb. 2018. [Online]. Available: https://doi.org/10.1007/s00521-018-3358-8

[9] A. Henschel, G. Laban, and E. S. Cross, "What makes a robot social? a review of social robots from science fiction to a home or hospital near you," *Current Robotics Reports*, vol. 2, no. 1, pp. 9–19, Feb. 2021. [Online]. Available: https://doi.org/10.1007/s43154-020-00035-0

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[11] W. Mellouk and W. Handouzi, "Facial emotion recognition using deep learning: review and insights," *Procedia Computer Science*, vol. 175, pp. 689–694, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.07.101

[12] J. A. Heredia Parillo, "A multi-modal emotion recogniser based on the integration of multiple fusion methods," Jan 1970. [Online]. Available: https://bibliotecadigital.oducal.com/Record/ir-20.500.12590-16940/Description

[13] T. B. Sheridan, "Human–robot interaction," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 58, no. 4, pp. 525–532, Apr. 2016. [Online]. Available: https://doi.org/10.1177/0018720816644364

[14] T. Song, W. Zheng, C. Lu, Y. Zong, X. Zhang, and Z. Cui, "Mped: A multi-modal physiological emotion database for discrete emotion recognition," *IEEE Access*, vol. 7, pp. 12 177–12 191, 2019.

[15] H. Meng, N. Bianchi-Berthouze, Y. Deng, J. Cheng, and J. P. Cosmas, "Time-delay neural network for continuous emotional dimension prediction from facial expression sequences," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 916–929, 2016.

[16] H. Zhang, A. Jolfaei, and M. Alazab, "A face emotion recognition method using convolutional neural network and image edge computing," *IEEE Access*, vol. 7, pp. 159 081–159 089, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2949741

[17] L. Bernedo-Flores, I. Dongo, Y. Cardinale, A. Aguilera, D. Pacheco, and J. Heredia, "Rihe: A robot-independent height emotion dataset," 2022.

[18] J. Heredia, E. Lopes-Silva, Y. Cardinale, J. Diaz-Amado, I. Dongo, W. Graterol, and A. Aguilera, "Adaptive multimodal emotion detection architecture for social robots," *IEEE Access*, vol. 10, pp. 20 727–20 744, 2022.

[19] "Papers with code - meld: A multimodal multi-party dataset for emotion recognition in conversations," 2022. [Online]. Available: https://paperswithcode.com/paper/meld-a-multimodal-multi-party-dataset-for

[20] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, no. 2, p. 102–107, mar 2016. [Online]. Available: https://doi.org/10.1109/MIS.2016.31

[21] M. Soleymani, D. Garcia, B. Jou, B. Schuller, S.-F. Chang, and M. Pantic, "A survey of multimodal sentiment analysis," *Image and Vision Computing*, vol. 65, pp. 3–14, 2017, multimodal Sentiment Analysis and Mining in the Wild Image and Vision Computing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885617301191

[22] E. D. P. Supervisor., *EDPS TechDispatch: facial emotion recognition. Issue 1, 2021.* Publications Office, 2021. [Online]. Available: https://data.europa.eu/doi/10.2804/014217

[23] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, "Speech emotion recognition using deep learning techniques: A review," *IEEE Access*, vol. 7, pp. 117 327–117 345, 2019.

[24] J. Guo, "Deep learning approach to text analysis for human emotion detection from big data," *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 113–126, 2022. [Online]. Available: https://doi.org/10.1515/jisys-2022-0001

[25] A. Clark, S. Abdullah, and S. Ameen, "A comparison of decision-feedback equalizers for a 9600 bit/s modem," *Journal of The Institution of Electronic and Radio Engineers*, vol. 58, 03 1988.

[26] C. Marechal, D. Mikołajewski, K. Tyburek, P. Prokopowicz, L. Bougueroua, C. Ancourt, and K. Wegrzyn-Wolska, *Survey on AI-Based Multimodal Methods for Emotion Detection*, 03 2019.

[27] Y.-T. Lan, W. Liu, and B.-L. Lu, "Multimodal emotion recognition using deep generalized canonical correlation analysis with an attention mechanism," 07 2020, pp. 1–6.

[28] P. Bhattacharya, R. Gupta, and Y. Yang, "The contextual dynamics of multimodal emotion recognition in videos," 04 2020.

[29] K. Kabacińska, T. J. Prescott, and J. M. Robillard, "Socially assistive robots as mental health interventions for children: A scoping review," *International Journal of Social Robotics*, vol. 13, no. 5, pp. 919–935, jul 2020. [Online]. Available: https://doi.org/10.1007%2Fs12369-020-00679-0

[30] S. B. Daily, M. T. James, D. Cherry, J. J. Porter, S. S. Darnell, J. Isaac, and T. Roy, "Chapter 9 - affective computing: Historical foundations, current applications, and future trends," in *Emotions and Affect in Human Factors and Human-Computer Interaction*, M. Jeon, Ed. San Diego: Academic Press, 2017, pp. 213–231. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128018514000094

[31] A. Thomaz, G. Hoffman, and M. Cakmak, "Computational human-robot interaction," *Foundations and Trends® in Robotics*, vol. 4, no. 2-3, pp. 105–223, 2016. [Online]. Available: http://dx.doi.org/10.1561/2300000049

[32] "What is a data set?" 2022. [Online]. Available: https://towardsdatascience.com/what-is-a-data-set-9c6e38d33198

[33] A. Dhall, R. Goecke, J. Joshi, J. Hoey, and T. Gedeon, "Emotiw 2016: Video and group-level emotion recognition challenges," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ser. ICMI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 427–432. [Online]. Available: https://doi.org/10.1145/2993148.2997638

[34] A. Bagher Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 2236–2246. [Online]. Available: https://aclanthology.org/P18-1208

[35] "Do you know what deep learning is?" 2022. [Online]. Available: https://www.oracle.com/data-science/machine-learning/what-is-deep-learning/

[36] [Online]. Available: https://doi.org/10.1145/2623330.2630809

[37] Priya_dharshini__, "Must-Know statistical data analysis techniques in machine learning!" https://www.analyticsvidhya.com/blog/2021/06/must-know-statistical-data-analysis-techniques-in-machine-learning/, Jun. 2021.

[38] M. Mohamad, A. Selamat, and K. Salleh, "An analysis on deep learning approach performance in classifying big data set," 09 2019, pp. 35–39.

[39] C. Liu, "More performance evaluation metrics for classification problems you should know - kdnuggets," 2022. [Online]. Available: https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html

[40] Y. Tan, Z. Sun, F. Duan, J. Solé-Casals, and C. F. Caiafa, "A multimodal emotion recognition method based on facial expressions and electroencephalography," *Biomedical Signal Processing and Control*, vol. 70, p. 103029, Sep. 2021. [Online]. Available: https://doi.org/10.1016/j.bspc.2021.103029

[41] H. Huang, Z. Hu, W. Wang, and M. Wu, "Multimodal emotion recognition based on ensemble convolutional neural network," *IEEE Access*, vol. 8, pp. 3265–3271, 2020. [Online]. Available: https://doi.org/10.1109/access.2019.2962085

[42] Y.-T. Lan, W. Liu, and B.-L. Lu, "Multimodal emotion recognition using deep generalized canonical correlation analysis with an attention mechanism," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2020. [Online]. Available: https://doi.org/10.1109/ijcnn48605.2020.9207625

[43] S. Gupta, "Facial emotion recognition in real-time and static images," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*. IEEE, Jan. 2018. [Online]. Available: https://doi.org/10.1109/icisc.2018.8398861

[44] Z.-T. Liu, F.-F. Pan, M. Wu, W.-H. Cao, L.-F. Chen, J.-P. Xu, R. Zhang, and M.-T. Zhou, "A multimodal emotional communication based humans-robots interaction system," in *2016 35th Chinese Control Conference (CCC)*. IEEE, Jul. 2016. [Online]. Available: https://doi.org/10.1109/chicc.2016.7554357

[45] L.-A. Perez-Gaspar, S.-O. Caballero-Morales, and F. Trujillo-Romero, "Multimodal emotion recognition with evolutionary computation for human-robot interaction," *Expert Systems with Applications*, vol. 66, pp. 42–61, Dec. 2016. [Online]. Available: https://doi.org/10.1016/j.eswa.2016.08.047

[46] P. Barros, C. Weber, and S. Wermter, "Emotional expression recognition with a cross-channel convolutional neural network for human-robot interaction," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, Nov. 2015. [Online]. Available: https://doi.org/10.1109/humanoids.2015.7363421

[47] L. Chen, M. Li, W. Su, M. Wu, K. Hirota, and W. Pedrycz, "Adaptive feature selection-based AdaBoost-KNN with direct optimization for dynamic emotion recognition in human–robot interaction," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 2, pp. 205–213, Apr. 2021. [Online]. Available: https://doi.org/10.1109/tetci.2019.2909930

[48] M. Anjum, "Emotion recognition from speech for an interactive robot agent," in *2019 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, Jan. 2019. [Online]. Available: https://doi.org/10.1109/sii.2019.8700376

[49] D. O. Melinte and L. Vladareanu, "Facial expressions recognition for human–robot interaction using deep convolutional neural networks with rectified adam optimizer," *Sensors*, vol. 20, no. 8, p. 2393, Apr. 2020. [Online]. Available: https://doi.org/10.3390/s20082393

[50] W. Zhang, Y. Zhang, L. Ma, J. Guan, and S. Gong, "Multimodal learning for facial expression recognition," *Pattern Recognition*, vol. 48, no. 10, pp. 3191–3202, Oct. 2015. [Online]. Available: https://doi.org/10.1016/j.patcog.2015.04.012

[51] L. Chen, M. Wu, M. Zhou, J. She, F. Dong, and K. Hirota, "Information-driven multirobot behavior adaptation to emotional intention in human–robot interaction," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 647–658, Sep. 2018. [Online]. Available: https://doi.org/10.1109/tcds.2017.2728003

[52] J. A. Heredia Parillo, "A multi-modal emotion recogniser based on the integration of multiple fusion methods," 2021.

