Pontificia Universidad
JAVERIANA
Cali

Facultad de Ingeniería y Ciencias

Ingeniería Biomédica

Informe Final de Trabajo de Grado

Reconocimiento de expresiones faciales a partir de imágenes para posible soporte de ayuda en personas con Trastorno del Espectro Autista (TEA)

Miguel Santiago González Torres
Sebastián Pérez Ramírez

Director
Dr. Hernán Darío Vargas Cardona

15 de junio de 2025

Santiago de Cali, 15 de junio de 2025

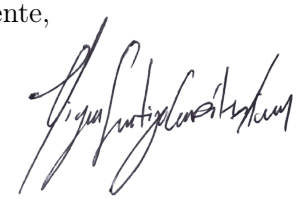
Señores
Pontificia Universidad Javeriana – Cali
Dr. Hernán Camilo Rocha Niño
Decano
Facultad de Ingeniería y Ciencias
Ciudad

Cordial Saludo.

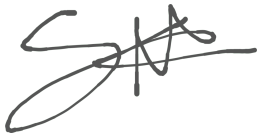
Por medio de la presente nos permitimos presentarle el Trabajo de Grado titulado "Reconocimiento de expresiones faciales a partir de imágenes para posible soporte de ayuda en personas con Trastorno del Espectro Autista (TEA)".

Esperamos que este trabajo reúna todos los requisitos académicos, cumpla el propósito para el cual fue creado y sirva de apoyo para futuros proyectos relacionados con la profesión.

Atentamente,



Miguel Santiago González Torres



Sebastián Pérez Ramírez

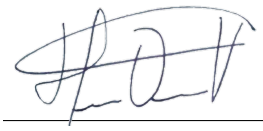
Santiago de Cali, 15 de junio de 2025

Señores
Pontificia Universidad Javeriana – Cali
Dr. Hernán Camilo Rocha Niño
Decano
Facultad de Ingeniería y Ciencias
Ciudad

Cordial Saludo.

Certifico que el presente Trabajo de Grado titulado "Reconocimiento de expresiones faciales a partir de imágenes para posible soporte de ayuda en personas con Trastorno del Espectro Autista (TEA)", realizado por Miguel Santiago González Torres y Sebastián Pérez Ramírez, estudiantes de Ingeniería Biomédica, se encuentra terminado y puede ser presentado para su sustentación.

Atentamente,



Dr. Hernán Darío Vargas Cardona
Director Trabajo de Grado

Agradecimientos

Queremos expresar nuestra profunda gratitud a todas las personas que hicieron posible la realización de este trabajo de grado. Este proyecto no solo representa un logro académico, sino también un proceso de aprendizaje, crecimiento personal y compromiso con el desarrollo de soluciones tecnológicas que aporten a la sociedad.

Agradecemos especialmente a nuestros familiares, quienes con su apoyo incondicional, palabras de aliento y paciencia nos impulsaron a seguir adelante, incluso en los momentos más exigentes. Su confianza en nosotros ha sido el motor que nos motivó a culminar este proceso con dedicación y responsabilidad.

A nuestros docentes de la Pontificia Universidad Javeriana de Cali, quienes, a través de sus enseñanzas y orientación, sembraron en nosotros el interés por la investigación, la tecnología y la empatía social. En especial, extendemos nuestro agradecimiento al profesor Hernán Darío Vargas Cardona, por su acompañamiento, asesoría constante y valiosos aportes técnicos y humanos que enriquecieron significativamente este proyecto.

A la universidad, por brindarnos los recursos académicos, tecnológicos y logísticos necesarios para llevar a cabo este trabajo. Valoramos profundamente el acceso a laboratorios, bases de datos, espacios de consulta, seminarios y el acompañamiento institucional durante todo el proceso.

Finalmente, agradecemos a cada persona que, de una u otra forma, contribuyó con palabras, ideas, pruebas, sugerencias o colaboración práctica. Cada gesto de apoyo fue una parte fundamental para que este trabajo se concretara en una herramienta funcional con potencial de impacto social.

Glosario

Términos

Algoritmo	Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.
Amígdala	Estructura cerebral involucrada en la gestión de emociones y respuestas emocionales.
Deep Learning	Es un subconjunto del machine learning que utiliza redes neuronales multicapa, llamadas redes neuronales profundas, para simular el complejo poder de toma de decisiones del cerebro humano.
Machine Learning	Ciencia de desarrollo de algoritmos y modelos estadísticos que utilizan los sistemas de computación con el fin de llevar a cabo tareas sin instrucciones explícitas, en vez de basarse en patrones e inferencias.
Base de datos	Conjunto organizado de imágenes de expresiones faciales utilizadas para entrenar, validar y probar el modelo de reconocimiento emocional.
CNN	(Convolutional Neural Network) Es una arquitectura de red neuronal especializada en el análisis de datos visuales. Extrae características jerárquicas de imágenes, como expresiones faciales.
Código fuente	Conjunto de instrucciones escritas en un lenguaje de programación, que ejecutan funciones específicas del sistema propuesto.
Corteza cingulada anterior	región del cerebro que forma parte del sistema límbico y está involucrada en funciones como la regulación de la atención, las emociones, el control inhibitorio, el monitoreo de errores y la motivación
Corteza prefrontal	Región cerebral asociada a la regulación emocional, la toma de decisiones y el comportamiento social. Su desarrollo es clave en habilidades sociales.
Entrenamiento	Fase del proceso de aprendizaje automático en la cual un modelo ajusta sus parámetros internos utilizando un conjunto de datos conocidos (datos de entrenamiento).
Exactitud (Accuracy)	Métrica que mide el porcentaje de predicciones correctas de un modelo en relación con el total de casos evaluados.
F1-Score	Métrica que combina precisión y exhaustividad (recall) en una sola medida para evaluar el rendimiento de modelos de clasificación.
GPU	Unidad de procesamiento gráfico utilizada para acelerar el entrenamiento de modelos de deep learning debido a su capacidad de cómputo paralelo.
Inteligencia Artificial	Disciplina que desarrolla sistemas capaces de simular comportamientos inteligentes. En este proyecto, se aplica para el reconocimiento emocional automatizado.

Kaggle	Plataforma en línea que proporciona datasets y herramientas para ciencia de datos. Fuente de las bases de datos usadas en este proyecto.
Neurodesarrollo	Proceso mediante el cual el sistema nervioso, especialmente el cerebro, crece, madura y adquiere sus funciones, desde la gestación hasta la edad adulta.
Pérdida (Loss)	Medida cuantitativa del error entre las predicciones realizadas por un modelo y los valores reales esperados. La función de pérdida guía el proceso de entrenamiento, ya que el objetivo del modelo es minimizar esta pérdida ajustando sus parámetros.
Phyton	Lenguaje de programación de alto nivel utilizado para desarrollar el software del proyecto.
Red neuronal	Modelo computacional inspirado en las neuronas del cerebro humano, que se emplea para aprender patrones complejos en datos.
Red neuronal profunda	Tipo de red con múltiples capas ocultas que permiten un aprendizaje jerárquico de características complejas. Es la base del deep learning.
Recall	Conocido como sensibilidad o tasa de verdaderos positivos, es una métrica usada principalmente en problemas de clasificación para evaluar qué tan bien un modelo identifica correctamente los casos positivos.
ResNet 50	Arquitectura de red neuronal profunda con bloques residuales, diseñada para facilitar el entrenamiento de redes muy profundas y mejorar el rendimiento.
ReLU	(Rectified Linear Unit) Es una función de activación utilizada en redes neuronales para introducir no linealidad y acelerar el aprendizaje.
SVM	(Support Vector Machine) Es un algoritmo de clasificación utilizado para distinguir entre diferentes emociones en el análisis automático de expresiones faciales.
Validación	Proceso mediante el cual se evalúa el rendimiento de un modelo de aprendizaje automático utilizando datos que no fueron empleados durante el entrenamiento, con el fin de medir su capacidad de generalización.
VGG16	Arquitectura de red neuronal convolucional profunda que se caracteriza por su simplicidad estructural. Usada como base para modelos de clasificación de imágenes.

Resumen

Este proyecto de grado se centra en el desarrollo de una aplicación software diseñada para la el reconocimiento de expresiones faciales a partir de imagenes para personas con trastorno de espectro autista (TEA), con un enfoque en los niños, buscando la posible mejora de sus habilidades sociales mediante el reconocimiento efectivo de emociones básicas. El núcleo de la aplicación software radica en el uso de técnicas de deep learning, mediante las cuales el sistema aprenderá a identificar emociones a partir de una extensa base de datos de expresiones faciales. La fase de entrenamiento implicará el procesamiento de imágenes de diversas emociones, permitiendo que el prototipo desarrolle un modelo de reconocimiento que se adapte a las variaciones en las expresiones faciales propias de las personas. Una vez entrenado, el aplicativo se utilizará en los autores del proyecto, donde podrá validar y corroborar las emociones detectadas en las imágenes capturadas durante las actividades. Esto permitirá una retroalimentación de calidad, reforzando el aprendizaje en la identificación correcta de las emociones.

Palabras clave: Trastorno del Espectro Autista (TEA), prototipo funcional, habilidades sociales, reconocimiento de emociones, deep learning, terapia infantil, inteligencia artificial, procesamiento de emociones.

Abstract

This degree project is centered on the development of a software application engineered for the recognition of facial expressions from images in individuals with Autism Spectrum Disorder (ASD), with a particular emphasis on children. Its aim is to potentially enhance their social skills through the accurate detection of basic emotions. At the heart of the application lies the deployment of deep learning techniques, whereby the system will be trained to identify emotions using an extensive facial-expression dataset. During the training phase, images representing a range of emotional states will be processed, enabling the prototype to construct a recognition model capable of accommodating the inherent variability in individuals' facial expressions. Once training is complete, the application will be employed by the project authors, allowing them to validate and confirm the emotions detected in the images captured throughout various activities. This will yield high-quality feedback, thereby reinforcing the learning process for correctly identifying emotional states.

Keywords: Autism Spectrum Disorder (ASD), functional prototype, social skills, emotion recognition, deep learning, child therapy, artificial intelligence, emotion processing.

Keywords: Autism Spectrum Disorder (ASD), functional prototype, social skills, emotion recognition, deep learning, child therapy, artificial intelligence, emotion processing.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

La comprensión y expresión de las emociones son aspectos fundamentales para el desarrollo social y emocional de cualquier individuo; sin embargo, los niños con Trastorno del Espectro Autista (TEA) enfrentan retos considerables en este ámbito. La interpretación de expresiones faciales y la conexión emocional con otros suelen ser áreas de especial dificultad, lo cual afecta tanto su capacidad para entablar relaciones significativas como su integración en contextos educativos y sociales. Estos desafíos no solo repercuten en la infancia, sino que pueden persistir a lo largo de la vida, limitando su desarrollo social y personal.

Este proyecto busca atender esta necesidad mediante la creación de un aplicativo software que emplee técnicas avanzadas de deep learning para una posible herramienta asistencial para la identificación de emociones básicas. La aplicación de esta tecnología permite automatizar el reconocimiento de expresiones emocionales, facilitando la interacción con el mundo real y ofreciendo a las personas una experiencia de aprendizaje continua y adaptada a sus necesidades. Esta herramienta no solo apunta a mejorar la capacidad de interpretación emocional, sino que también ofrece un recurso accesible para terapeutas y familias, promoviendo una intervención temprana que optimice el desarrollo social y facilite la integración en la sociedad.

Capítulo 2

Planteamiento del Problema

2.1. Planteamiento del Problema

El Trastorno del Espectro Autista (TEA) es una condición del neurodesarrollo que se manifiesta de manera heterogénea en los individuos que lo experimentan. Este trastorno impacta significativamente áreas fundamentales del desarrollo, siendo la comunicación, la interacción social y el comportamiento las más afectadas. Los niños con TEA pueden presentar una amplia variedad de síntomas, desde dificultades en el lenguaje y la expresión emocional hasta patrones repetitivos y estereotipados de conducta. A nivel fisiológico, el desarrollo de las habilidades sociales en los niños, durante los primeros años de vida, especialmente en la infancia temprana, el cerebro experimenta un rápido crecimiento y desarrollo sináptico, formando conexiones neuronales fundamentales para las funciones cognitivas y socioemocionales [1].