[53] H. M and S. M.N, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining &amp Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, Mar. 2015. [Online]. Available: https://doi.org/10.5121/ijdkp.2015.5201

[54] D. McColl, A. Hong, N. Hatakeyama, G. Nejat, and B. Benhabib, "A survey of autonomous human affect detection methods for social robots engaged in natural HRI," *Journal of Intelligent & Robotic Systems*, vol. 82, no. 1, pp. 101–133, Aug. 2015. [Online]. Available: https://doi.org/10.1007/s10846-015-0259-2

[55] S. Ramis, J. M. Buades, and F. J. Perales, "Using a social robot to evaluate facial expressions in the wild," *Sensors*, vol. 20, no. 23, p. 6716, Nov. 2020. [Online]. Available: https://doi.org/10.3390/s20236716

[56] B. Zhang and E. M. Provost, "Automatic recognition of self-reported and perceived emotions," in *Multimodal Behavior Analysis in the Wild*. Elsevier, 2019, pp. 443–470. [Online]. Available: https://doi.org/10.1016/b978-0-12-814601-9.00027-4

[57] O. Apuke, "Quantitative research methods : A synopsis approach," *Arabian Journal of Business and Management Review (kuwait Chapter).*, vol. 6, pp. 40–47, 10 2017.

[58] "Challenges in representation learning: Facial expression recognition challenge — kaggle," 2022. [Online]. Available: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

[59] "fer2013/expressions at master · gitshanks/fer2013," 2022. [Online]. Available: https://github.com/gitshanks/fer2013/tree/master/Expressions

[60] "About kdef," 2022. [Online]. Available: https://www.kdef.se/home/aboutKDEF.html

[61] "Github - mlsmall/facial-expression-recognition: Created a web application that recognizes facial expressions," 2022. [Online]. Available: https://github.com/mlsmall/Facial-Expression-Recognition

[62] A. Zadeh, P. P. Liang, S. Poria, P. Vij, E. Cambria, and L.-P. Morency, "Multi-attention recurrent network for human communication comprehension," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[63] J. Chen, C. Wang, K. Wang, C. Yin, C. Zhao, T. Xu, X. Zhang, Z. Huang, M. Liu, and T. Yang, "Heu emotion: a large-scale database for multimodal emotion recognition in the wild," *Neural Computing and Applications*, vol. 33, 07 2021.

[64] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "MELD: A multimodal multi-party dataset for emotion recognition in conversations," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 527–536. [Online]. Available: https://aclanthology.org/P19-1050

[65] C.-C. Hsu, S.-Y. Chen, C.-C. Kuo, T.-H. Huang, and L.-W. Ku, "EmotionLines: An emotion corpus of multi-party conversations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1252

[66] V. Vielzeuf, S. Pateux, and F. Jurie, "Temporal multimodal fusion for video emotion classification in the wild," 11 2017, pp. 569–576.

[67] B. Knyazev, R. Shvetsov, N. Efremova, and A. Kuharenko, "Leveraging large face recognition data for emotion classification," in *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*. IEEE, 2018, pp. 692–696.

[68] "Github - caterina1996/sfew_dataset: Sfew train and test data," 2022. [Online]. Available: https://github.com/Caterina1996/SFEW_dataset

[69] Y. Li, J. Tao, L. Chao, W. Bao, and Y. Liu, "CHEAVD: a chinese natural emotional audio–visual database," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 6, pp. 913–924, Sep. 2016. [Online]. Available: https://doi.org/10.1007/s12652-016-0406-z

[70] O. Martin, I. Kotsia, B. Macq, and I. Pitas, "The eNTERFACE&ampamp#14605 audio-visual emotion database," in *22nd International Conference on Data*

Engineering Workshops (ICDEW'06). IEEE, 2006. [Online]. Available: https://doi.org/10.1109/icdew.2006.145

[71] D. Kollias and S. Zafeiriou, "Aff-wild2: Extending the aff-wild database for affect recognition," *arXiv preprint arXiv:1811.07770*, 2018.

[72] "i·bug - resources - aff-wild2 database," 2022. [Online]. Available: https://ibug.doc.ic.ac.uk/resources/aff-wild2/

[73] "Facial expressions in the wild (sfew / afew)." [Online]. Available: http://cs.anu.edu.au/few

[74] "Facial expressions in thewild project," 2022. [Online]. Available: https://cs.anu.edu.au/few/AFEW.html

[75] K. Venkataramanan and H. R. Rajamohan, "Emotion recognition from speech," *ArXiv*, vol. abs/1912.10458, 2019.

[76] "Reconocimiento de emociones del habla basado en mfcc mejorado." [Online]. Available: https://doi-org.bibliotecadigital.uv.cl/10.1145/3207677.3278037

[77] W. Shen, J. Chen, X. Quan, and Z. Xie, "Dialogxl: All-in-one xlnet for multi-party conversation emotion recognition," 2020. [Online]. Available: https://arxiv.org/abs/2012.08695

[78] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1906.08237

[79] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," 2019. [Online]. Available: https://arxiv.org/abs/1905.00641

[80] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania, and S. Chintala, "Pytorch distributed: Experiences on accelerating data parallel training," *Proc. VLDB Endow.*, vol. 13, no. 12, p. 3005–3018, aug 2020. [Online]. Available: https://doi.org/10.14778/3415478.3415530

[81] "Google colab," 2022. [Online]. Available: https://research.google.com/colaboratory/faq.html

[82] "How to use google drive - computer - google drive help," 2022. [Online]. Available: https://support.google.com/drive/answer/2424384?hl=en&co=GENIE.Platform%3DDesktop

[83] C. Kirwan and F. Zhiyong, "Smart city functions," in *Smart Cities and Artificial Intelligence*. Elsevier, 2020, pp. 163–192. [Online]. Available: https://doi.org/10.1016/b978-0-12-817024-3.00008-8

[84] Vaishali and S. Singh, "Real-time object detection system using caffe model," 2019. [Online]. Available: https://www.academia.edu/40034205/IRJET_Real_Time_Object_Detection_System_using_Caffe_Model

[85] A. Aguilera, D. Mellado, and F. Rojas, "An assessment of in-the-wild datasets for multimodal emotion recognition," *Sensors*, vol. 23, no. 11, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/11/5184

[86] S. Gowrishankar and A. Veena, *Introduction to python programming*. CRC Press, 2018.

[87] "¿qué es pytorch? todo lo que debes saber — ciberseguridad," 2022. [Online]. Available: https://ciberseguridad.com/guias/nuevas-tecnologias/machine-learning/pytorch/

[88] 2022. [Online]. Available: https://anydesk.com/en/features

[89] "A manager's guide to successful strategy implementation — hbs online," 2022. [Online]. Available: https://online.hbs.edu/blog/post/strategy-implementation-for-managers#:~:text=Strategy%20implementation%20is%20the%20process,efficiently%2C%20effectively%2C%20and%20consistently.

Apéndice A

Anexo

A.1. Contenido del dataset Sfew

El archivo que se analizó fue la carpeta angry donde se descomprimió con la finalidad de apreciar su contenido que viene siendo muchas imágenes que están etiquetadas cada una de ellas y provienen de extractos de diferentes películas de Hollywood.

Este dataset actúa como una base de datos de modalidad facial la cual contiene 7 emociones con diferentes frames de videos que corresponde al dataset Afew por lo cual tiene una cierta dependencia y similitudes al analizar cada imagen que se encuentra en las diferentes carpetas. La cantidad de frames por video va a depender exclusivamente a la duración de dicho video que fue extraído del dataset afew en un ambiente no controlado.

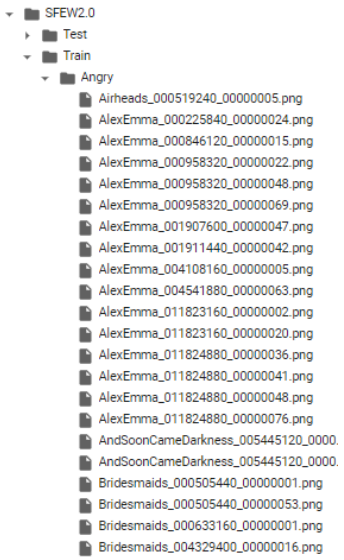


Figura A.1: Frame de imágen en Sfew

A.2. Tabla Resultados Unimodales

Para mantener un orden de las métricas obtenidas mediante el entrenamiento de la data en cada modalidad se realiza una tabla como se muestra en la figura A.2, con la finalidad de presentar las métricas de evaluación con su respectivo promedio por cada emoción y llevar a cabo una discusión sobre estos resultados.

Emociones	Modalidad facial		Modalidad de audio		Modalidad de texto	
	F1	Acc(%)	F1	Acc(%)	F1	Acc(%)
Angry Happy Neutral Sad						
Promedio						

Figura A.2: Tabla de Resultados Unimodal

A.3. Tabla Resultados Multimodales

En este caso se realiza una tabla como se muestra en la figura A.3 para resumir los resultados multimodales al momento de pasar por el embracenet+ y ser entrenado la data con la misma finalidad de realizar una discusión de resultados a partir de las métricas obtenidas, sus promedios y las emociones dentro de cada evaluación.

Emociones	F+A+T		F+T		A+T		F+A	
	F1	Acc(%)	F1	Acc(%)	F1	Acc(%)	F1	Acc(%)
Angry Happy Neutral Sad								
Promedio								

Figura A.3: Tabla de Resultados Multimodales

A.4. Software utilizado

- Pandas es una librería que cambió el panorama del análisis de datos en python por completo y está disponible bajo licencia BSD. Pandas está construido sobre numpy y tiene dos estructuras de datos importantes como son las series y dataframe. Puede contener cualquier tipo de datos como INT, FLOAT, SRINGS, OBJECTS y otros.

Cada uno de los datos almacenados en serie se etiqueta después del índice. Dataframe es una estructura de datos tabular con filas y columnas. En el mundo real, los datos nunca están en orden y los pandas se pueden usar para completar los datos faltantes, remodelar los conjuntos de datos, cortar, indexar, fusionar y unir el conjunto de datos. Pandas se puede utilizar para leer archivos CSV, Microsoft Excel, SQL, archivos de formato HDF5 [86].

- SciPy es una librería que está diseñada para trabajar con matrices NumPy y proporciona muchas rutinas numéricas fáciles de usar y eficientes, como rutinas para integración y optimización numérica [86].
- Numpy es una librería que proporciona objetos de matriz N-dimensional que se pueden utilizar para realizar álgebra lineal, transformada de Fourier y otras operaciones matemáticas [86].
- Matplotlib es la biblioteca de gráficos más antigua y popular disponible para Python. Con estas herramientas, tenemos mejores posibilidades de resolver problemas científicos y crear prototipos de trabajo más rápidamente que cualquier otra herramienta de la competencia[86].
- Pytorch es una biblioteca de aprendizaje automático de código abierto que se especializa en cálculos de tensor, diferenciación automática y aceleración de GPU. Se utiliza en investigaciones y aplicaciones de aprendizaje profundo para hacer computacionalmente más rápida y menos costosa [87]. Por otro lado, emplea computación dinámica lo que brinda una mayor flexibilidad en la creación de redes más complejas y emplea ideas básicas de python como clases, estructuras, bucles que son más fácil de entender para las personas.
- AnyDesk es una aplicación que ofrece acceso remoto independiente de la plataforma a computadores personales. Permite compartir el escritorio sin latencia, un control remoto estable y una transmisión de datos rápida y segura entre dispositivos. [88]

A.5. Lenguajes de programación

Python es un lenguaje de programación gratuito de propósito general con una hermosa sintaxis. Está disponible en muchas plataformas, incluyendo Windows, Linux y Mac OS. Debido a su naturaleza inherentemente fácil de aprender junto con las características orientadas a objetos, Python se utiliza para desarrollar y demostrar aplicaciones rápidamente. Es conocida por su simplicidad y amabilidad para los desarrolladores, donde es el lenguaje de programación principal de más rápido crecimiento. El lenguaje de programación estándar viene con una variedad de conjuntos de bibliotecas integradas. Las soluciones de

alojamiento para aplicaciones Python también son muy baratas. Un lenguaje versátil como Python se puede usar no solo para escribir scripts simples para manejar operaciones de archivos, sino también para desarrollar sitios web con tráfico masivo para organizaciones de TI corporativas [86].

A.6. Estrategia de implementación

Es el proceso de convertir los planes en acción para alcanzar el resultado deseado, donde se implementan decisiones y ejecutan procesos de manera eficiente, eficaz y consistente. En donde la principal estrategia para abordar la evaluación de los distintos conjuntos de datos es modularizar el trabajo individualmente por cada dataset en diferentes colab, pero basándose en la misma arquitectura [89]. En la figura A.4 se da a conocer todas las etapas que se describen a continuación:

Recolección de datos: Se realizó un criterio de selección de los dataset a evaluar donde primero se tiene que considerar que el dataset esté balanceado, la cantidad de modalidades, la calidad de los datos, y el ambiente de trabajo. Luego se aplicó el análisis descriptivo que permitirá conocer el contenido de cada uno de ellos, los datos procesados, la cantidad de emociones, modalidades, sentimientos y adecuar la entrada de datos a la arquitectura implementada con Deep Learning [18].

Almacenamiento de datos: Todos los datos que provienen de los diferentes dataset se fueron adecuando a la arquitectura ya implementada con técnicas Deep Learning dentro de una red neuronal las cuales se procesaron las distintas modalidades de una forma independiente para luego fusionarlo con el método Embracenet+ y aplicar ciertas métricas de evaluación [18].

Análisis de datos: Consiste en seleccionar las métricas de evaluación que son el F1 score y la exactitud en donde se van a llevar a cabo evaluaciones para las 3 modalidades de forma individual y todo el sistema compuesto se evalúan por separado donde las emociones a analizar se desarrollarán sus métricas individualmente. Luego se tomará un enfoque multimodal para aplicar 4 experimentos que consisten en la evaluación de todas las modalidades [17].

Resultados: Al tener los resultados se pudieron visualizar los datos a través de distintos gráficos de comparaciones de las métricas de evaluación y un gráfico de comparación de las precisiones reportadas con diferentes modalidades y tablas comparativas para tomar ciertas decisiones y conclusiones [18].

Control o cambios: Lo fundamental es cumplir con los objetivos planteados al comienzo y las actividades a realizar en la planificación del proyecto. La idea es seguir la arquitectura y en el caso de que se tenga que realizar una modificación importante en alguna función, método o clase se tiene que evidenciar y explicar la razón por la que se llevó a cabo. Es por ello que se llevó un control constante por cada dataset a evaluar y de los cambios a realizar con el fin de tener un manual del usuario lo más claro posible y para el proceso de la implantación [89].

Retrospectiva: Por último, se aplicó una retrospectiva o revisión de cómo fue el proceso una vez que se haya implementado por completo, con el fin de evaluar si es que se lograron los objetivos o no y explicar las razones, que obstáculos o desafíos surgieron en el transcurso del proyecto y que lecciones se puede aprender del proceso [89].

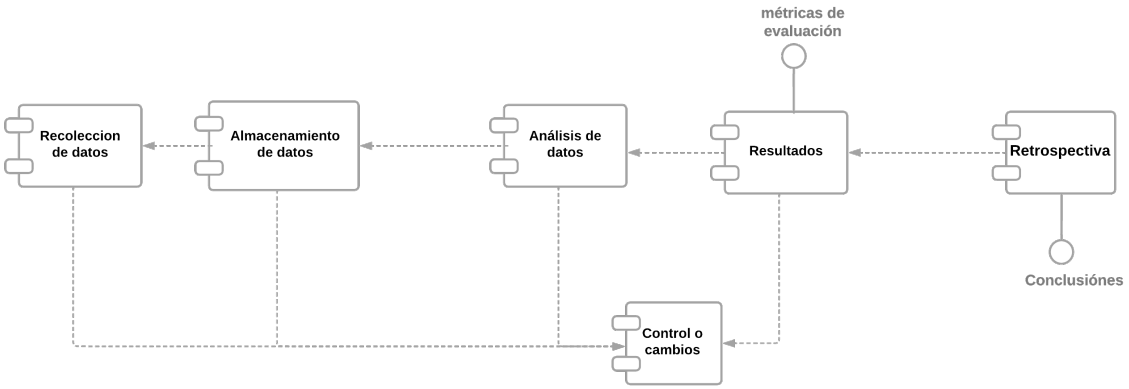


Figura A.4: Estrategia de implementación

A.7. Imágenes de la Ejecución de experimentos

En la figura A.5 se puede apreciar un extracto del entrenamiento donde se van recorriendo los conjuntos de train y test a partir de 20 épocas establecidas donde se van obteniendo diferentes resultados como son el train_loss y val_loss que representan los valores de la función loss para train y test que dan a conocer lo que está aprendiendo la red y, por otro lado, el train_acc y val_acc representan la exactitud que se obtiene a partir de cada época con la finalidad de saber la calidad y la precisión del modelo a la hora de aprender y reconocer las emociones detectadas en cada imagen a partir de los datos obtenidos en el dataset original.

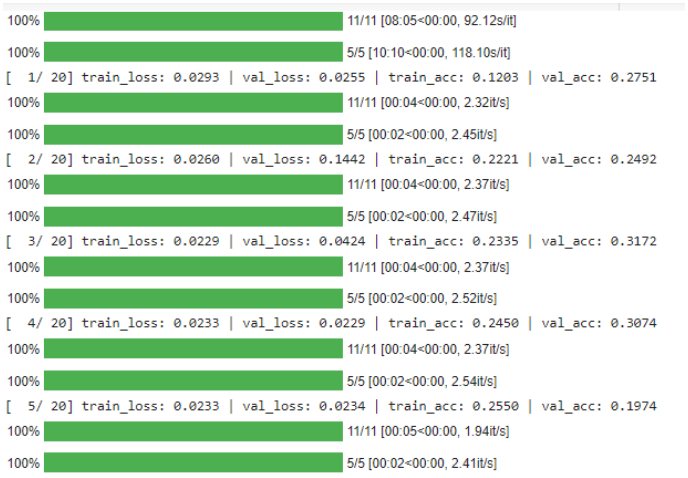


Figura A.5: Entrenamiento Modalidad Facial en Sfew

En las figuras A.6 A.7 está representado la carpeta creada en donde se alojan todos los frames de cada video por emoción a utilizar durante el entrenamiento. Donde principalmente se tomarán en cuenta las emociones estándar como son el angry, happy, neutral y sad para seguir el proceso de comparación con las mismas emociones al igual que los otros datasets.

Figura A.6: Carpeta de frames categorizadas en Afew

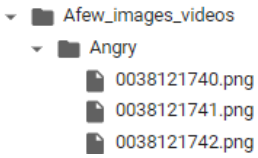


Figura A.7: Frames de un video en Afew

Un claro ejemplo de los frames obtenidos por cada video y que están etiquetados se presenta en la figura A.8

Figura A.8: Ejemplos de un frame con etiquetado angry en Afew

Cabe destacar que se ocupó la carpeta train en específico, ya que la carpeta test se encuentran sin etiquetas para diferenciar los videos por emoción y la carpeta val es simplemente es un subconjunto de train. Lo que se llevó a cabo fue dividir la carpeta train manualmente en un 70 % para entrenamiento y un 30 % para test lo que solucionará el problema presentado.