En el contexto de las habilidades sociales, varias áreas cerebrales están implicadas, y la falta de estimulación adecuada durante la infancia puede afectar negativamente su desarrollo. La corteza prefrontal, esencial para la regulación emocional y la toma de decisiones sociales, se beneficia de la estimulación temprana, contribuyendo al desarrollo de funciones ejecutivas necesarias para comportamientos sociales apropiados. La amígdala, involucrada en la respuesta emocional y la percepción de las emociones de los demás, también se ve afectada por la falta de estimulación temprana, afectando la regulación emocional y la interpretación adecuada de las señales sociales. La corteza cingulada anterior, crucial para la empatía y la comprensión de las intenciones de los demás, puede desarrollarse de manera más efectiva con experiencias sociales positivas en la infancia. Asimismo, el sistema de espejo neuronal, que permite la imitación y el aprendizaje social, se beneficia de la estimulación temprana, permitiendo a los niños aprender a través de la observación y la imitación de comportamientos sociales. Cuando los niños no reciben la estimulación adecuada durante este periodo crítico de desarrollo cerebral, estas áreas pueden no alcanzar su pleno potencial y la falta de exposición a estímulos sociales y emocionales puede llevar a un desequilibrio en el desarrollo de estas estructuras cerebrales, resultando en dificultades para interpretar las señales sociales, expresar emociones, establecer conexiones empáticas y participar en interacciones sociales recíprocas[2].

Es crucial señalar que cada niño es único, y el impacto de la falta de estimulación temprana puede variar. Sin embargo, proporcionar un entorno enriquecido con experiencias sociales y emocionales positivas durante la infancia temprana puede tener efectos beneficiosos en el desarrollo de

las habilidades sociales, incluso en niños con TEA. Tanto así que, de acuerdo con el informe de la Organización Mundial de la Salud (OMS), las habilidades sociales desempeñan un papel crucial en la vida cotidiana de los individuos, contribuyendo al bienestar emocional, la adaptación social y el éxito en las interacciones sociales. Sin embargo, como se mencionó anteriormente, los niños con TEA enfrentan dificultades significativas en el desarrollo y la expresión de estas habilidades, lo cual tiene repercusiones a lo largo de su vida. En el contexto colombiano, la falta de programas especializados y recursos destinados a abordar las habilidades sociales de los niños con TEA se convierte en un problema emergente. Aunque existen esfuerzos en la atención médica y educativa, aún persisten brechas en la identificación temprana y la implementación de intervenciones específicas para mejorar las habilidades sociales en este grupo poblacional.

Es importante resaltar que el desarrollo adecuado de habilidades sociales en niños con TEA no solo impacta su calidad de vida, sino que también influye en su integración social y académica. La carencia de estrategias efectivas para abordar este problema puede llevar a consecuencias a largo plazo, como dificultades en la formación de relaciones interpersonales, limitaciones en la participación educativa y laboral, y una mayor probabilidad de enfrentar discriminación social [3].

2.1.1. Formulación

¿Cómo puede la ingeniería biomédica contribuir al diseño y desarrollo de tecnologías innovadoras que faciliten la posible intervención temprana y mejoren las habilidades sociales en niños con Trastorno del Espectro Autista (TEA) en Colombia, considerando las necesidades específicas de esta población y los recursos disponibles en el país?

2.1.2. Sistematización

1. ¿Cómo puede gestionarse y procesarse la base de datos para garantizar su compatibilidad con el entrenamiento y validación de un sistema de reconocimiento de expresiones faciales orientado a niños con TEA?

2. ¿Cuáles técnicas de aprendizaje profundo son más efectivas para entrenar modelos que permitan el reconocimiento preciso y confiable de emociones básicas en personas con TEA?

3. ¿Cómo se pueden evaluar y comparar diferentes algoritmos de reconocimiento de emociones para identificar el más confiable y eficiente en el contexto del TEA?

4. ¿Cómo puede integrarse el reconocimiento de emociones en un sistema práctico para contribuir al desarrollo de habilidades sociales en personas con TEA?

Capítulo 3

Justificación

La investigación se realiza para abordar una necesidad urgente y fundamental en la comunidad: la mejora de las habilidades sociales de los niños con Trastorno del Espectro Autista (TEA). Los niños con TEA enfrentan desafíos significativos en el desarrollo de habilidades sociales y comunicativas, lo que afecta su capacidad para interactuar con el mundo que les rodea [1]. Esta investigación busca comprender mejor las dificultades específicas que enfrentan estos niños en sus interacciones sociales y proponer soluciones innovadoras y efectivas para mejorar su calidad de vida.

La propuesta de utilizar tecnologías innovadoras, como el uso de la inteligencia artificial, para mejorar las habilidades sociales de los niños con TEA es altamente viable. La investigación previa, como el proyecto ERIK, ha demostrado el potencial de las tecnologías robóticas para mejorar la interacción social de los niños con TEA [4]. Además, la disponibilidad de dispositivos electrónicos de fácil acceso, como computadores, celulares y tablets, hace que la implementación de estas tecnologías sea práctica y alcanzable.

Adicionalmente, las herramientas específicas para esta población son limitadas y, en muchos casos, costosas, lo que dificulta su acceso a familias de bajos recursos [5]. Este proyecto no solo busca mejorar las habilidades sociales en niños con TEA, sino también contribuir a la reducción de las brechas en la inclusión social y educativa mediante una solución económica y adaptable a las condiciones locales. La pertinencia de esta propuesta radica en su capacidad para atender a una población desatendida, aprovechando las tecnologías actuales para potenciar su desarrollo personal y social, alineándose con iniciativas internacionales de inclusión y bienestar infantil promovidas por entidades como la Organización Mundial de la Salud (OMS) [6].

El uso de algoritmos de aprendizaje profundo para el reconocimiento de emociones ha demostrado ser una herramienta efectiva en terapias para niños con TEA, al facilitar una mayor precisión en la identificación de expresiones emocionales y proporcionar retroalimentación personalizada. Estudios recientes han señalado que los sistemas basados en inteligencia artificial potencian la capacidad de los niños para comprender y expresar emociones, promoviendo interacciones sociales más fluidas [4].

Además, investigaciones en entornos terapéuticos han destacado la importancia de combinar

tecnologías accesibles con métodos diseñados específicamente para niños con TEA, permitiendo una intervención más inclusiva y efectiva. Por ejemplo, la implementación de dispositivos de bajo costo con componentes modulares ha permitido ampliar el acceso a terapias en regiones con recursos limitados, generando un impacto significativo en poblaciones vulnerables [7].

Desde el punto de vista técnico y económico, la viabilidad del proyecto está respaldada por la disponibilidad de herramientas de desarrollo accesibles, como Google Colab y Python, junto con bases de datos de expresiones faciales previamente establecidas y modelos de aprendizaje profundo validados. Esto permite minimizar los costos de implementación iniciales y enfocar los recursos en la optimización y adaptación del prototipo a las necesidades de la población objetivo. Asimismo, el equipo de trabajo cuenta con el apoyo de especialistas en ingeniería biomédica, deep learning y electronica, lo que asegura un desarrollo riguroso y orientado a resultados aplicables.

Capítulo 4

Objetivos

4.1. Objetivo General

Diseñar e implementar una aplicación digital para el reconocimiento de expresiones faciales con deep learning que pueda ser aplicado como posible soporte de ayuda en personas con Trastorno del Espectro Autista (TEA) en Colombia.

4.2. Objetivos Específicos

1. Gestionar bases de datos para usar la información relevante sobre emociones básicas, organizando y procesando la misma para garantizar su compatibilidad con el entrenamiento y validación del sistema.

2. Implementar algoritmos de aprendizaje profundo que permita el reconocimiento de emociones básicas usando la base de datos como entrada y validación.

3. Evaluar el rendimiento de los algoritmos de reconocimiento de emociones utilizando métricas estandarizadas en el estado del arte para clasificación de modo que se identifique el método más exacto para la aplicación final.

4. Desarrollar una aplicación funcional que integre el método de reconocimiento de emociones seleccionado con una interfaz gráfica y que posiblemente pueda ser usada por personas con TEA.

Capítulo 5

Marco de Referencia

5.1. Áreas Temáticas

- Ingeniería Biomédica: Aplicación de principios de ingeniería para el diseño de tecnologías enfocadas en la mejora de la salud y bienestar, especialmente en el contexto de intervenciones para niños con TEA.
- Neurodesarrollo y Trastorno del Espectro Autista (TEA): Estudio de los procesos de desarrollo neurológico y comprensión de los desafíos específicos que enfrenta la población con TEA, enfocados en el reconocimiento emocional y habilidades sociales.
- Inteligencia Artificial y Deep Learning: Uso de técnicas de inteligencia artificial, específicamente deep learning, para el procesamiento y análisis de expresiones faciales y el reconocimiento de emociones.
- Reconocimiento de Emociones: Desarrollo y aplicación de modelos para identificar y clasificar expresiones emocionales básicas, facilitando la interacción social de personas con TEA.
- Tecnologías de Asistencia: Creación de herramientas y dispositivos de bajo costo enfocados al posible apoyo en terapias y estimulación en niños con necesidades especiales, promoviendo su inclusión y desarrollo.
- Psicología Infantil y Habilidades Sociales: Estudio de las habilidades sociales y emocionales en la infancia, en particular el impacto del TEA en estas áreas y la intervención para mejorar la empatía y la interacción social.
- Desarrollo de Software y Programación en Python: Diseño, programación y optimización de código en Python para implementar modelos de reconocimiento emocional en una aplicación funcional.
- Educación y Terapias Especializadas para TEA: Enfoques educativos y terapéuticos en la intervención temprana de habilidades sociales y emocionales para niños con TEA, integrando tecnología como apoyo.

5.2. Marco Teórico

El Trastorno del Espectro Autista (TEA) representa una condición neuropsiquiátrica prevalente en la infancia, caracterizada por déficits en la comunicación social y la presencia de comportamientos repetitivos o estereotipados. Se postula que esta condición surge de una compleja interacción entre factores genéticos, epigenéticos y ambientales, dando lugar a una disfunción cerebral que impacta diversos aspectos del desarrollo infantil [1]. El reconocimiento temprano del TEA es fundamental para la implementación de intervenciones adecuadas que promuevan el desarrollo óptimo de la persona y mejoren su calidad de vida.

La creación de escuelas inclusivas ha surgido como una respuesta necesaria para abordar los desafíos que enfrentan los niños con TEA en entornos educativos convencionales. Se reconoce que las dificultades sociales asociadas con el TEA pueden interferir significativamente con el aprendizaje y la adaptación en el entorno escolar [8]. En este contexto, la inclusión educativa se ha convertido en una prioridad en muchos sistemas educativos, impulsando el desarrollo de programas y políticas destinados a garantizar que todos los estudiantes, incluidos aquellos con TEA, reciban el apoyo necesario para prosperar en un entorno educativo inclusivo [9].

Por otro lado, las estrategias pedagógicas específicamente diseñadas para niños con TEA han evolucionado para abordar sus necesidades únicas en el aula. Se reconoce la importancia de adaptar el currículo y las técnicas de enseñanza para maximizar el aprendizaje y la participación de estos niños [10]. Desde la implementación de rutinas estructuradas hasta el uso de recursos visuales y tecnológicos, se ha buscado facilitar la comprensión y la comunicación de los niños con TEA en el entorno educativo [11]. Estas estrategias pedagógicas, cada vez más centradas en las necesidades individuales de los estudiantes con TEA, reflejan un enfoque inclusivo y personalizado para promover su desarrollo académico y social [12].

Sin embargo, los desafíos asociados con la socialización temprana pueden tener consecuencias significativas en el bienestar emocional y el desarrollo de los niños con TEA. El acoso escolar, en particular, representa una preocupación importante, ya que estos niños enfrentan un mayor riesgo de victimización y exclusión social [13]. La experiencia de rechazo y marginalización puede contribuir al desarrollo de problemas de salud mental, como la ansiedad social y la depresión, exacerbando aún más sus dificultades emocionales y sociales. Por lo tanto, es crucial abordar estos desafíos de manera integral, mediante intervenciones terapéuticas y de apoyo en la comunidad que promuevan la inclusión y la aceptación de los niños con TEA [14].

En este contexto, se han desarrollado diversas terapias destinadas a mejorar la socialización y el bienestar de los niños con TEA. Desde la terapia cognitivo-conductual hasta las intervenciones basadas en el juego y la terapia equina, estas intervenciones se centran en proporcionar un entorno seguro y estructurado donde los niños con TEA puedan desarrollar habilidades sociales y emocionales de manera gradual y con el apoyo adecuado [15]. La investigación ha demostrado que estas

terapias pueden ser efectivas para reducir la ansiedad social, mejorar la autoestima y promover interacciones sociales positivas en niños con TEA. Además, los avances tecnológicos, como los robots sociales, han surgido como herramientas prometedoras para apoyar el desarrollo social y emocional de estos niños, proporcionando plataformas seguras para practicar habilidades sociales y recibir retroalimentación personalizada [16].