Lo principal es que se va a crear un diccionario para las emociones a utilizar las cuales se van recorriendo las carpetas en donde se alojan los frames por emociones y se van denotando con un índice en representación de cada emoción. Cada ejemplo que se va cargando tendrá la data que será el frame como tal con su etiqueta correspondiente y el nombre de la imagen.

```
classs={'Angry':0, 'Happy':1, 'Neutral':2, 'Sad':3}

class AFEW(Dataset):
    def __init__(self, root_dir='', categories={}, transform=None):
        super(AFEW, self).__init__()
        self.DataRoot = root_dir
        self.Categories = categories
        self.Transform = transform
        self.load_data()

    def load_data(self):
        self.Data = {}

        for cat in self.Categories.keys():

            for img in os.listdir(join(self.DataRoot,cat)):

              self.Data[img]=self.Categories[cat]

        self.DataKeys=list(self.Data.keys())
        print (self.Data)

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

    def make_shuffle(self):
        random.shuffle(self.DataKeys)

    def __getitem__(self, idx):
        if torch.is_tensor(idx):
            idx = idx.tolist()

        lb= self.Data[self.DataKeys[idx]]
        dt= cv2.imread(join(self.DataRoot,list(self.Categories.keys())[lb],self.DataKeys[idx]))

        sample=  {'data': dt, 'label': lb,  'name': self.DataKeys[idx]}

        if self.Transform:
          sample= self.Transform(sample)

        return sample
```

Figura A.9: Clase Personalizada Afew

Como se muestra en la figura A.10 se define primero que nada la cantidad de épocas que se va a recorrer el conjunto de train y test. Luego se inicializan las métricas correspondientes y se procede a iterar los conjuntos de datos bajo el modelo VGG19 y sus etiquetas cada uno.

```
epochI = 0
epochF = 20
```

```
for epoch in range(epochI, epochF):
    Tacloss, Vacloss = 0, 0
    Tacacc, Vacacc = 0, 0
    Tcont, Vcont = 0, 0

    model.train()

    for sample in tqdm(train_dataloader):
        data = sample['data'].float().cuda()
        label = sample['label'].flatten()
        optimizer.zero_grad()
        # print(data.shape)
        out = model(data).to('cpu')
        loss = criterion(out, label)
        loss.backward()
        optimizer.step()
        Tacloss += loss.item()
        Tacacc += (torch.argmax(out, dim=1) == label).float().sum()
        Tcont += data.shape[0]
    model.eval()
    with torch.no_grad():
        for sample in tqdm(test_dataloader):
            data = sample['data'].float().cuda()
            label = sample['label'].flatten()
            out = model(data).to('cpu')
            loss = criterion(out, label)
```

Figura A.10: Entrenamiento modalidad facial en Afew

Como se muestra en la figura A.11 se van obteniendo las métricas a partir de las diferentes épocas donde lo destacable es que estos resultados van a permitir conocer lo viable que viene siendo el modelo utilizado con respecto a su calidad y precisión para predecir ciertas emociones y la capacidad de adaptarse a un ambiento no controlado.

```
            Vacloss += loss.item()
            Vacacc += (torch.argmax(out, dim=1) == label).float().sum()
            Vcont += data.shape[0]
            # sleep(0.25)
    H_train_loss += [Tacloss/Tcont]
    H_train_acc += [Tacacc/Tcont]

    H_val_loss += [Vacloss/Vcont]
    H_val_acc += [Vacacc/Vcont]

    print('[%3d/%3d] train_loss: %.4f | val_loss: %.4f | train_acc: %.4f | val_acc: %.4f'%
          (epoch+1, epochF, H_train_loss[-1], H_val_loss[-1], H_train_acc[-1], H_val_acc[-1]))
```

```
100% 152/152 [02:14<00:00, 1.25it/s]
100% 79/79 [01:08<00:00, 1.52it/s]
[  1/ 20] train_loss: 0.0219 | val_loss: 0.0222 | train_acc: 0.3504 | val_acc: 0.2697
100% 152/152 [02:13<00:00, 1.31it/s]
100% 79/79 [01:07<00:00, 1.55it/s]
[  2/ 20] train_loss: 0.0205 | val_loss: 0.0226 | train_acc: 0.3789 | val_acc: 0.2701
100% 152/152 [02:09<00:00, 1.32it/s]
100% 79/79 [01:07<00:00, 1.52it/s]
[  3/ 20] train_loss: 0.0191 | val_loss: 0.0253 | train_acc: 0.3958 | val_acc: 0.2769
100% 152/152 [02:09<00:00, 1.33it/s]
100% 79/79 [01:07<00:00, 1.57it/s]
[  4/ 20] train_loss: 0.0164 | val_loss: 0.0265 | train_acc: 0.5261 | val_acc: 0.2916
100% 152/152 [02:11<00:00, 1.31it/s]
100% 79/79 [01:08<00:00, 1.52it/s]
```

Figura A.11: Continuación entrenamiento modalidad facial en Afew

La siguiente figura A.12 representa la carpeta mel creada con los audios del conjunto de train y test en formato npy categorizados por emoción en cada carpeta listo para ser entrenados dentro de la arquitectura.

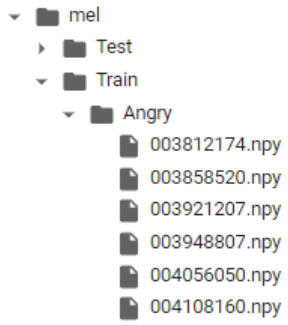


Figura A.12: Carpeta mel en Afew

Aquí se creó la clase personalizada para el audio donde se enfoca en la carpeta mel, recorriendo toda la carpeta de train y test hasta llegar a los archivos numpy que contiene los espectrogramas con sus características. Para ello se van separando por conjuntos de datos para el entrenamiento y las pruebas con tal de ingresar a la arquitectura de la mejor forma posible, como se muestra en la figura A.13.

```
class AudioDataset(Dataset):
    def __init__(self, root_dir='', categories={}, transform=None):

        super(AudioDataset, self).__init__()
        self.DataRoot = root_dir
        self.Categories= categories
        self.Transform = transform

        self.load_data()

    def load_data(self):
        self.Data = {}

        for cat in self.Categories.keys():
          for aud in os.listdir(join(self.DataRoot,cat)):
            self.Data[aud]=self.Categories[cat]

        self.DataKeys=list(self.Data.keys())

        print(self.Categories)
```

Figura A.13: Clase Personalizada audio Afew

Donde se van cargando los archivos numpy en la ruta donde se alojan con sus etiquetas correspondientes para identificar por cada ejemplo la data, su etiqueta y el nombre de cada archivo como se muestra en la figura A.14.

```
def __len__(self):
    return len(self.DataKeys)

def __getitem__(self, idx):

    if torch.is_tensor(idx):
        idx = idx.tolist()

    label = self.Data[self.DataKeys[idx]]
    data = np.load(self.DataRoot+list(self.Categories.keys())[label]+'/'+self.DataKeys[idx])

    sample = {'data': data, 'label': label, 'name': self.DataKeys[idx]}

    if self.Transform:
        sample = self.Transform(sample)
    return sample

def make_shuffle(self):
    random.shuffle(self.DataKeys)
```

Figura A.14: Continuación Clase Personalizada audio Afew

Con el entrenamiento en curso se van definiendo las métricas correspondientes como el F1 score, la exactitud y la función de pérdida para conocer el rendimiento del modelo, la calidad a la hora de predecir las emociones en la modalidad de audio, la precisión y cuanto se ha equivocado el modelo durante cada época. Todo se va a iterar por 20 épocas al igual que la modalidad anterior para seguir con el estándar como se muestra en las figuras A.15, A.16 y A.17

```
epochI, epochF = 40, 60

H_train_loss = []
H_train_acc = []

H_val_loss = []
H_val_acc = []
```

```
for epoch in range(epochI, epochF):
    Tacloss, Vacloss = 0, 0
    Tacacc, Vacacc = 0, 0
    Tcont, Vcont = 0, 0
    model.train()
    for sample in tqdm(train_dataloader):
        data = sample['data'].float().cuda()
        label = sample['label'].flatten()

        out = model(data).to('cpu')

        if scheduler:
            scheduler.step()

        loss = criterion(out, label)
        optimizer.zero_grad()
        loss.backward()
        if clip:
            nn.utils.clip_grad_norm_(model.parameters(), 1)
        optimizer.step()

        Tacloss += loss.item()
        Tacacc += (torch.argmax(out, dim=1) == label).float().sum()
        Tcont += data.shape[0]
    model.eval()
```

Figura A.15: Entrenamiento Afew Audio

```
with torch.no_grad():
    for sample in tqdm(test_dataloader):
        data = sample['data'].float().cuda()
        label = sample['label'].flatten()
        out = model(data).to('cpu')
        loss = criterion(out, label)

        Vacloss += loss.item()
        Vacacc += (torch.argmax(out, dim=1) == label).float().sum()
        Vcont += data.shape[0]
        # sleep(0.25)
H_train_loss += [Tacloss/Tcont]
H_train_acc += [Tacacc/Tcont]

H_val_loss += [Vacloss/Vcont]
H_val_acc += [Vacacc/Vcont]

print('[%3d/%3d] train_loss: %.4f | val_loss: %.4f | train_acc: %.4f | val_acc: %.4f'%
        (epoch+1, epochF, H_train_loss[-1], H_val_loss[-1], H_train_acc[-1], H_val_acc[-1]))
```

Figura A.16: Continuación Entrenamiento Afew Audio

```
100% 18/18 [00:01<00:00, 9.25it/s]
100% 8/8 [00:00<00:00, 15.12it/s]
[ 41/ 60] train_loss: 0.0431 | val_loss: 0.0449 | train_acc: 0.3716 | val_acc: 0.3629
100% 18/18 [00:01<00:00, 9.27it/s]
100% 8/8 [00:00<00:00, 15.25it/s]
[ 42/ 60] train_loss: 0.0425 | val_loss: 0.0449 | train_acc: 0.3860 | val_acc: 0.4008
100% 18/18 [00:01<00:00, 9.26it/s]
100% 8/8 [00:00<00:00, 14.50it/s]
[ 43/ 60] train_loss: 0.0418 | val_loss: 0.0450 | train_acc: 0.4165 | val_acc: 0.3924
100% 18/18 [00:01<00:00, 9.24it/s]
100% 8/8 [00:00<00:00, 14.58it/s]
[ 44/ 60] train_loss: 0.0424 | val_loss: 0.0446 | train_acc: 0.3932 | val_acc: 0.4093
100% 18/18 [00:01<00:00, 9.25it/s]
100% 8/8 [00:00<00:00, 14.94it/s]
[ 45/ 60] train_loss: 0.0420 | val_loss: 0.0446 | train_acc: 0.4039 | val_acc: 0.4219
100% 18/18 [00:01<00:00, 9.18it/s]
100% 8/8 [00:00<00:00, 12.22it/s]
```

Figura A.17: Métricas del Entrenamiento en Afew

Lo siguiente es pasar por el embracenet+ todas las modalidades presentes en el dataset, por lo que primero se realizó una clase del dataset con diferentes parámetros como el diccionario de clases, las predicciones de las tres modalidades, el dataframe a utilizar del texto, el método avg para calcular el promedio de cada una de las métricas de evaluación y el modo con respecto al conjunto de entrenamiento. Dando paso a la función load_data que va recibiendo los resultados de cada modalidad por parámetros y luego los va recopilando en una lista por cada uno de ellos como se muestra en la figura A.18

Paralelamente, se va a recorrer el dataframe que contiene los textos del conjunto de train y test con su etiqueta correspondiente, considerando cada modalidad a través de los condicionales, verificando si la key corresponde al nombre del archivo como tal para llegar y guardarlo en la lista definida en la función.

```
class DatasetAFEW(Dataset):
    def __init__(self, classes, FaceR, AudioR, TextR, T30, method='avg', mode='Train', transform=None):
        super(DatasetAFEW, self).__init__()
        self.Transform = transform
        self.Classes = classes
        self.Mode = mode
        self.Method = method
        self.loadData(FaceR, AudioR, TextR, T30)

    def loadData(self, face_results, audio_results, text_results, T30):
        LFks = list(face_results.keys())
        LAks = list(audio_results.keys())
        LTks = list(text_results.keys())

        self.Data = {}

        for i,r in T30.iterrows():

            if r['Mode']!= self.Mode:

                continue

            key=r['name'][0:9]

            if key in LFks:
                FD = self.convert(face_results[key][0])
            else:
                FD = None
            if key+'.txt' in LTks:
                TD = text_results[key+'.txt'][0]
            else:
                TD = None
            if key+'.npy' in LAks:
                AD = audio_results[key+ '.npy'][0].numpy()
            else:
                AD =None

            self.Data[key] = (r['target'], FD, AD, TD)

        self.DataKeys = list(self.Data.keys())
```

Figura A.18: Clase del dataset multimodal en Afew

Se procede a entrenar el conjunto de train y test durante 20 épocas que van a permitir obtener diferentes métricas como el F1 score, exactitud, precisión y la función de pérdida. En este caso se van a recorrer tanto los ejemplos del conjunto de train y test que están formados por la data del rostro, audio y texto, con sus etiquetas y una variable availS para verificar si se encuentra o no disponible la modalidad dentro del conjunto. Como se muestra en las figuras A.19 y A.20

```
epochI = 0
epochF = 20

for epoch in range(epochI, epochF):
    Tacloss, Vacloss = 0, 0
    Tacacc, Vacacc = 0, 0
    Tcont, Vcont = 0, 0
    model.train()
    for sample in tqdm(train_dataloader):
        face_data = sample['face']
        audio_data = sample['audio']
        text_data = sample['text']
        label = sample['label'] #torch.argmax(, dim=-1)#.flatten()
        availS = sample['availabilities']

        optimizer.zero_grad()
        out = model([face_data, audio_data, text_data], availS)
        loss = criterion(out, label)
        loss.backward()
        optimizer.step()
        Tacloss += loss.item()
        Tacacc += (torch.argmax(out, dim=1) == label).float().sum()
        Tcont += face_data.shape[0]
```

Figura A.19: Entrenamiento multimodal en Afew

```
model.eval()
with torch.no_grad():
    for sample in tqdm(test_dataloader):
        face_data = sample['face']
        audio_data = sample['audio']
        text_data = sample['text']
        label = sample['label'] #torch.argmax(, dim=-1)#.flatten()
        availS = sample['availabilities']
        out = model([face_data, audio_data, text_data], availS)
        loss = criterion(out, label)

        Vacloss += loss.item()
        Vacacc += (torch.argmax(out, dim=1) == label).float().sum()
        Vcont += face_data.shape[0]

H_train_loss += [Tacloss/Tcont]
H_train_acc += [Tacacc/Tcont]

H_val_loss += [Vacloss/Vcont]
H_val_acc += [Vacacc/Vcont]

print('[%3d/%3d] train_loss: %.4f | val_loss: %.4f | train_acc: %.4f | val_acc: %.4f'%
    (epoch+1, epochF, H_train_loss[-1], H_val_loss[-1], H_train_acc[-1], H_val_acc[-1]))
```

Figura A.20: Continuación Entrenamiento multimodal en Afew

En la siguiente figura A.21 se da a conocer las métricas del entrenamiento bajo la arquitectura del embracenet por cada época, donde se detalla la función de pérdida para el conjunto de train y test, como también la exactitud de ambos conjuntos.

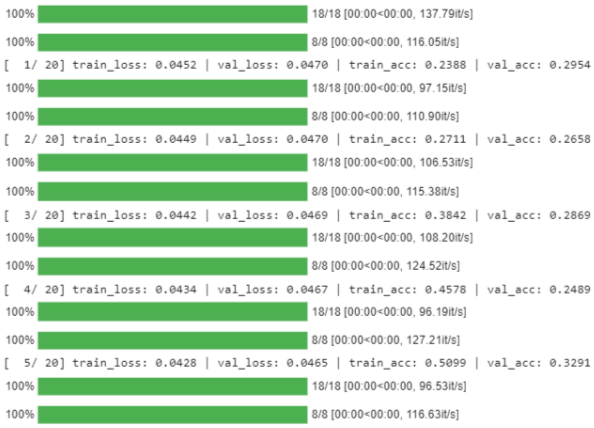


Figura A.21: Métricas Entrenamiento multimodal en Afew

Se generan 56958 imágenes en train y 16362 en test las cuales se van a ir alojando dentro de sus carpetas correspondientes como se muestra en la figura A.22

Figura A.22: Carpeta de frames train y test en Meld

Se realiza una clasificación manualmente de los 6 actores principales para el conjunto de train y test dejando un 80 % y un 20 % respectivamente. La data de entrenamiento son 2957 imágenes y 754 imágenes de test donde cada actor contiene su propia carpeta como se muestra en las figuras A.23 y A.24

Figura A.23: Carpeta Train y Test en Meld

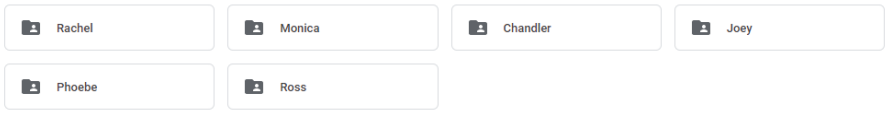


Figura A.24: Distribución de carpetas de actores de Train en Meld

En la figura A.25 se puede observar el formato de cada imagen, el cual es result_dia(id)_utt(id).(count).png, en donde los id están referenciados en los csv de cada conjunto de Train y Test y count se refiere a las cantidades de frames de cada video en particular.

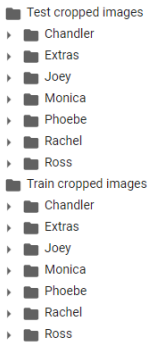


Figura A.25: Distribución de carpetas de Train y Test en Meld

En la figura A.26 se muestra el formato de cada imagen que es result_dia(id)_utt(id).(count).png, en donde los id están referenciados en los csv de cada conjunto de Train y Test donde representan el índice del diálogo y el índice de la expresión en particular. Por otro lado, count se refiere a las cantidades de frames de cada video en particular y todas en un formato png.

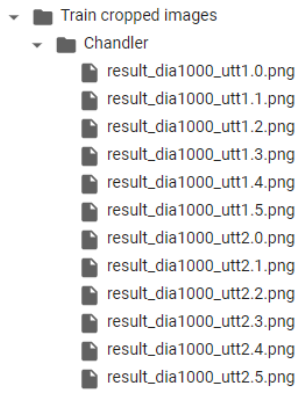


Figura A.26: Formato especifico de cada frames en Meld.

A partir de la figura A.27 lo que se va a crear es una clase para el dataset donde se establecen en un diccionario las 4 emociones principales(neutral, joy , anger y surprise) a utilizar. En base a ello se va recorriendo el csv de train y test excluyendo las emociones que no pertenecen al diccionario y balanceando hasta un cierto limite la emoción neutral al tener muchos ejemplares dentro del csv.

```
class Meld(Dataset):
    def __init__(self, roots='', csv_s='', categories={}, limit=900, transform=None):
        super(Meld, self).__init__()
        self.DataRoot = roots
        self.AnnotationFile = csv_s
        self.Transform = transform
        self.Categories = categories
        self.limit = limit
        self.load_data()

    def load_data(self):
        self.Data = {}
        self.Annotation = pd.read_csv(self.AnnotationFile)
        cont2=0

        for i, row in self.Annotation.iterrows():

            if row['Emotion'] not in list(self.Categories.keys()):
                continue

            if row['Emotion']=='neutral':
                    if cont2>self.limit :
                      continue
                    cont2+=1
            cont=0
```

Figura A.27: Clase del Dataset Modalidad Facial en Meld.