En resumen, los antecedentes proporcionados ofrecen una base sólida para comprender la complejidad del TEA y las diversas estrategias utilizadas para abordar sus desafíos. Al integrar estos elementos en el marco teórico de la investigación, se establece un contexto significativo para comprender el TEA y sus implicaciones para la práctica educativa y clínica.

Tema 1: Situación Actual de los Niños con TEA en Ambientes de Alta Socialización

A. Creación de Escuelas Inclusivas para Niños con TEA

El Trastorno del Espectro Autista (TEA) se caracteriza por déficits en la comunicación social, así como por patrones de comportamiento restrictivos y repetitivos (DSM-5). Estos desafíos afectan la capacidad de los individuos con TEA para participar en interacciones sociales significativas y pueden manifestarse en una variedad de formas, incluida la falta de interés en jugar con pares y dificultades para hacer amigos [17]. Además, los niños con TEA a menudo enfrentan dificultades en la comunicación verbal y no verbal, lo que puede dificultar la comprensión de señales sociales y la iniciación y mantenimiento de conversaciones [18].

Las consecuencias de estas dificultades pueden incluir aislamiento social, acoso escolar y problemas de salud mental [18]. A pesar de los esfuerzos por crear escuelas inclusivas para niños con TEA, el entorno escolar sigue siendo desafiante debido a factores como la sensibilidad sensorial, la comprensión verbal y de lectura, el funcionamiento ejecutivo y las habilidades motoras [18].

B. Estrategias Pedagógicas para Niños con TEA

El uso de computadoras ha surgido como una herramienta efectiva para ayudar a los individuos con TEA en áreas como la comunicación, la socialización, el comportamiento y lo académico. El uso adecuado de las computadoras puede fomentar la participación social y aumentar las interacciones positivas entre los estudiantes con TEA [19]. Sin embargo, es crucial individualizar la programación para satisfacer las necesidades específicas de los estudiantes y proporcionar intervenciones docentes que fomenten la interacción social [19]. A pesar de los beneficios potenciales, existe el riesgo de que el uso de computadoras pueda conducir a comportamientos compulsivos y obsesivos si no se supervisa adecuadamente [19]. Por lo tanto, se requieren estrategias pedagógicas específicas para reducir estos comportamientos y maximizar los beneficios del uso de computadoras como herramienta educativa para niños con TEA.

Tema 2: Efectos Negativos Producidos por los Problemas de Socialización a Temprana Edad

A. Bullying Actual hacia los Niños con TEA

Pocas investigaciones han investigado las experiencias de acoso escolar entre niños diagnosticados con TEA. La investigación preliminar sugiere que los niños con TEA corren un mayor riesgo de ser acosados que sus compañeros con desarrollo típico. Un estudio indicó que las experiencias de acoso escolar eran más de 4 veces más probables entre 411 niños con TEA que para una muestra nacional de jóvenes en Estados Unidos (55 % vs. 13 %, respectivamente) [20]. Un estudio posterior de 34 padres de niños de 5 a 21 años con TEA indicó que aproximadamente el 65 % de los niños habían sido víctimas de acosos por parte de sus compañeros [21], encontrando tasas similares entre 57 jóvenes diagnosticados con TEA de alto funcionamiento. Estos investigadores también encontraron que los jóvenes con TEA participaban en menos interacciones sociales en la escuela y reportaban tener menos amigos que sus compañeros [21].

Los hallazgos de este estudio indican que los niños con TEA que muestran ciertos comportamientos específicos, como no estar óptimamente sintonizados con la situación social y mostrar resistencia a los cambios, son especialmente vulnerables a la victimización por intimidación. Este resultado resalta la importancia de comprender cómo los síntomas del TEA pueden influir en las experiencias de victimización por intimidación y sugiere que los programas de intervención deben dirigirse específicamente a estas áreas de dificultad social y adaptativa.

Por lo tanto, en el marco teórico, podemos destacar la investigación existente que ha identificado los síntomas del TEA, como la falta de sintonización social y la resistencia al cambio, como factores de riesgo significativos para la victimización por intimidación entre los niños con TEA [21].

B. Ansiedad Social en los Niños con TEA

Los individuos con TEA a menudo sufren de ansiedad social severa porque luchan por comprender las señales sociales, mantener contacto visual, interpretar señales no verbales como expresiones faciales o lenguaje corporal, o participar en conversaciones recíprocas. Otros factores cognitivos incluyen una preferencia hacia situaciones predecibles, intolerancia a la incertidumbre y una tendencia hacia patrones de pensamiento rígidos. La imprevisibilidad en entornos sociales a menudo aumenta los niveles de ansiedad en los individuos con TEA, lo que los lleva a evitar tales situaciones. Otros factores de riesgo incluyen trastornos en el reconocimiento emocional y una competencia social reducida. Estos hallazgos sirven como guía para el desarrollo de mejores estrategias de intervención para ayudar a los individuos con TEA a superar la ansiedad social [22].

C. Depresión en Niños con TEA

Los síntomas depresivos en niños con trastorno del espectro autista (TEA) pueden presentarse de diversas formas según la edad y el nivel de desarrollo [23]. Mientras que algunos niños más pequeños pueden exhibir quejas somáticas y problemas de comportamiento, como un mayor compromiso con actividades o juegos con temas relacionados con la muerte [24], los adolescentes tienden a mostrar hipersomnia y un mayor riesgo de suicidio. Las manifestaciones específicas de la depresión en niños con TEA a menudo se ven exacerbadas por las dificultades en la comunicación, lo que dificulta la expresión de sentimientos de tristeza, desesperanza o desinterés [23].

La identificación de la depresión en niños con TEA suele basarse en la observación de comportamientos externos por parte de los cuidadores, ya que la capacidad de autorreporte puede estar limitada. Estos comportamientos pueden incluir un aumento de la tristeza, la tendencia al llanto, la apatía o cambios en el autocuidado. Además, los niños con TEA pueden mostrar signos vegetativos de depresión, como trastornos del sueño y del apetito, así como comportamientos autolesivos, como golpearse la cabeza [25].

Las dificultades asociadas con el déficit en la "teoría de la mente"[26] en niños con TEA pueden complicar aún más la identificación de la depresión. La falta de reflexión interna innata puede llevar a una mayor dependencia de signos alternativos, como un aumento en los comportamientos repetitivos o autodestructivos, como manifestaciones de una visión negativa de sí mismos [27].

Tema 3: Emociones Básicas y su Relevancia en el Reconocimiento Emocional en Niños con TEA

El estudio y reconocimiento de las emociones básicas ha sido un eje central en la psicología evolutiva, cognitiva y afectiva, siendo fundamental para comprender cómo las personas, incluyendo los niños con Trastorno del Espectro Autista (TEA), interpretan y responden a su entorno social. Las emociones básicas como la alegría, tristeza, miedo, sorpresa, disgusto, ira y la neutralidad (a menudo considerada como ausencia o equilibrio emocional) son denominadas “básicas” por su universalidad, expresión facial distintiva, aparición temprana en la vida y mecanismos biológicos subyacentes comunes a todas las culturas y contextos humanos.

A.Fundamentos teóricos de las emociones básicas

El psicólogo Paul Ekman, pionero en el estudio de las emociones humanas y su relación con las expresiones faciales, destacado por estar entre los 100 psicólogos más destacados del siglo XX [28], fue uno de los principales investigadores en establecer que ciertas emociones son universalmente reconocidas y expresadas mediante microexpresiones faciales. En sus investigaciones interculturales, Ekman identificó seis emociones universales: alegría (happiness), tristeza (sadness), miedo (fear), enojo (anger), sorpresa (surprise) y asco (disgust) [29][30][31].

Teorías modernas como la de la construcción psicológica de las emociones, sugieren que aunque las emociones pueden tener una base biológica, también están moldeadas por la experiencia y el contexto cultural [32]. Sin embargo, incluso desde esta perspectiva, las categorías emocionales como las mencionadas anteriormente son operativas y útiles para el entrenamiento de modelos computacionales, dado que reflejan patrones reconocibles en la conducta humana.

B. Importancia de las emociones en el desarrollo infantil y en el TEA

Durante el desarrollo típico, los niños aprenden a reconocer, expresar e interpretar estas emociones básicas desde la infancia temprana, lo que constituye una base esencial para el desarrollo de la empatía, la comunicación y la regulación emocional. Sin embargo, en niños con TEA, el reconocimiento e interpretación de estas emociones se ve alterado debido a déficits en la teoría de la mente y en la percepción de las señales sociales [33]. Por esta razón, el enfoque terapéutico y educativo en TEA ha dado prioridad al reconocimiento y entrenamiento de estas emociones, considerando que representan un núcleo funcional mínimo para fomentar habilidades sociales.

Estudios han demostrado que los niños con TEA presentan un rendimiento significativamente menor al intentar identificar estas emociones en expresiones faciales, en comparación con sus pares neurotípicos [34]. Por tanto, la enseñanza explícita de estas emociones mediante sistemas tecnológicos adaptados, como interfaces digitales o juegos serios, se ha convertido en una herramienta prometedora de intervención [35].

5.3. Trabajos Relacionados

Tema 4: Terapias existentes para niños con TEA
A. Terapias de juego

Aunque el mecanismo exacto del autismo sigue siendo controvertido, se han documentado defectos neurobiológicos atípicos en áreas como el cerebelo, las estructuras límbicas y el cortex, especialmente el lóbulo frontal. La disfunción del lóbulo frontal afecta a funciones cognitivas específicas, como la memoria, el aprendizaje, el reconocimiento facial y la interpretación de señales biológicas [36].

Los niños con TEA experimentan dificultades en la comprensión y el procesamiento sensorial, lo que limita su comportamiento adaptativo y su capacidad para relacionarse e interactuar socialmente con otros [37]. Esto se manifiesta en la preferencia por jugar solos, participar en juegos repetitivos o mostrar rabietas cuando se interrumpe su juego. Además, tienen dificultades para compartir experiencias, entender los sentimientos de los demás y prestar atención a su entorno. La interpretación de las señales no verbales del lenguaje corporal también representa un desafío para ellos [38].

A medida que los niños con TEA crecen, algunos pueden mejorar sus habilidades de juego y comprender algunas reglas de los juegos, pero su comportamiento durante el juego sigue siendo diferente al de sus pares típicamente desarrollados. A menudo, siguen estrictamente las reglas del juego y pueden obsesionarse con ciertos aspectos del juego, lo que puede desarrollarse en un trastorno obsesivo-compulsivo a largo plazo si no se trata adecuadamente [39].

La terapia ocupacional, especialmente la terapia de integración sensorial, es una estrategia ampliamente utilizada para apoyar el desarrollo de los niños con TEA y mejorar su juego. Esta terapia se basa en la recepción de estímulos sensoriales del entorno y su procesamiento en el cerebro para generar una respuesta adaptativa. Al combinar la terapia de juego con la integración sensorial, se busca involucrar a los niños y desarrollar su bienestar emocional, su desarrollo funcional y su crecimiento típico [40].

A pesar de las dificultades en el juego de los niños con TEA, todavía pueden aprender a jugar, especialmente con la ayuda de sus padres y hermanos. El papel de los padres es crucial en la gestión de las dificultades del niño, y la terapia temprana puede tener un impacto significativo en su desarrollo [41]. Además, la terapia ocupacional utilizando la teoría de integración sensorial se basa en evidencia y emplea un método lúdico y amigable para el niño, con el objetivo de mejorar las conexiones neuronales en el cerebro de los niños con TEA [40].

Tema 5: Avances tecnológicos actuales en pro de la problemática
A. Avances en realidad virtual para ayudar a los niños con TEA

El reconocimiento y comprensión de expresiones emocionales a través de múltiples señales es uno de los déficits sociales fundamentales en la población con TEA [42]. Los niños con TEA a menudo muestran un desarrollo emocional atípico en comparación con los niños con un desarrollo típico, manifestado como una falta de empatía hacia otras personas y una incapacidad para reaccionar emocionalmente ante los estados mentales de los demás [43]. La utilización de entrenamientos en Realidad Virtual (RV) ha demostrado ser particularmente útil para mejorar el reconocimiento de emociones en la población con TEA. Estudios previos han reportado un mejor rendimiento conductual, así como predictores neurales de cambio, en el reconocimiento emocional y la teoría de la mente en participantes con TEA después de completar programas de entrenamiento en RV [7].