Por otro lado, al mismo tiempo en la figura A.28 se va recorriendo la carpeta en la que se alojan los frames recortados por actores, se van leyendo las imágenes, almacenando su nombre específico y asociándolo con la emoción que le pertenece dentro del csv. Cada imagen será redimensionada a 48X48 y se presentarán con su data, etiqueta y su nombre en específico.

```
        while True:
            true_name = row['Speaker'] + '/result_' + 'dia' + str(row['Dialogue_ID']) + '_utt' + str(row['Utterance_ID']) + '.' + str(cont) + '.png'
            x= cv2.imread(self.DataRoot + true_name)

            if x is None :
              print(self.DataRoot + true_name)
              break
            self.Data[true_name] = row['Emotion']
            cont+=1
    self.DataKeys = list(self.Data.keys())
    print(self.Data)

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

def make_shuffle(self):
    random.shuffle(self.DataKeys)

def __getitem__(self, idx):
    if torch.is_tensor(idx):
        idx = idx.tolist()
    label = self.Categories[self.Data[self.DataKeys[idx]]]
    data = cv2.imread(join(self.DataRoot, self.DataKeys[idx]))
    sample = {'data': data, 'label': label, 'name': self.DataKeys[idx]}

    if self.Transform:
      sample = self.Transform(sample)
    return sample
```

Figura A.28: Continuacion Clase Dataset Modalidad Facial en Meld.

Se va realizando el entrenamiento de las imágenes con el mismo procedimiento que en la figura A.10 y obteniendo sus métricas como la exactitud y la función de pérdida en ambos conjuntos a lo largo de las épocas establecidas donde se da a conocer en la figura A.29 los resultados que representan la capacidad para predecir ciertas emociones por parte del modelo y su calidad como tal y también la data del propio dataset.

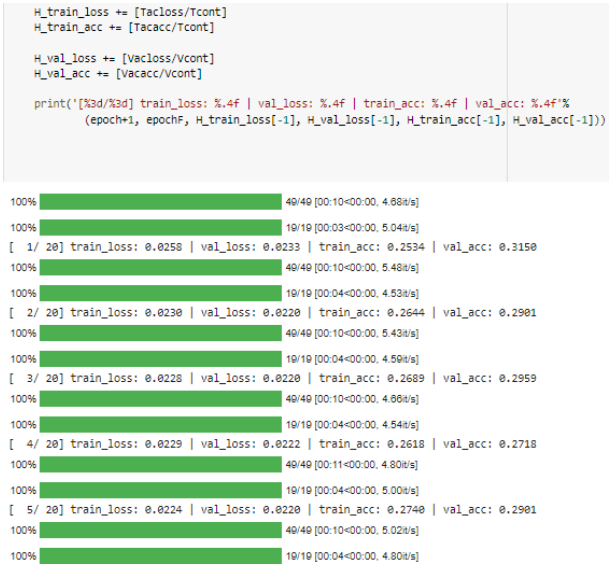


Figura A.29: Entrenamiento Modalidad Facial en Meld

Luego se creó la clase personalizada para el audio realizando el mismo procedimiento que en los otros dataset como se muestra en la figura A.13 lo único de diferente es que se va recorriendo el csv por las etiquetas que contiene y por los identificadores que conforman el nombre del archivo. También se puede destacar que la modalidad neutral se le realizó un balanceo con respecto a las otras emociones, ya que tenía muchos ejemplares. Como se muestra en la figura A.30.

```
class AudioDataset(Dataset):
    def __init__(self, roots='', csv_s='',categories={}, limit=5000, transform=None):
        super(AudioDataset, self).__init__()
        self.DataRoot = roots
        self.AnnotationFile=csv_s
        self.Transform = transform
        self.Categories = categories
        self.limit = limit
        self.LoadData()

    def LoadData(self):
        self.Data = {}
        self.Annotation = pd.read_csv(self.AnnotationFile)

        cont=0
        for i, row in self.Annotation.iterrows():

          if row['Emotion'] not in list(self.Categories.keys()):
              continue

          if row['Emotion']=='neutral':
              if cont>self.limit :
                  continue
              cont+=1

          true_name=join(self.DataRoot, 'dia' + str(row['Dialogue_ID']) + '_utt' + str(row['Utterance_ID']) + '.mp4.npy')
          self.Data[true_name] = row['Emotion']
```

Figura A.30: Clase Audio Meld

En la modalidad de texto, al tener en los csv cada texto de los videos, se procedió a realizar la clase personalizada, almacenando las emociones en un diccionario, balanceando la emoción neutral y recorriendo el csv en la columna donde se encuentra el texto y las etiquetas para asociarlas cada una de ellas. Todo lo mencionado se representa en la figura A.31

```
class TextDataset(Dataset):
    def __init__(self, csv_s='',categories={}, limit=5000, transform=None):

        super(TextDataset, self).__init__()
        self.AnnotationFile=csv_s
        self.Transform = transform
        self.Categories = categories
        self.limit = limit
        self.LoadData()

    def LoadData(self):
        self.Data = {}
        self.Annotation = pd.read_csv(self.AnnotationFile)
        cont=0

        for i, row, in self.Annotation.iterrows():

          if row['Emotion'] not in list(self.Categories.keys()):
              continue

          if row['Emotion']=='neutral':
              if cont>self.limit :
                  continue
              cont+=1

          true_name='dia' + str(row['Dialogue_ID']) + '_utt' + str(row['Utterance_ID'])
          self.Data[true_name]=(row['Utterance'],row['Emotion'])
        self.DataKeys=list(self.Data.keys())
```

Figura A.31: Clase Personalizada Texto en Meld

Lo siguiente es pasar por el embracenet+ todas las modalidades presentes en el dataset, por lo que primero se realiza una clase del dataset con diferentes parámetros como el diccionario de clases, las predicciones de las tres modalidades y el método avg para calcular el promedio de cada una de las métricas de evaluación. Dando paso a la función load data que va recibiendo los resultados de cada modalidad por parámetros y luego los va recopilando en una lista por cada uno de ellos como se muestra en la figura A.32, considerando cada modalidad a través de los condicionales verificando si la key corresponde al nombre del archivo con su formato en específico para llegar y guardarlo en la lista definida en la función.

```
class DatasetMELD(Dataset):
    def __init__(self, classes, FaceR, AudioR, TextR, csv_s='', method='avg', transform=None):
        super(DatasetMELD, self).__init__()
        self.Transform = transform
        self.Classes = classes
        self.AnnotationFile=csv_s
        self.Method = method
        self.loadData(FaceR, AudioR, TextR)

    def loadData(self, face_results, audio_results, text_results):

        LFks = list(face_results.keys())
        LAks = list(audio_results.keys())
        LTks = list(text_results.keys())

        self.Data = {}
        self.Annotation = pd.read_csv(self.AnnotationFile)
        for i, row, in self.Annotation.iterrows():

            if row['Emotion'] not in self.Classes:
                continue
            key= 'dia' + str(row['Dialogue_ID']) + '_utt' + str(row['Utterance_ID'])

            if key in LFks:
                FD = self.convert(face_results[key][0])
            else:
                FD = None

            if key in LTks:
                TD = text_results[key][0]
            else:
                TD = None
            if ('/content/mel/Train/' + key + '.mp4.npy') in LAks:
                AD = audio_results['/content/mel/Train/' + key + '.mp4.npy'][0].numpy()

            else:
                AD =None

            if (FD is None) and (TD is None) and (AD is None):
              continue

            self.Data[key] = (self.Classes[row['Emotion']],
                              FD, AD, TD)

        self.DataKeys = list(self.Data.keys())
```

Figura A.32: Clase del Dataset Meld Multimodal

En la siguiente figura A.33 se muestra el desempeño y la calidad del modelo a la hora de predecir las emociones bajo las diferentes épocas que se va entrenando el conjunto de train y test.

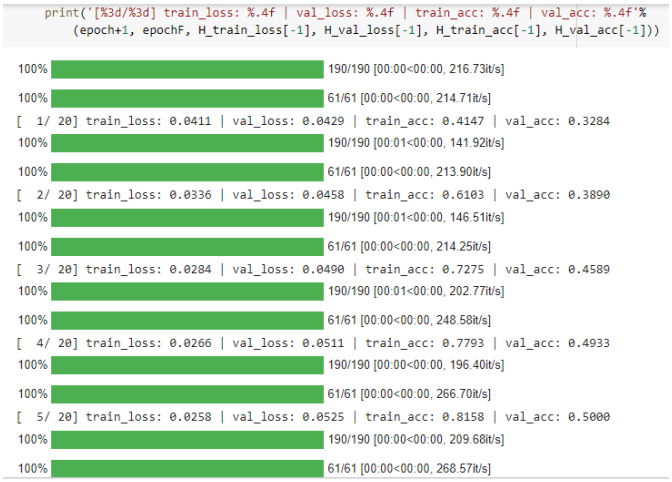


Figura A.33: Métricas Entrenamiento Multimodal Meld

A.8. Implantación

A.8.1. Producto Final

Tal como se ha mencionado anteriormente, en este trabajo se realizó una investigación aplicada con respecto a diferentes dataset en un contexto de robótica social, realizando diversas evaluaciones con el fin de adaptarlos a la arquitectura ya implementada con deep learning para las diferentes modalidades. El producto final que se desarrolló durante todo el 2022 fue trabajar por separado los dataset en diferentes colab para realizar el preprocesamiento correspondiente, adaptar la entrada de datos a la arquitectura, procesar independientemente las modalidades que contiene el dataset, obtener las métricas de desempeño representado a través de una matriz de confusión, un clasification report y diversos gráficos que dan a conocer el comportamiento del entrenamiento y de las pruebas, todo esto para comparar e interpretar los resultados.