La inmersión en el entorno virtual podría influir en el efecto de la intervención en el reconocimiento emocional. Se encontró que los participantes mostraban comportamientos emocionales más apropiados en un entorno de RV inmersivo en comparación con aplicaciones de RV de escritorio. Además, se observó que la alta inmersión de las tecnologías de RV con auriculares (HMD) fomenta una mayor extroversión en las actividades de aprendizaje y en la interacción de los niños con TEA [43].

Recientes estudios han intentado integrar la tecnología de RV con señales psicofisiológicas dinámicas para mejorar los enfoques de intervención en el reconocimiento emocional. Se desarrollaron sistemas de seguimiento ocular dinámico y sistemas que registran señales electrofisiológicas para proporcionar retroalimentación individualizada y objetiva durante el entrenamiento en el reconocimiento emocional [44]. Estos enfoques han demostrado mejoras significativas en la capacidad de reconocimiento emocional de los niños con TEA.

Además del entrenamiento en reconocimiento emocional, los investigadores han utilizado la RV para comprender cómo los individuos con TEA perciben y manejan expresiones emocionales. Por ejemplo, se utilizó una prueba llamada V-REST [45] para examinar la percepción emocional y la distancia interpersonal en TEA con la ayuda de un joystick. Este estudio reveló diferencias significativas en cómo los niños con TEA se acercan a expresiones positivas en comparación con los niños con un desarrollo típico, lo que sugiere diferencias en la motivación social o sensibilidad a eventos socioemocionales positivos. Estos hallazgos plantean la necesidad de revisar y actualizar el modelo de motivación social en TEA [46].

B. Tecnología asistida para el reconocimiento de emociones en niños con TEA

El avance de las tecnologías asistidas ha permitido mejorar significativamente el reconocimiento de estados afectivos en niños con Trastorno del Espectro Autista (TEA). Estas tecnologías abordan el déficit de los niños con TEA en la expresión y comprensión de emociones, facilitando la intervención tanto en entornos terapéuticos como educativos. En particular, la tecnología de reconocimiento de emociones asistida por sistemas automatizados permite evaluar y clasificar los estados afectivos de manera más eficiente, lo que puede ser de gran ayuda en las sesiones de terapia [47].

Una revisión sistemática reciente identificó varias metodologías que se han implementado para medir los estados afectivos de niños con TEA, a través del uso de imágenes térmicas, análisis de expresiones faciales y señales acústicas. Estas tecnologías han demostrado ser eficaces, destacando en el uso de estímulos visuales y de video para inducir estados emocionales específicos, como alegría, tristeza, miedo y sorpresa. Este enfoque puede superar las dificultades de los niños con TEA para expresar verbalmente sus emociones, facilitando la identificación precisa de sus estados afectivos por parte de los terapeutas [47].

Además, se han empleado diversos clasificadores, como las máquinas de soporte vectorial (SVM) y redes bayesianas, que han mostrado altos niveles de precisión en la diferenciación de emociones. Este uso de clasificadores automatizados reduce la dependencia de los juicios subjetivos de los terapeutas y permite una evaluación más objetiva de las emociones de los niños con TEA. Los resultados de estas investigaciones sugieren que los sistemas basados en tecnología asistida no solo mejoran el reconocimiento emocional, sino que también ayudan a establecer una interacción más efectiva entre los niños con TEA y sus cuidadores o terapeutas [47].

C. Juegos serios y evaluación para el aprendizaje emocional en niños con TEA

El uso de juegos serios ha demostrado ser una herramienta efectiva para enseñar a los niños con Trastorno del Espectro Autista (TEA) habilidades emocionales clave. En particular, estos juegos permiten a los niños practicar el reconocimiento y la expresión de emociones en un entorno lúdico e interactivo. Uno de los componentes más importantes de este enfoque es la capacidad de los niños no solo para aprender a expresar emociones, sino también para reconocer las emociones en los demás, lo que es fundamental para el desarrollo de habilidades sociales.

En el juego serio EmoTEA, los niños interactúan con la aplicación mediante interfaces tangibles y reconocimiento facial. La tecnología de reconocimiento facial se utiliza para monitorear y evaluar las expresiones emocionales de los niños mientras intentan imitar diversas emociones que se les muestran en la pantalla. Sin embargo, uno de los elementos más destacados del juego es la fase en la que los niños deben reconocer emociones en otras personas. Este reconocimiento se lleva a cabo mediante la observación de videos en los que se muestran expresiones emocionales en contextos específicos. En esta etapa, el objetivo es que el niño identifique correctamente las emociones que se presentan en las caras de los personajes en pantalla, utilizando tarjetas de emociones como interfaz para responder [48].

El reconocimiento de emociones en personas externas es un aspecto crucial en el desarrollo emocional de los niños con TEA, quienes generalmente enfrentan dificultades para interpretar correctamente los estados emocionales de los demás. Esta habilidad es esencial para mejorar la reciprocidad social y la interacción en situaciones cotidianas. Según el artículo, la capacidad de interpretar las emociones de otros permite a los niños con TEA comprender mejor las señales sociales, lo que a su vez mejora su participación en interacciones sociales, como juegos grupales, conversaciones o la simple interpretación de los estados emocionales de sus compañeros y familiares [48].

La importancia de reconocer emociones externas radica en que esta habilidad ayuda a los niños a entender cómo sus propias acciones o palabras pueden afectar a los demás, fomentando una mayor empatía y mejorando su capacidad para adaptarse socialmente. Esta habilidad es una de las mayores dificultades para los niños con TEA, pero mediante el uso de tecnologías como el reconocimiento facial y el aprendizaje basado en juegos, pueden adquirirla de manera más efectiva.

D. Apoyo académico y avances en robótica para el desarrollo de competencias socioemocionales en niños con TEA

El proyecto ERIK (Entwicklung einer Roboterplattform zur Unterstützung neuer Interaktionsstrategien in der Therapie von Kindern mit eingeschränkten sozio-emotionalen Fähigkeiten), respaldado por el Ministerio Federal de Educación e Investigación de Alemania (BMBF), tiene como objetivo desarrollar una plataforma robótica que apoye la terapia de niños con Trastorno del Espectro Autista (TEA) en la mejora de sus habilidades socioemocionales. Este proyecto ha sido llevado a

cabo por un consorcio de universidades como la Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) y centros de investigación destacados en el ámbito de la tecnología y la robótica terapéutica como el Fraunhofer Institute for Integrated Circuits IIS y el audEERING, quienes han desarrollado algoritmos avanzados de reconocimiento de emociones y pulsaciones para ser integrados en esta plataforma. [4].

El estudio destaca que la robótica ofrece una oportunidad única para abordar las limitaciones de las terapias tradicionales. Los robots sociales, como los utilizados en ERIK, proporcionan un entorno estructurado y predecible, lo que es crucial para los niños con TEA, quienes suelen tener dificultades en entornos sociales impredecibles. Los robots ayudan a que los niños practiquen habilidades socioemocionales de forma controlada y repetitiva, lo que aumenta la retención de los aprendizajes [4].

Una de las grandes innovaciones del proyecto ERIK es la implementación de tecnologías de reconocimiento facial y de medición del pulso para capturar y analizar en tiempo real las respuestas emocionales de los niños durante las sesiones de terapia. Esto permite a los terapeutas ajustar el enfoque de la terapia de manera más precisa, basándose en datos objetivos sobre el estado emocional de los niños. Además, los juegos serios integrados en la plataforma no solo promueven la identificación de las emociones propias, sino que también enseñan a los niños a reconocer las emociones en otras personas a través de la observación de expresiones faciales y contextuales, lo cual es fundamental para mejorar su interacción social [4].

El estudio también enfatiza la importancia de la colaboración interdisciplinaria en el diseño y la implementación de esta tecnología, reuniendo a expertos en psicología, terapia ocupacional y robótica para crear un sistema que sea a la vez efectivo y seguro para los niños con TEA. Los primeros resultados de los ensayos clínicos realizados en Alemania han mostrado una mejora significativa en las capacidades de regulación emocional y empatía de los niños, lo que subraya el potencial de los robots asistidos por inteligencia artificial en intervenciones terapéuticas para poblaciones con necesidades especiales [4].

Tema 6: Deep Learning y Arquitecturas

El deep learning es una subrama del machine learning que se basa en el uso de redes neuronales artificiales de múltiples capas, conocidas como redes neuronales profundas. Estas redes están diseñadas para imitar la estructura y el funcionamiento del cerebro humano, permitiendo a los modelos aprender representaciones de datos de manera jerárquica y automática. A diferencia de los algoritmos tradicionales de machine learning, que requieren la ingeniería manual de características, el deep learning permite el aprendizaje de características complejas directamente a partir de datos brutos.

Una red neuronal profunda consta de varias capas de neuronas, cada una de las cuales realiza operaciones matemáticas sobre los datos de entrada para extraer características relevantes. Las capas iniciales suelen aprender características de bajo nivel, como bordes y texturas en el caso de las

imágenes, mientras que las capas más profundas capturan características de alto nivel, como objetos y escenas completas. Este proceso se logra mediante la combinación de operaciones lineales (multiplicación de matrices y vectores) y no lineales (funciones de activación como ReLU, Sigmoid y Tanh).

El entrenamiento de modelos de deep learning se lleva a cabo mediante un proceso llamado retropropagación, que ajusta los pesos de las conexiones neuronales para minimizar el error en las predicciones del modelo. Este proceso implica el uso de algoritmos de optimización, como el gradiente descendente, que ajustan iterativamente los pesos basándose en la derivada del error con respecto a cada peso. Debido a la gran cantidad de parámetros involucrados, el entrenamiento de redes neuronales profundas requiere un poder computacional significativo, a menudo utilizando unidades de procesamiento gráfico (GPUs) y técnicas de paralelización.[49].

A. Arquitectura Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales (CNN) son una clase de arquitecturas de deep learning diseñadas específicamente para el procesamiento y análisis de datos con estructura de cuadrícula, como imágenes y secuencias de video. A diferencia de las redes neuronales totalmente conectadas, las CNNs utilizan capas convolucionales que aplican filtros de manera local a las entradas, lo que permite la detección de características locales y la reducción de la complejidad computacional. Estos filtros son capaces de capturar patrones espaciales jerárquicos, desde bordes y texturas en las primeras capas hasta objetos completos en las capas más profundas[49].

Las CNNs han sido fundamentales en el avance de la visión por computadora, impulsando el desarrollo de sistemas de reconocimiento de imágenes, detección de objetos y segmentación semántica. La arquitectura típica de una CNN incluye capas convolucionales, capas de pooling (agrupación) y capas totalmente conectadas al final. Las capas convolucionales extraen características de bajo nivel, mientras que las capas de pooling reducen la dimensionalidad de las representaciones, y las capas totalmente conectadas combinan estas características para realizar la clasificación final[50].

Una capa convolucional funciona aplicando múltiples núcleos de convolución (kernels) que recorren la entrada mediante un desplazamiento (stride). El resultado se conoce como mapa de activación o *feature map*. Luego, se aplica una función de activación no lineal, como la ReLU (Rectified Linear Unit), que transforma cada valor negativo en cero, introduciendo no linealidad al modelo y acelerando el aprendizaje. Para reducir la dimensionalidad espacial y mitigar el sobreajuste, se emplean capas de agrupamiento como *max pooling*, que selecciona el valor máximo dentro de una región del mapa de activación.

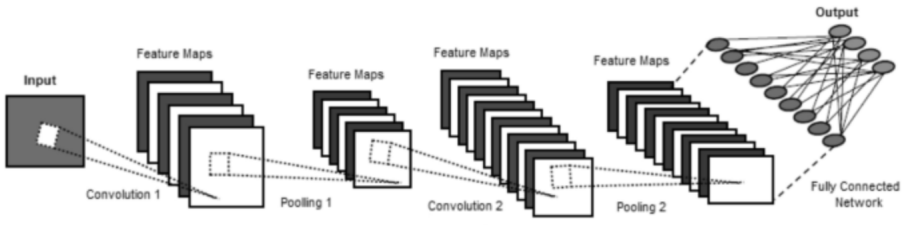


Figura 5.1: Red Neuronal Convolucional (CNN).

Figura 5.1 tomada de: [51]

Tras varias capas convolucionales y de pooling, la salida se aplana y se conecta a una red densa o Perceptrón Multicapa (MLP, por sus siglas en inglés). El MLP es una red neuronal totalmente conectada compuesta por una o más capas densas, donde cada neurona de una capa está conectada a todas las neuronas de la capa anterior. Su función es combinar las características extraídas por las convoluciones y aprender representaciones más abstractas que permitan realizar la predicción final. [52]

El proceso de entrenamiento incluye la retropropagación del error y la actualización de los pesos utilizando algoritmos de optimización como SGD o Adam, ajustando la red para minimizar la función de pérdida. Este enfoque ha demostrado ser eficaz en una variedad de aplicaciones como reconocimiento facial, análisis médico y conducción autónoma. Además, las CNN's modernas integran estrategias como *batch normalization, data augmentation y dropout*, que permiten entrenar redes más profundas con estabilidad y buena capacidad de generalización. [53]

B. Arquitectura VGG16

La VGG16 es una arquitectura de red neuronal convolucional profunda desarrollada por Karen Simonyan y Andrew Zisserman en 2014, en el Visual Geometry Group de la Universidad de Oxford. Esta arquitectura fue presentada en el artículo "Very Deep Convolutional Networks for Large-Scale Image Recognition"[50]. VGG16 se caracteriza por su simplicidad estructural, utilizando principalmente convoluciones de tamaño 3x3 y capas de pooling de tamaño 2x2, dispuestas de manera uniforme a lo largo de la red.Esta simplicidad estructural permite construir redes muy profundas manteniendo bajo control la complejidad computacional. La arquitectura está compuesta por cinco bloques de convolución, seguidos por tres capas densas (MLP) que realizan la clasificación. Cada bloque aumenta progresivamente el número de filtros (64, 128, 256, 512, 512), mientras que la resolución espacial se reduce mediante pooling.

Figura 5.2: Arquitectura VGG16.

Figura 5.2 tomada de: [54]

Con 16 capas con pesos entrenables, VGG16 ha demostrado un rendimiento sobresaliente en la clasificación de imágenes, alcanzando una precisión del 92.7 % en el desafío ImageNet. A pesar de su efectividad, uno de los principales inconvenientes de VGG16 es su alta demanda de recursos computacionales y memoria debido al gran número de parámetros que maneja. Esta arquitectura ha sido ampliamente adoptada y utilizada como base para desarrollos posteriores en visión por computadora y otras aplicaciones de deep learning[49][50].

VGG16 procesa una imagen de entrada de 224 x 224 x 3 y transforma su representación a través de 13 capas convolucionales. La salida final de la última capa convolucional tiene una dimensión de 7 x 7 x 512, que se aplana a un vector de 25088 características. Este vector pasa por una red densa que constituye el Perceptrón Multicapa (MLP) del modelo: dos capas con 4096 neuronas y una capa final con el número de clases requeridas, con activación softmax. En este contexto, el MLP funciona como un clasificador de alto nivel que traduce los patrones espaciales en probabilidades asociadas a cada clase[55].

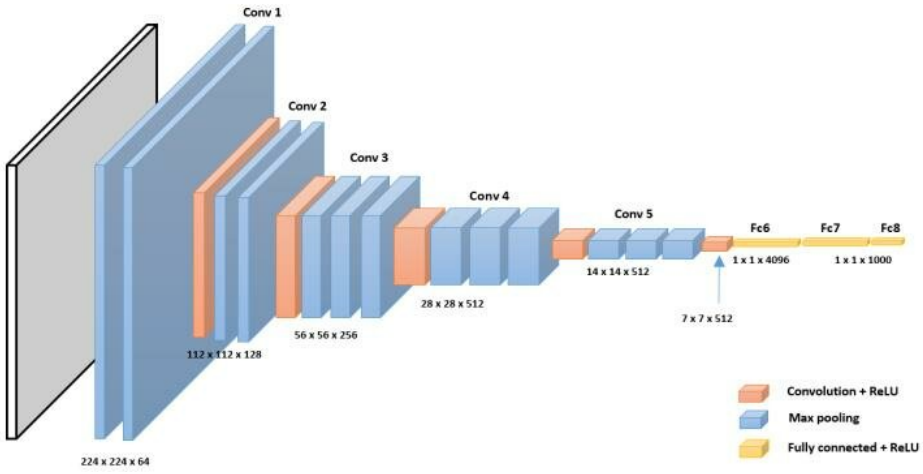


Figura 5.3: Estructura de Red VGG16.

Figura 5.3 tomada de: [56]

En contextos de transferencia de aprendizaje *(transfer learning)*, el cuerpo convolucional de VGG16 se congela, lo que permite adaptar la arquitectura a nuevas tareas con menor esfuerzo computacional. Esta característica ha hecho de VGG16 una elección popular para aplicaciones médicas, detección de emociones, clasificación de escenas y más. La arquitectura también es altamente interpretable y sirve como base para variantes ligeras que mantienen su rendimiento con menor cantidad de parámetros [57].

Se han evaluado mejoras sobre VGG16, como reducir las dimensiones de las capas densas o reemplazar funciones de activación, logrando una mayor eficiencia sin perder precisión. También se han explorado adaptaciones que integran módulos de atención o normalización adaptativa para refinar aún más el desempeño del modelo en tareas complejas[58].

C. Arquitectura ResNet50

La ResNet50, representa un avance significativo en el diseño de redes neuronales profundas[59]. La principal innovación de ResNet50 es el uso de bloques residuales, que permiten la formación de redes extremadamente profundas al facilitar la propagación de gradientes a través de la red. Esto se logra mediante la introducción de conexiones de salto *(skip connections)* que eluden una o más capas, permitiendo que el gradiente fluya directamente a través de la red durante el proceso de entrenamiento.

ResNet50 consta de 50 capas, incluyendo convoluciones, normalización por lotes y activaciones ReLU, estructuradas en bloques residuales que mitigan el problema del desvanecimiento de gradientes. Esta arquitectura ha demostrado una gran efectividad en tareas de clasificación y detección de imágenes, y ha sido adoptada ampliamente en la investigación y la industria debido a su capacidad para entrenar redes profundas sin pérdida significativa de rendimiento. La introducción de conexiones residuales ha permitido desarrollar modelos aún más profundos y complejos, como ResNet101 y ResNet152, que continúan estableciendo nuevos estándares en el campo del deep learning[59].

Desde el punto de vista estructural, las 50 capas con pesos entrenables, organizadas en una arquitectura modular basada en bloques residuales. El modelo inicia con una convolución de 7×7 y una capa de *max pooling*, seguidas por cuatro bloques principales (Conv2_x, Conv3_x, Conv4_x y Conv5_x), cada uno compuesto por varios sub-bloques residuales. Cada sub-bloque sigue un patrón "bottleneck" que incluye tres capas convolucionales: una convolución 1×1 para reducir la dimensionalidad, una convolución 3×3 para el procesamiento principal y otra convolución 1×1 para restaurar la dimensionalidad original. Estas capas están intercaladas con normalización por lotes *(batch normalization)* y activaciones ReLU, lo que estabiliza y acelera el aprendizaje en redes profundas[59].

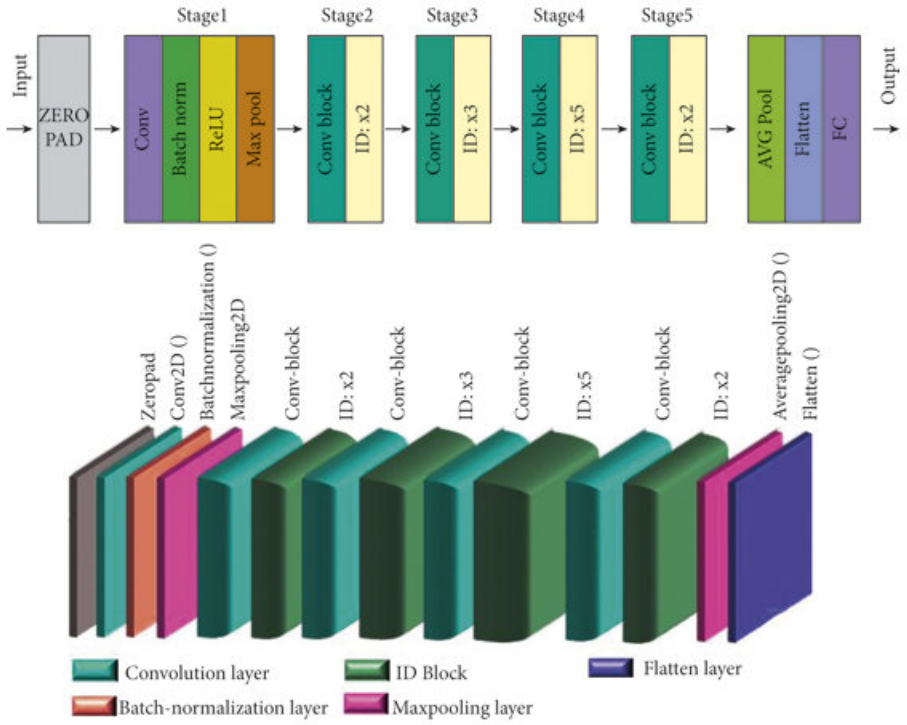


Figura 5.4: Arquitectura ResNet50

Figura 5.4 tomada de: [60]

El entrenamiento de ResNet50 se basa en técnicas avanzadas de optimización como el descenso del gradiente estocástico (SGD) con momentum o algoritmos adaptativos como Adam [59]. Para mejorar la generalización del modelo y reducir el sobreajuste, se emplean técnicas como *data augmentation, label smoothing y learning rate scheduling* [61]. Además, en contextos de *transfer learning*, el modelo preentrenado en ImageNet se adapta fácilmente a nuevas tareas mediante la congelación de las capas convolucionales y la sustitución de la capa final por una específica a la tarea objetivo. Esta flexibilidad ha hecho que ResNet50 se convierta en un estándar industrial y académico para tareas de clasificación, detección de objetos y análisis biomédico [62].

Adicionalmente, investigaciones recientes han confirmado que la arquitectura residual no solo facilita el entrenamiento de redes muy profundas, sino que también las lleva a aprender representaciones más robustas. Esta robustez ha sido clave para su integración como componente fundamental en modelos más complejos, como redes generativas o incluso en el diseño de nuevas arquitecturas como los transformadores visuales [63].

Tema 7: Métricas y Fórmulas

En el campo del aprendizaje automático y, específicamente, en la evaluación de modelos de clasificación como los utilizados para el reconocimiento de emociones, es fundamental emplear métricas estandarizadas que permitan cuantificar su rendimiento de manera objetiva. Estas métricas no solo ofrecen una medida de la efectividad del modelo, sino que también guían el proceso de optimización y ajuste.

A. Matriz de Confusión:

La matriz de confusión es una herramienta fundamental para visualizar el rendimiento de un algoritmo de clasificación. Organiza las predicciones del modelo en una tabla que compara las clases predichas con las clases reales, permitiendo un análisis detallado de los aciertos y errores. [64] [65] La matriz se compone de cuatro valores clave:

- **VP (Verdaderos Positivos):** Casos en los que el modelo clasifica correctamente un ejemplo que pertenece a una determinada clase.
- **VN (Verdaderos Negativos):** Casos en los que el modelo no clasifica algo como perteneciente a una clase determinada, y en efecto, no pertenece a esa clase.
- **FP (Falsos Positivos):** Casos en los que el modelo predice que algo pertenece a una clase, pero en realidad pertenece a otra.
- **FN (Falsos Negativos):** Casos en los que algo pertenece realmente a una clase, pero el modelo lo clasifica como si fuera otra.

En un escenario multiclase, como el reconocimiento de varias emociones, la matriz se expande a una dimensión de N x N, donde N es el número de emociones, mostrando cómo se distribuyen las predicciones para cada clase real. Para todos los escenarios, la diagonal de la matriz de confusión representan las predicccciones correctas. [64].

B. Exactitud (Accuracy):

La exactitud (Accuracy) mide la proporción de predicciones correctas sobre el total de predicciones realizadas. Es la métrica más intuitiva para evaluar un modelo, pero puede ser engañosa cuando las clases están desbalanceadas[65][66].

Su fórmula es:

$$\text{Accuracy} = \frac{VP + VN}{VP + VN + FP + FN}$$

Para escenarios multiclase su fórmula es:

$$\text{Accuracy} = \frac{\text{Número de predicciones correctas}}{\text{Número total de predicciones}}$$

Matemáticamente su fórmula con N muestras se expresa como:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = y_i)$$

$$\text{donde } \mathbf{1}(\hat{y}_i = y_i) = \begin{cases} 1, & \text{si } \hat{y}_i = y_i \\ 0, & \text{si } \hat{y}_i \neq y_i \end{cases}$$

Donde:

- N: número total de muestras (ejemplos clasificados).
- $\hat{y}_i$: etiqueta predicha para la muestra i.
- y_i: etiqueta real (verdadera) para la muestra i.