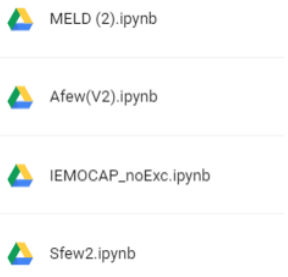


Figura A.34: Colab por cada datasets

Lo que se buscó es saber que dataset, emoción y modalidad son las más recomendables para llevarlo a un contexto de robótica social. En donde todo este procedimiento se realizó para los dataset a evaluar y de forma ordenada se llevó a un GitHub explicando de forma detallada el contenido de cada dataset trabajado. Desde siempre, el usuario final para estos dataset son investigadores que trabajan en esta área o para fines académicos.

A.8.2. Implementación en Producción

En la siguiente figura A.35 se muestran los diferentes dataset evaluados dentro de este Trabajo de investigación en los cuales estarán disponibles para los participantes de este proyecto en forma privada en GitHub. En la dirección https://github.com/FelipeRojas1998/SentiRobots, donde se dispone varias carpetas por cada dataset donde se encuentran los colab que contienen todo el código trabajado como se aprecia en la figura A.36

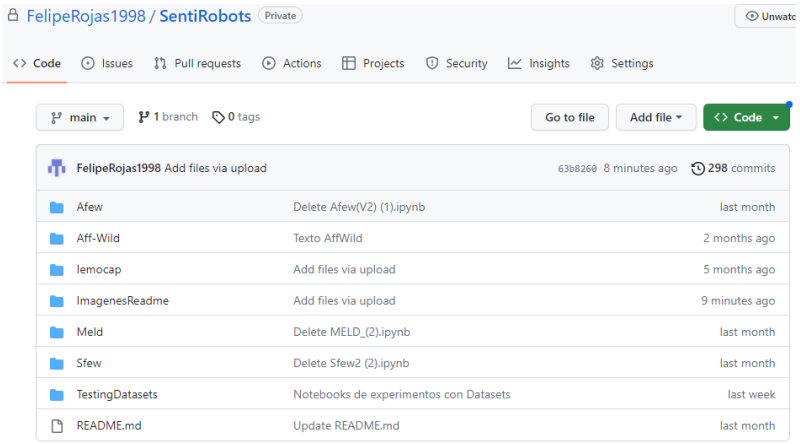


Figura A.35: Github donde se hospedan los datasets

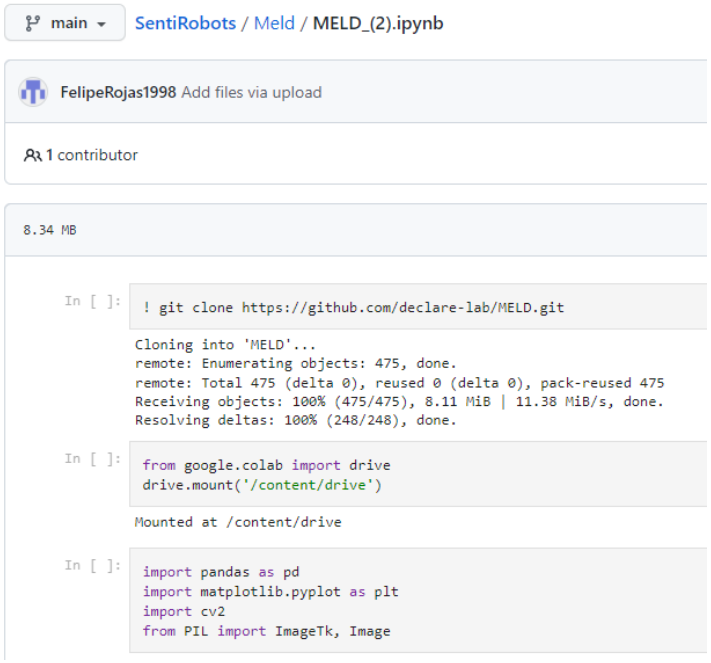


Figura A.36: Contenido Carpeta Meld

En la siguiente figura A.37 se tiene una carpeta llamada TestingDatasets que contiene los colab de cada uno de los dataset trabajados, pero con la diferencia de que se realizó un preprocesamiento único para tener una estandarización y realizar pruebas unimodales como multimodales con todas las emociones para publicarlo en un artículo científico de la mejor forma posible.

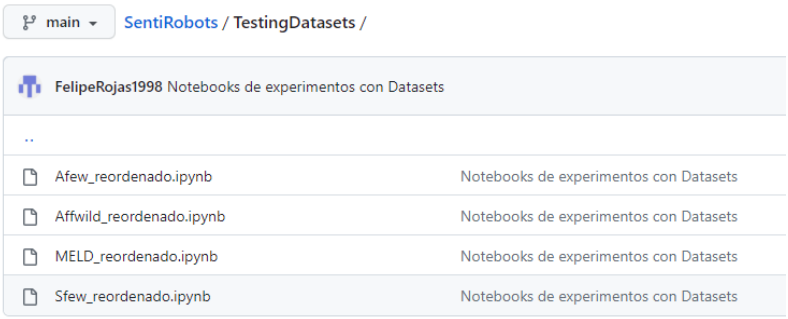


Figura A.37: Carpeta con los Datasets Reordenados

Además de las carpetas de cada dataset con su código respectivo, se encuentra otra carpeta llamada ImagenesReadme con el mismo organigrama de carpetas, pero para almacenar las imágenes que contiene el archivo readme como se muestra en la figura A.38.

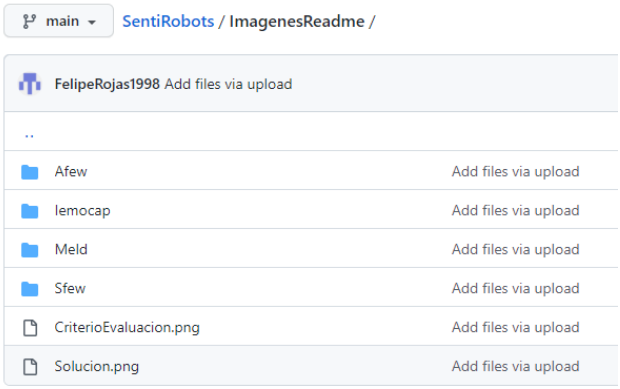


Figura A.38: Imágenes Readme

En la siguiente figura A.39 se considera como ejemplo la carpeta meld en la cual se alojan las imágenes a utilizar en el archivo readme.

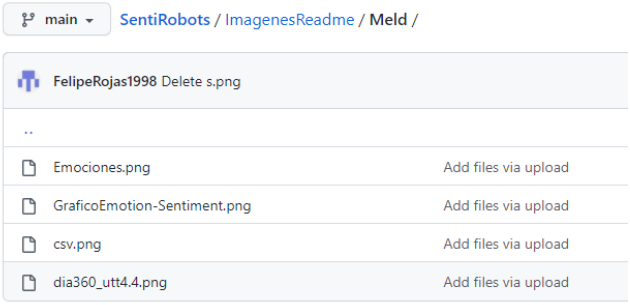


Figura A.39: Imágenes Readme Meld

Por último, se encuentra el archivo readme que consiste en dar una explicación a modo resumen de lo que se realizó. Como se muestra en la siguiente figura A.40 da a conocer el contexto del trabajo realizado como también la propuesta de la solución explicada en un diagrama.

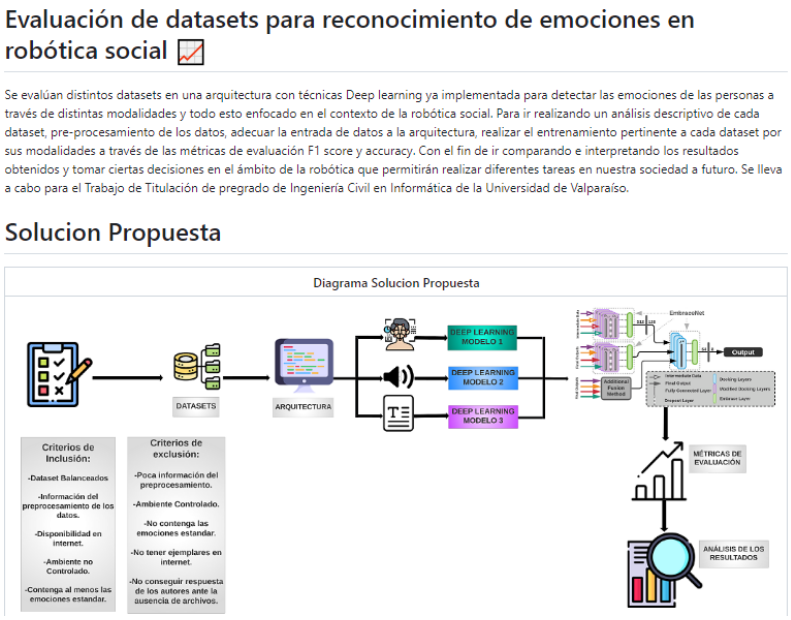
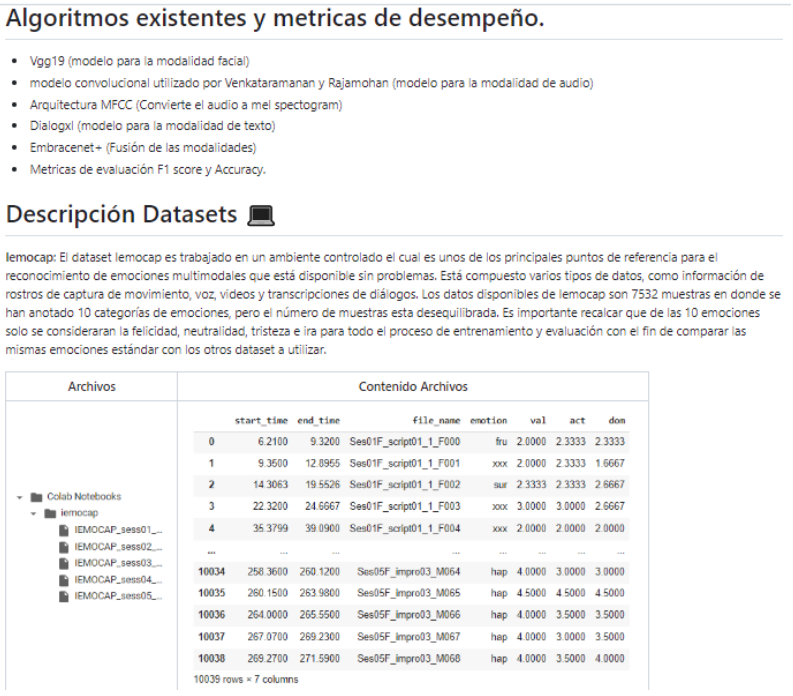