C. Precisión:

La precisión evalúa la calidad de las predicciones positivas[64]. Responde a la pregunta: de todas las predicciones positivas que hizo el modelo, ¿cuántas fueron realmente correctas? Es una métrica crucial cuando el costo de los falsos positivos es alto[65][66].

Su fórmula es:

$$\text{Precisión}_k = \frac{VP_k}{VP_k + FP_k}$$

Donde:

- VP_k: Verdaderos Positivos para la clase k (casos correctamente clasificados como clase k).
- FP_k: Falsos Positivos para la clase k (casos de otra clase que se clasificaron incorrectamente como k).

D. Recall (Sensibilidad):

El recall, también conocido como sensibilidad o tasa de verdaderos positivos[64], mide la capacidad del modelo para detectar correctamente los ejemplos que realmente pertenecen a una clase determinada. Responde a la pregunta: de todos los casos positivos reales, ¿cuántos logró identificar el modelo? Es fundamental cuando el costo de los falsos negativos es significativo[65].

Su fórmula es:

$$\text{Recall}_k = \frac{VP_k}{VP_k + FN_k}$$

Donde:

- VP_k: Verdaderos Positivos para la clase k (casos correctamente clasificados como clase k).
- FN_k: Falsos Negativos para la clase k (casos reales de la clase k que fueron clasificados como otra clase).

E. F1-Score:

El F1-Score es la media armónica de la precisión y el recall. Ofrece un equilibrio entre ambas métricas, siendo particularmente útil cuando existe un desbalance de clases. Un F1-Score alto indica que el modelo tiene un buen rendimiento tanto en precisión como en recall[65][64][66].

Su fórmula es:

$$\text{F1-Score}_k = 2 \cdot \frac{\text{Precisión}_k \cdot \text{Recall}_k}{\text{Precisión}_k + \text{Recall}_k}$$

Donde:

- Precisión_k: Porcentaje de predicciones correctas para la clase k entre todas las predicciones que el modelo hizo para esa clase.
- Recall_k: Porcentaje de ejemplos de la clase k que fueron correctamente reconocidos por el modelo.

F. Loss:

La función de pérdida (Loss) cuantifica el error del modelo durante el entrenamiento. El objetivo del entrenamiento es minimizar esta función. Un valor de pérdida bajo indica que las predicciones del modelo se acercan a los valores reales.

G. Curvas de aprendizaje:

Las curvas de aprendizaje (Learning Curves) son gráficos que muestran la evolución del rendimiento del modelo (generalmente usando la pérdida o la exactitud) a lo largo de las épocas de entrenamiento, tanto en el conjunto de datos de entrenamiento como en el de validación [67]. Son esenciales para diagnosticar problemas como:

- **Underfitting:** El modelo tiene un mal rendimiento tanto en los datos de entrenamiento como en los de validación, lo que indica que es demasiado simple para capturar la complejidad de los datos[67].
- **Overfitting:** El modelo funciona muy bien en los datos de entrenamiento pero mal en los de validación, lo que significa que ha memorizado el ruido de los datos de entrenamiento en lugar de generalizar[67].

H. Macro Avg y Weighted Avg:

Cuando se trabaja con clasificación multiclase, es importante promediar métricas como la precisión, el recall y el F1-Score para obtener una visión general del rendimiento.

- **Macro Avg (Promedio Macro)**: Calcula la métrica (Precisión, Recall o F1-Score) de forma independiente para cada clase (Cada emoción) y luego toma el promedio simple [68]. Trata a todas las clases por igual, sin importar su frecuencia.

$$\text{Métrica}_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \text{Métrica}_k$$

  Donde:

  - K: número total de clases (3 o 7 emociones).
  - Métrica_k: Precisión, Recall o F1-Score para la clase k.
  - Métrica_{macro}: Macro Avg para Precisión, Recall o F1-Score

- **Weighted Avg (Promedio Ponderado):** Calcula la métrica (Precisión, Recall o F1-Score) para cada clase (emoción) y luego obtiene un promedio ponderado según la cantidad de ejemplos reales de cada clase (el soporte) [68]. Este promedio es más representativo cuando hay un desbalance de clases, es decir, le da más peso a las clases con más ejemplos.

$$\text{Métrica}_{\text{weighted}} = \frac{1}{N} \sum_{k=1}^{K} n_k \cdot \text{Métrica}_k$$

Donde:

- K: número total de clases (por ejemplo, 3 o 7 emociones).
- N: número total de ejemplos en el conjunto de evaluación.
- n_k: cantidad de ejemplos reales pertenecientes a la clase k (Soporte de la clase).
- Métrica_k: Precisión, Recall o F1-Score para la clase k.
- $\text{Métrica}_{weighted}$: Weighted Avg para Precisión, Recall o F1-Score

Capítulo 6

Materiales y Métodos

6.1. Materiales

Para el desarrollo del proyecto se usaron múltiples materiales físicos, así como recursos computacionales.

6.1.1. Base de Datos

Se seleccionó la base de datos Real-world Affective Faces Database (RAF-DB). Esta es una base de datos a gran escala diseñada específicamente para el reconocimiento de expresiones faciales en entornos no controlados, comúnmente denominados -in the wild-, es decir, que son emociones auténticas.

La base de datos RAF-DB se distingue por la alta variabilidad de sus imágenes, las cuales fueron obtenidas de internet. Estas presentan una diversidad significativa en términos de edad, género y etnia de los sujetos, así como variaciones en poses de la cabeza, condiciones de iluminación y oclusiones (por ejemplo, uso de gafas, vello facial o auto-oclusión). Cada imagen fue etiquetada de forma independiente por 40 anotadores, lo que garantiza una alta fiabilidad en las etiquetas emocionales asignadas.

La base de datos fue cargada hace 2 años (2023) y se encuentra disponible en el siguiente enlance: https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset

6.1.2. Criterios de Selección de la Base de Datos

La elección de RAF-DB sobre otras bases de datos de expresiones faciales (como CK+ o JAFFE) se fundamentó en los siguientes criterios estratégicos, alineados con los objetivos y limitaciones del presente trabajo de grado:

- **Realismo y Generalización** A diferencia de bases de datos creadas en laboratorios con fondos neutros y expresiones posadas, RAF-DB contiene imágenes del "mundo real". Entrenar los modelos con datos tan diversos es crucial para que el sistema final pueda generalizar y funcionar correctamente en situaciones prácticas y no controladas, como las que enfrentaría un niño con TEA en su día a día.

- **Tamaño y Rigurosidad Académica** Para un proyecto de grado que involucra el entrenamiento de redes neuronales profundas (CNN, VGG16, ResNet50), es indispensable contar con un volumen de datos suficiente para evitar el sobreajuste (*overfitting*) y lograr un modelo robusto. Las bases de datos más pequeñas no serían adecuadas para este fin. La versión de RAF-DB utilizada, con un total de 12271 imágenes, ofrece la escala necesaria para cumplir con la rigurosidad académica requerida.
- **Accesibilidad y Viabilidad del Proyecto** Se optó por la versión de RAF-DB de acceso público, la cual no requiere de un proceso de solicitud formal por correo electrónico. Esta decisión fue clave para agilizar la fase inicial del proyecto y asegurar su desarrollo dentro de los plazos establecidos. Además, aunque es un conjunto de datos grande, su tamaño es manejable para el entrenamiento con recursos computacionales limitados, como los disponibles a través de servicios como Google Colab o Visual Studio Code.

(a) Surprise

(b) Fear

(c) Disgust

(d) Happy

(e) Sad

(f) Angry

(g) Neutral

Figura 6.1: Mosaico de imágenes por emoción correspondiente a la base de datos usada en el proyecto.

6.1.3. Distribución de Datos Utilizada

Para este estudio, se trabajó con el subconjunto de imágenes etiquetadas con las siete emociones básicas. La distribución de las imágenes por clase fue la siguiente:

- **Happy (Feliz):** 4772 imágenes
- **Neutral (Neutro):** 2524 imágenes
- **Sad (Triste):** 1982 imágenes
- **Surprise (Sorprendido):** 1290 imágenes
- **Disgust (Disgusto):** 717 imágenes
- **Anger (Enojado):** 705 imágenes
- **Fear (Miedo):** 281 imágenes

Todas las imágenes están en formato JPG, tienen un tamaño de 100 × 100 píxeles (entrada).

6.1.4. Hardware

Hardware MacBook Pro M4 con las siguientes especificaciones técnicas:

- CPU Apple M4 (10 núcleos: 4 de alto rendimiento, 6 de eficiencia)
- RAM: 16 GB unificada
- macOS 15.4.1

ASUS TUF Gaming FX705GM con las siguientes especificaciones técnicas:

- CPU Intel Core i7-8750H de octava generación
- GPU NVIDIA GeForce GTX 1060
- RAM: 16 GB DDR4 2666 MHz

6.1.5. Software

- Lenguaje de programación: Python 3.11

Frameworks y librerías:

- TensorFlow: Permite el aprendizaje automático y el aprendizaje profundo.

- keras: es una API de alto nivel integrada en TensorFlow que simplifica la construcción y entrenamiento de modelos de redes neuronales. Juntas, estas herramientas permiten desarrollar, entrenar y desplegar modelos de aprendizaje profundo de manera eficiente.

- Os: Esta librería proporciona funciones para interactuar con el sistema operativo. Permite realizar tareas como navegar por el sistema de archivos, crear o eliminar directorios, y manipular rutas de archivos.

- OpenCV: Proporciona herramientas para procesamiento de imágenes, es fundamental para la detección y reconocimiento de objetos.

- NumPy: Es fundamental para el cálculo numérico en Python, permitió ejecutar funciones para realizar operaciones matemáticas complejas de manera rápida y eficiente.

- Matplotlib: Permite generar gráficos de líneas, histogramas, diagramas de dispersión y más, siendo útil para visualizar datos y resultados de modelos.

- ImageDataGenerator: En el marco del proceso, permitio realizar Data Augmentation lo que facilita la preparación y aumento de datos de imágenes en tiempo real durante el entrenamiento de modelos.

- sklearn.model selection: Permite dividir conjuntos de datos en subconjuntos de entrenamiento y prueba. Es esencial para evaluar el rendimiento de los modelos de aprendizaje automático, asegurando que se prueben con datos no vistos durante el entrenamiento.

- sklearn.metrics: Estas funciones de scikit-learn se utilizan para evaluar el rendimiento de modelos de clasificación. confusion matrix calcula la matriz de confusión, que muestra las predicciones correctas e incorrectas del modelo. ConfusionMatrixDisplay permite visualizar esta matriz de manera gráfica, facilitando la interpretación de los resultados.

6.1.6. Recursos Computacionales

Como recurso fundamental en el proyecto, se usó el editor de código fuente ligero Visual Studio (VScode), programa disponible en sistemas operativos como Windows y MacOs.

Como parte del proceso, la aplicación Anaconda Navigator, fue imprescindible para facilitar los entornos virtuales y paquetes. Por medio de este programa se creó el Environment y se instalaron las librerias necesarias para que los códigos se ejecutaran exitosamente en VScode.

Inicialmente Google Colab en sus versiones gratuita y pro, implicó un gran avance para la creación de los códigos, no obstante, su dependencia a la red y a los servidores de Google implicaron un problema debido a los largos tiempos de ejecución y problemas con la RAM (límite excedido), se optó por usar VScode usando los recursos propios de los 2 dispositivos empleados.

6.2. Métodología

6.2.1. Método y plan de pruebas general

El plan de pruebas se diseñó en dos fases estratégicas para abordar el problema de forma incremental. La fase inicial se centró en un problema de clasificación de **tres clases** (Felicidad, Tristeza y Neutralidad), la cual sirvió como **prueba de concepto y para establecer una línea base de rendimiento**. La justificación para comenzar con este subconjunto simplificado es doble:

1. **Validación Metodológica:** Permitió verificar la correcta implementación y la capacidad de aprendizaje de las arquitecturas de red neuronal seleccionadas (`CNN` personalizada, `VGG16` y `ResNet50`) en un escenario controlado y con clases bien representadas en el dataset, antes de introducir la complejidad del problema completo.
2. **Establecimiento de una Línea Base** Proporcionó un punto de referencia claro y medible del rendimiento de los modelos. Esto fue fundamental para luego poder cuantificar de manera precisa el impacto en la dificultad y en las métricas de evaluación al escalar el problema a las siete clases, que presentan un mayor desbalance.