Figura A.40: Archivo Readme Parte1

En la figura A.41 se mencionan los algoritmos y métricas de desempeño que se utilizaron para las distintas modalidades durante el trabajo de investigación y destacando los dataset utilizados con sus características más importantes, cantidades de emociones, imágenes y el preprocesamiento a modo resumen.



Algoritmos existentes y metricas de desempeño.

- Vgg19 (modelo para la modalidad facial)
- modelo convolucional utilizado por Venkataramanan y Rajamohan (modelo para la modalidad de audio)
- Arquitectura MFCC (Convierte el audio a mel spectogram)
- Dialogxl (modelo para la modalidad de texto)
- Embracenet+ (Fusión de las modalidades)
- Metricas de evaluación F1 score y Accuracy.

Descripción Datasets 💻

Iemocap: El dataset Iemocap es trabajado en un ambiente controlado el cual es unos de los principales puntos de referencia para el reconocimiento de emociones multimodales que está disponible sin problemas. Está compuesto varios tipos de datos, como información de rostros de captura de movimiento, voz, videos y transcripciones de diálogos. Los datos disponibles de Iemocap son 7532 muestras en donde se han anotado 10 categorías de emociones, pero el número de muestras esta desequilibrada. Es importante recalcar que de las 10 emociones solo se consideraran la felicidad, neutralidad, tristeza e ira para todo el proceso de entrenamiento y evaluación con el fin de comparar las mismas emociones estándar con los otros dataset a utilizar.

Archivos

Contenido Archivos

Colab Notebooks
iemocap
IEMOCAP_sess01_...
IEMOCAP_sess02_...
IEMOCAP_sess03_...
IEMOCAP_sess04_...
IEMOCAP_sess05_...

	start_time	end_time	file_name	emotion	val	act	dom
0	6.2100	9.3200	Ses01F_script01_1_F000	fru	2.0000	2.3333	2.3333
1	9.3500	12.8955	Ses01F_script01_1_F001	xxx	2.0000	2.3333	1.6667
2	14.3063	19.5526	Ses01F_script01_1_F002	sur	2.3333	2.3333	2.6667
3	22.3200	24.6667	Ses01F_script01_1_F003	xxx	3.0000	3.0000	2.6667
4	35.3799	39.0900	Ses01F_script01_1_F004	xxx	2.0000	2.0000	2.0000
...	...	...	...	...	...	...	...
10034	258.3600	260.1200	Ses05F_impro03_M064	hap	4.0000	3.0000	3.0000
10035	260.1500	263.9800	Ses05F_impro03_M065	hap	4.5000	4.5000	4.5000
10036	264.0000	265.5500	Ses05F_impro03_M066	hap	4.0000	3.5000	3.5000
10037	267.0700	269.2300	Ses05F_impro03_M067	hap	4.0000	3.0000	3.5000
10038	269.2700	271.5900	Ses05F_impro03_M068	hap	4.0000	3.5000	4.0000

10039 rows × 7 columns

Figura A.41: Archivo Readme Parte2

Posteriormente, en la figura A.42 se menciona la información relevante de cada conjunto de datos con las características más importantes que se encontraron al realizar el preprocesamiento inicial y se presentan los criterios de evaluación para definir que dataset es el más recomendable a utilizar dentro de un contexto de robótica social.

Informacion Relevante

Dataset	Características
Iemocap	-Ambiente Controlado, contiene las 3 modalidades, Baja calidad y precisión en las predicciones de emociones.
Sfew	-Una modalidad y ausencia de un enfoque multimodal, Ambiente no controlado, Base de datos de expresiones faciales.
Afew	-Ambiente no controlado, tres modalidades (audio,texto y rostro) y con un enfoque multimodal, conjunto original de train y test con videos repetidos.
Meld	-Ambiente no controlado, contiene tres modalidades (rostro,audio y texto), conjunto de Train y Test en csv, videos con un formato especifico donde participan muchas personas que no están etiquetadas, contiene emociones y sentimientos.
Aff-Wild2	-Ambiente no controlado, En principio contiene 1 modalidad con una etiquetación de las emociones solo en la modalidad de rostros, sus videos son reacciones a ciertas películas en donde existen cierto ruido, poca luminosidad para captar las diversas emociones y conversaciones en ciertos momentos.

Criterio de evaluación

La finalidad es verificar que el dataset que cumpla con los criterios de evaluación que se muestran a continuación en una tabla a modo resumen con el fin de dar cuenta que dataset es el más adecuado para trabajar en un contexto de robótica social en base a los criterios previamente establecidos. Paralelamente esta información va a ayudar a responder las preguntas de investigación que necesariamente son importantes para la discusión de resultados.

Criterio de evaluación

	Iemocap	Sfew	Afew	Meld	Aff-Wild2
Cantidad de Modalidades	✓	✗	✓	✗	✓
Emociones Estandar	✓	✓	✓	✗	✓
Enfoque Multimodal	✓	✗	✓	✓	✓
Datos Balanceados	✗	✓	✓	✗	✗
Ambiente No controlado	✗	✓	✓	✓	✓
Calidad de train y test	✓	✗	✗	✗	✗
Métricas de desempeño	✓	✗	✗	✓	✗

Figura A.42: Archivo Readme Parte3

Para terminar el contenido del archivo readme en la figura A.43 se explica el contenido del GitHub y las instrucciones para acceder a los diferentes dataset pidiendo permisos a las personas autorizadas donde se encuentra el link correspondiente a cada dataset con toda la información entregada y las diferentes referencias que se tomaron como base para investigar y obtener información de utilidad para el trabajo.

Contenido del Github a disposición

En la carpeta raiz de SentiRobots se encuentran todos los dataset que fueron evaluados donde cada uno de ellos esta en una carpeta especifica donde se aloja el colab con una extensión .ipynb , donde se desarrollo todo el pre-procesamiento de los datos, adecuacion de la entrada de datos, entrenamiento del conjunto de datos como también las métricas de evaluación para el enfoque unimodal y multimodal correspondiente.

Omitir la carpeta SentiRobots/ImagenesReadme donde se encuentran las imagenes por cada dataset del analisis descriptivo que se llevo acabo y se ingreso al Readme.md

Acceso a los datasets completos.

Cualquier interesado en acceder a los diferentes dataset que se encuentran en este trabajo para ser usado con fines académicos o de investigación, puede contactarse con los encargados de este dataset que son felipe.rojasg@alumnos.uv.cl, diego.mellado@uv.cl o la profesora encargada de este trabajo que es ana.aguilera@uv.cl. Solicitando los permisos para acceder al contenido con el fin de compartir la carpeta asociada con el interesado/a.

En estos links se encuentran todos los archivos de los diferentes datasets trabajado.

Dataset Iemocap: https://drive.google.com/drive/u/1/folders/1sQg2HobxBM_i65ls7IMw0WmeuBkUH20S

Dataset Sfew: https://drive.google.com/drive/u/1/folders/19Cg-oB_MkdBbVCrIW9ohPd6c0KNnhy4c

Dataset Afew: https://drive.google.com/drive/u/1/folders/1c5S5vH0zB8slWVJxL2toT4XpUwpkZmOz

Dataset Meld: https://drive.google.com/drive/u/1/folders/1US5Qr0F-QnY2QV0o1irlV9KWJ2JH3-Cu

Referencias

[1] Venkataramanan and H. R. Rajamohan, "Emotion recognition from speech," Ar-Xiv, vol. abs/1912.10458, 2019.

[2] Heredia, E. Lopes-Silva, Y. Cardinale, J. Diaz-Amado, I. Dongo, W. Graterol, and A. Aguilera, "Adaptive multimodal emotion detection architecture for social robots," IEEE Access, vol. 10, pp. 20 727–20 744, 2022.

[3] W. Shen, J. Chen, X. Quan, and Z. Xie, "Dialogxl: All-in-one xlnet for multi-party conversation emotion recognition," 2020. [Online]. Available: https: //arxiv.org/abs/2012.08695

Figura A.43: Archivo Readme Parte4

Por temas de políticas de privacidad de cada dataset no se recomienda hacer público el contenido de estos en GitHub, por esto cualquier interesado en usar los diferentes datasets completos con fines académicos o de investigación, puede solicitar mandando un correo a los diferentes colaboradores de estos dataset como son felipe.rojasg@alumnos.uv.cl, diego.mellado@uv.cl o a la profesora guía ana.aguilera@uv.cl.