Una vez validada la metodología en la primera fase, se procedió a la segunda, que abordó el problema completo con las **7 clases** de emociones.

Para ambas fases, la división de los datos se estableció en una proporción de **85 % para el conjunto de entrenamiento y 15 % para el de validación**. Esta distribución se seleccionó por razones empiricas, ya que cuando se usaban distribuciones como 75-25, 80-20 o 90-10 los resultados en terminos de precisión caían abruptamente (menos del 40 %). Con esta decisión se pudo maximizar la cantidad de ejemplos disponibles para el entrenamiento, permitiendo que el modelo aprenda patrones más robustos, al tiempo que se conserva un subconjunto estadísticamente significativo para validar el rendimiento en datos no vistos y monitorear eficazmente el sobreajuste (*overfitting*).

Además, se aplicó aumento de datos (*data augmentation*) para mejorar la generalización del modelo, cada clase fue ajustada a la clase con mayor numero de imagenes, en este caso "happy", para ser mas especificos despues de la división de datos la clase quedó con 4056 imagenes (cabe resaltar que el conjunto de validación no fue sujeto al aumento de imagenes). En conclusión, cada clase recibio un aumento porcentual especifico:

- Happy: Aumento del 0 %
- Sad: Aumento del 104.64 %
- Neutral: Aumento del 60.7 %
- Anger: Aumento del 475.32 %
- Surprise: Aumento del 214.42 %

- Disgust: Aumento del 465.69 %
- Fear: Aumento del 1343.42 %

En dicho proceso de aumento de imagenes fue crucial someter las imagenes a ciertas operaciones, a continuación se especifican y se justifica su elección tomando en cuenta la naturaleza del problema tratado en este proyecto de grado:

Rotation range=20, aplica una rotación aleatoria a las imágenes, seleccionando un ángulo al azar dentro de un intervalo de [20,+20] grados. El propósito de esta transformación es lograr que el modelo desarrolle invarianza a la orientación. Al exponerlo a múltiples rotaciones de un mismo objeto, el modelo aprende a reconocerlo independientemente de su inclinación, lo cual es crucial para manejar las variaciones de perspectiva que ocurren en escenarios del mundo real.

Width shift range=0.2, realiza traslaciones horizontales aleatorias, desplazando la imagen hacia la izquierda o la derecha hasta un máximo del 20 % de su ancho total. Esta técnica enseña al modelo a no depender de la centralización perfecta del objeto en el encuadre. Al forzarlo a encontrar y reconocer características en diferentes posiciones horizontales, se mejora significativamente su capacidad para localizar y clasificar objetos sin importar dónde aparezcan en la imagen.

Height shift range=0.2, introduce traslaciones verticales aleatorias, moviendo la imagen hacia arriba o abajo hasta un máximo del 20 % de su altura total. Esta técnica fomenta la invarianza a la posición en el eje vertical.

Shear range=0.2, aplica una transformación de corte que inclina la forma de la imagen a lo largo de un eje, con una intensidad máxima definida por un ángulo de 0.2 radianes. Esta operación distorsiona la perspectiva de la imagen, simulando cómo un objeto puede ser visto desde diferentes ángulos de cámara.

Zoom range=0.2, aplica un acercamiento o alejamiento aleatorio a las imágenes dentro de un rango de [80 %, 120 %] de su tamaño original. Al variar la escala de los objetos durante el entrenamiento, esta técnica hace que el modelo sea robusto a las distancias. En consecuencia, el modelo aprende a identificar un objeto con la misma eficacia tanto si aparece grande y en primer plano, como si se muestra pequeño y a lo lejos.

Horizontal flip=True, se introduce la posibilidad de invertir horizontalmente las imágenes de forma aleatoria con una probabilidad del 50 %, creando un .efecto espejo". Esta es una de las técnicas de aumento más efectivas, ya que asume que la orientación izquierda-derecha de un objeto no altera su identidad.

Fill mode='nearest', define la estrategia para rellenar los píxeles que puedan quedar vacíos después de aplicar transformaciones geométricas como rotaciones o desplazamientos. La opción 'nearest'

asigna a estas nuevas áreas el valor del píxel más cercano del borde de la imagen original. Esta elección asegura que no se introduzcan artefactos o píxeles negros que puedan confundir al modelo, manteniendo la coherencia visual de la imagen transformada de una manera computacionalmente eficiente.

Posteriormente, dentro del mismo codigo, se imprimieron las graficas de validación y las matrices de confusión correspondientes, esto se realizó comparando las predicciones con las etiquetas verdaderas del conjunto de validación (`y_val`) y mediante las funciones `confusion_matrix` y `classification_report` de la biblioteca `scikit-learn`. Estas funciones generaron automáticamente la matriz de confusión y el reporte de clasificación detallado (con las métricas de precisión, recall y F1 Score) que se presentan y analizan en los capítulos de resultados.

Por último, la fase final consistió en el desarrollo y la implementación de una interfaz que integra los modelos entrenados en una interfaz gráfica de usuario (GUI) para el reconocimiento de emociones en tiempo real. Para la construcción de esta herramienta se utilizó un conjunto de bibliotecas de Python de código abierto, seleccionadas por su robustez y compatibilidad: TensorFlow con su API de Keras para la carga y ejecución de los modelos de deep learning, OpenCV para las tareas de visión por computadora en tiempo real (captura de video y detección de rostros), y Kivy como framework para la creación de la interfaz de usuario multiplataforma.

El flujo de trabajo general de la interfaz, diseñada para ser intuitiva, comienza con la interacción del usuario en la pantalla principal. A través de esta interfaz, el usuario selecciona uno de los modelos de reconocimiento previamente entrenados y activa el proceso de detección, tal como se ilustra en el diagrama de flujo general de la **Figura 6.2**.

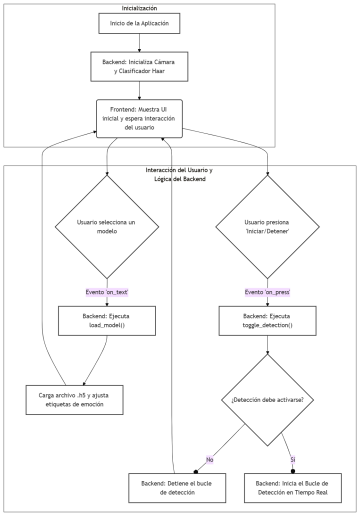


Figura 6.2: Diagrama de flujo de inicio de la interfaz.

Una vez iniciada, se entra en un bucle de procesamiento continuo en tiempo real, cuyo funcionamiento se esquematiza en el diagrama de la **Figura 6.3**. Dentro de este ciclo, el sistema utiliza la cámara para capturar fotogramas, detecta un rostro en ellos, lo procesa y lo pasa al modelo de IA cargado para obtener una predicción de la emoción. El resultado de esta predicción es enviado de vuelta y mostrado instantáneamente en la interfaz gráfica.

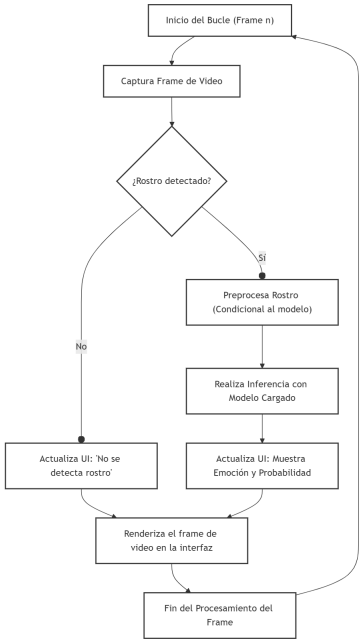


Figura 6.3: Diagrama de flujo del bucle Update.

Cabe destacar que esta es una descripción general del flujo de trabajo. Un análisis técnico profundo sobre la arquitectura de la interfaz, la interacción entre el backend y el frontend, y la explicación detallada del código se presenta en la **Sección 6.2.3.2**.

6.2.2. Tipo de Estudio

Para este proyecto, el tipo de estudio es descriptivo y experimental en entorno controlado, orientado al desarrollo y validación de una interfaz funcional de bajo costo que facilita la posible intervención en habilidades sociales de personas con Trastorno del Espectro Autista (TEA). Al ser un estudio experimental en la creación de dicha interfaz, su objetivo principal es probar y mejorar las capacidades técnicas de la misma en cuanto al reconocimiento de emociones, evaluando su precisión en la identificación de expresiones faciales a través de técnicas de deep learning.

6.2.2.1. Implicaciones Bioéticas y Legales

Aunque el proyecto no involucra experimentación directa con seres humanos, se asegura de cumplir estrictamente con las normativas éticas y legales aplicables a la privacidad y manejo de datos. Este proyecto se limita exclusivamente al uso de imágenes provenientes de bases de datos públicas o con permisos explícitos para fines de investigación. No se recopilarán datos personales, ni se adquirirán imágenes o información directamente de niños con TEA. Tampoco se tomarán fotografías ni se realizarán intervenciones directas, por lo que no es necesario someter el proyecto a la aprobación de un comité de bioética en esta etapa. Estas medidas garantizan el respeto por la privacidad y el cumplimiento de las normativas vigentes de protección de datos.

Dicho esto, tambien es **importante aclarar que**, para demostrar la funcionalidad de la interfaz, en el presente documento se incluirán capturas de pantalla en las que los propios autores de este trabajo actúan como sujetos de prueba. Se recalca que el uso de la interfaz en humanos para fines distintos a esta demostración no está previsto ni autorizado en la fase actual del proyecto (como se acaba de mencionar anteriormente). El único propósito de estas imágenes es validar la operatividad de la interfaz en tiempo real y proporcionar evidencia tangible de la eficacia de los modelos propuestos. Se hace esta aclaración pues es un factor necesario a la hora de evaluar el presente proyecto de grado.

Para proteger la integridad y privacidad de cualquier información visual, el proyecto se compromete a:

- Utilizar únicamente bases de datos de acceso público y con permisos explícitos para la investigación.
- Evitar el almacenamiento o tratamiento de datos personales o sensibles sin consentimiento explícito en fases futuras de desarrollo y prueba.
- Asegurar que, en caso de expandir la investigación con pruebas en usuarios finales, se obtendrá previamente el aval de un comité de bioética y se garantizará el cumplimiento de todas las regulaciones locales y normativas internacionales de investigación con datos de menores de edad y población vulnerable.

Dado que la interfaz se diseñará y evaluará en un entorno controlado sin la participación de usuarios finales durante esta fase, se minimizan los riesgos bioéticos y legales, permitiendo centrarse en la validación técnica del dispositivo.

6.2.3. Actividades

6.2.3.1. Gestion de la base de datos O.E 1

- Se implementó data augmentation de forma automatizada con ImageDataGenerator, de modo que cada clase contara con 4056 imágenes tras el aumento de datos, este proceso se realizó con el fin de equilibrar el conjunto de datos y mejorar la robustez del modelo.
- Para la validación de la carga de datos, en Google Colab se configuró la conexión directa desde la API de Kaggle hacia el notebook, verificando rutas y tamaños de los archivos sin necesidad de descargas en los dispositivos usados. En VS Code, se descargó previamente la base de datos al entorno local de cada computador y se comprobó la integridad de los archivos mediante scripts en Python que contabilizaron el número de imágenes en cada carpeta antes de iniciar el entrenamiento, tal como se puede observar en la figura 6.4 (en caso del codigo destinado a detección de 3 clases) y en la figura 6.5 (en caso del codigo destinado a detección de 7 clases), para posteriormente comprobar el número de archivos después del proceso de data augmentation.

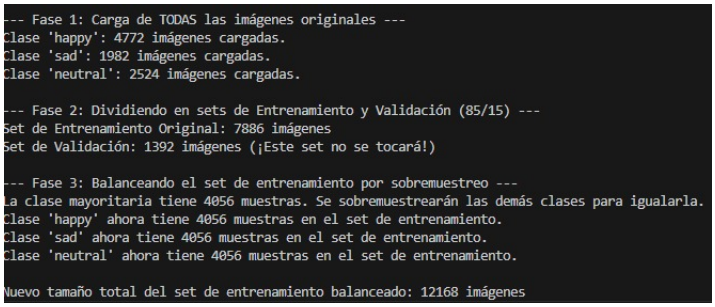


Figura 6.4: Gestión base de datos 3 clases.

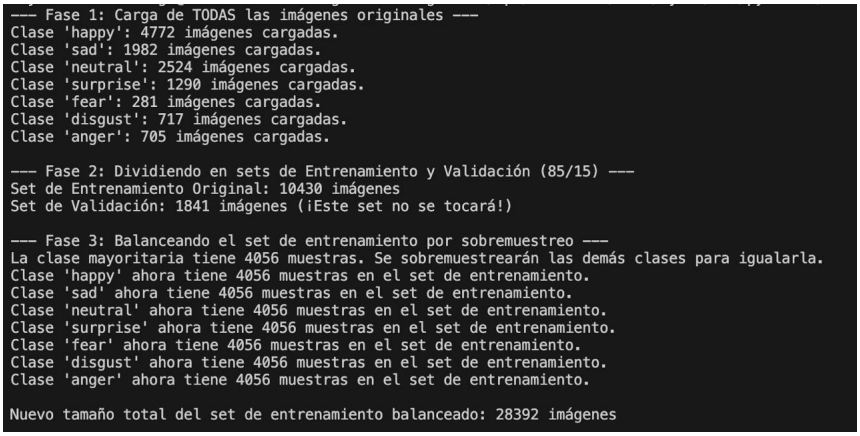


Figura 6.5: Gestión base de datos 7 clases.

- Por ultimo, para ofrecer una visión secuencial de todo el flujo de trabajo de preparación de datos ya mencionado, desde la carga inicial hasta la generación de los conjuntos finales para el entrenamiento, se ha elaborado el diagrama de flujo que se presenta en la Figura 6.6.

  Este diagrama esquematiza la lógica procedimental seguida. Una de las fases más importantes que el diagrama clarifica es la aplicación de aumento de datos (*data augmentation*) de manera exclusiva sobre el conjunto de entrenamiento. Finalmente, el proceso concluye con un paso de verificación para confirmar que la gestión de los datos se realizó correctamente.

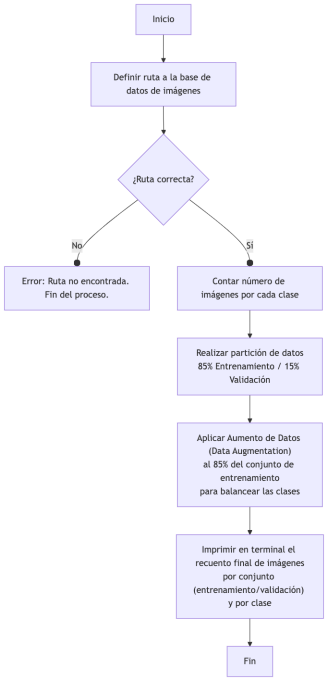


Figura 6.6: Proceso de gestión de datos

6.2.3.2. Modelos de deep learning O.E 2

Modelo 1: Red Neuronal Convolucional (CNN) Personalizada Para la implementación de los modelos se utilizaron principalmente las bibliotecas de Python `TensorFlow`, con su API de alto nivel `Keras`, para la construcción y entrenamiento de las redes neuronales; `OpenCV` para el procesamiento de imágenes; y `scikit-learn` para la evaluación de métricas.

Una vez completada la fase de gestión y preparación de la base de datos, como se describió en la sección anterior, el siguiente paso metodológico fue la construcción y entrenamiento del primer modelo: una Red Neuronal Convolucional (CNN) diseñada a medida.

La arquitectura de este modelo, cuya implementación se detalla en la **Figura 6.7**, se compone de dos partes principales: un extractor de características convolucional y un clasificador..

1. **Extractor de Características:** Consiste en tres bloques convolucionales secuenciales:
   - **Bloque 1:** Una capa `Conv2D` con **32 filtros** (kernel 3x3, activación `ReLU`), seguida de una capa `MaxPooling2D` (2x2).
   - **Bloque 2:** Una capa `Conv2D` con **64 filtros** (kernel 3x3, activación `ReLU`), seguida de una capa `MaxPooling2D` (2x2).
   - **Bloque 3:** Una capa `Conv2D` con **128 filtros** (kernel 3x3, activación `ReLU`), seguida de una capa `MaxPooling2D` (2x2).
2. **Clasificador:** Tras la extracción de características, el tensor de salida se aplana con una capa `Flatten` y se pasa a un clasificador compuesto por:
   - Una capa densa (`Dense`) con **256 neuronas** y activación `ReLU`.
   - Una capa de regularización `Dropout` con una tasa de **0.5**.
   - Una capa final `Dense` con activación `softmax`, cuya salida se ajustó a **3 neuronas** para el primer experimento y a **7 neuronas** para el segundo.

```
# Modelo CNN
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(IMG_SIZE,
    IMG_SIZE, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, <strong>3</strong>), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(len(class_names), activation='softmax')
])

# Compilacion
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['
    accuracy'])

# Entrenamiento
history = model.fit(train_generator, epochs=40, validation_data=(X_val, y_val_cat)
    )
```

Figura 6.7: Implementación de la arquitectura de la CNN personalizada.

Configuración de Hiperparámetros y Entrenamiento Para asegurar la replicabilidad, a continuación se detallan los hiperparámetros clave utilizados en la compilación y entrenamiento de este modelo:

- **Optimizador:** Se utilizó `Adam` con su tasa de aprendizaje por defecto.
- **Función de Pérdida:** `categorical_crossentropy`.
- **Métrica de Evaluación:** `accuracy`.
- **Tamaño del Lote (`batch_size`):** 32.
- **Número de Épocas (`epochs`):** 40.
- **Tasa de Dropout:** 0.5.
- **Dimensiones de Entrada (`input_shape`):** 100x100x3 píxeles.

Modelo 2: VGG16 con Transfer Learning Para evaluar la eficacia de arquitecturas más profundas y pre-entrenadas, el segundo modelo implementado se basó en la arquitectura **VGG16**. Se empleó la técnica de *transfer learning*. Este enfoque aprovecha el conocimiento aprendido por VGG16 en el conjunto de datos ImageNet, utilizando su base convolucional como un extractor de características de alto nivel, sin alterar sus pesos. La implementación del modelo, mostrada en la **Figura 6.8**, se estructuró de la siguiente manera:

1. **Modelo Base:** Se instanció el modelo VGG16 sin su cabezal clasificador original (include_top=False) y se cargaron los pesos pre-entrenados de ImageNet. Crucialmente, todas las capas de este modelo base se configuraron como no entrenables (base_model.trainable = False).
2. **Clasificador:** Sobre la salida del modelo base congelado, se construyó un nuevo clasificador, compuesto por:
   - Una capa Flatten para convertir los mapas de características multidimensionales en un vector unidimensional.
   - Una capa densa (`Dense`) con **256 neuronas** y función de activación `ReLU`.
   - Una capa de regularización `Dropout` con una tasa de **0.5**.
   - Una capa final `Dense` con activación `softmax`, configurada con **3 neuronas** en el caso del modelo de reconocimiento de 3 clases y **7 neuronas** en el caso del modelo de reconocimiento de 7 clases.

```python
# Transfer learning con VGG16
base_model = VGG16(include_top=False, weights='imagenet', input_shape=(IMG_SIZE,
    IMG_SIZE, 3))
base_model.trainable = False # Congelamos las capas de VGG16

model = models.Sequential([
    base_model,
    layers.Flatten(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_classes, activation='softmax') # 3 o 7 clases
])

model.summary()

# Compilacion
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['
    accuracy'])

# Entrenamiento
history = model.fit(train_generator, epochs=40, validation_data=(X_val, y_val_cat)
    )
```

Figura 6.8: Implementación del modelo VGG16 con transfer learning.

Esta configuración de trainable = False tiene una implicación directa en el número de parámetros que se optimizan durante el entrenamiento. Como se puede observar en el resumen de los modelos en las **Figuras 6.9 y 6.10**, la base VGG16 contiene **14,714,688 parámetros no entrenables** (non-trainable), los cuales se mantuvieron fijos. El proceso de optimización se centró exclusivamente en los **aproximadamente 1.18 millones de parámetros entrenables** (trainable) del clasificador añadido. La gran mayoría de estos (1,179,904) pertenecen a la primera capa densa, mientras que el resto corresponde a la capa de salida final, variando ligeramente entre el modelo de 3 clases (771 parámetros) y el de 7 clases (1,799 parámetros). En efecto, el entrenamiento consistió en ajustar únicamente un clasificador relativamente pequeño que aprende a interpretar el vasto conjunto de características extraídas por la potente pero estática base VGG16.

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 3, 3, 512)	14,714,688
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1,179,904
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 3)	771

Total params: 15,895,363 (60.64 MB)
Trainable params: 1,180,675 (4.50 MB)
Non-trainable params: 14,714,688 (56.13 MB)

Figura 6.9: Parámetros del modelo VGG16 (3 clases).

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 3, 3, 512)	14,714,688
flatten (Flatten)	(None, 4608)	0
dense (Dense)	(None, 256)	1,179,904
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 7)	1,799

Total params: 15,896,391 (60.64 MB)
Trainable params: 1,181,703 (4.51 MB)
Non-trainable params: 14,714,688 (56.13 MB)

Figura 6.10: Parámetros del modelo VGG16 (7 clases).

Configuración de Hiperparámetros y Entrenamiento Los hiperparámetros utilizados para la compilación y entrenamiento de los modelos VGG16 fueron consistentes con los del modelo anterior para permitir una comparación directa:

- **Optimizador:** Se utilizó `Adam` con su tasa de aprendizaje por defecto.
- **Función de Pérdida:** `categorical_crossentropy`.
- **Métrica de Evaluación:** `accuracy`.
- **Tamaño del Lote (`batch_size`):** 32.
- **Número de Épocas (`epochs`):** 40.
- **Preprocesamiento de Imagen:** Normalización simple de los valores de los píxeles al rango [0, 1] mediante la división por 255.0.

Es fundamental aclarar que, dado que las capas del modelo base VGG16 se mantuvieron congeladas, la técnica utilizada fue la de extracción de características. No se realizó un proceso de ajuste fino (fine-tuning), el cual implicaría descongelar y re-entrenar algunas de las capas convolucionales del modelo base.

Modelo 3: ResNet50 El tercer y último modelo evaluado se basó en la arquitectura **ResNet50**. Al igual que los modelos anteriores, se implementó en su modalidad de **extracción de características**, congelando los pesos de la base pre-entrenada. La implementación de este enfoque, como se muestra en el código de la **Figura 6.11**, incluyó una serie de mejoras metodológicas clave:

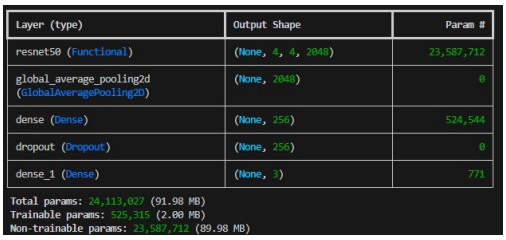
1. **Preprocesamiento Específico:** Se utilizó la función `preprocess_input` propia de la familia de modelos ResNet, para asegurar que los datos de entrada tuvieran la misma distribución que los datos con los que el modelo fue originalmente entrenado (ImageNet).
2. **Optimización de la Arquitectura del Clasificador:** Se sustituyó la capa `Flatten` por una capa `GlobalAveragePooling2D`. Esta técnica reduce drásticamente el número de parámetros en el modelo, buscando mitigar el sobreajuste que se experimentó en versiones preliminares.
3. **Ajuste de Hiperparámetros:** Se utilizó una tasa de aprendizaje explícitamente baja y se introdujo una técnica de regularización adicional en la función de pérdida, como se detalla más adelante.

```
# Transfer learning con ResNet50
base_model = ResNet50(include_top=False, weights='imagenet', input_shape=(IMG_SIZE
    , IMG_SIZE, 3))
base_model.trainable = False

model = models.Sequential([
    base_model,
    # Se usa GlobalAveragePooling2D en lugar de Flatten
    layers.GlobalAveragePooling2D(),
    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(num_classes, activation='softmax')
])

model.summary()

# Se usa una tasa de aprendizaje mas baja para el optimizador Adam
optimizer = Adam(learning_rate=1e-4) # 1e-4 es 0.0001
model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['
    accuracy'])

# Entrenamiento
history = model.fit(train_generator, epochs=40, validation_data=(X_val, y_val_cat)
    )
```

Figura 6.11: Implementación del modelo ResNet50.

La arquitectura del clasificador consistió en la capa `GlobalAveragePooling2D`, seguida de una capa `Dense` con **256 neuronas** (`ReLU`), una capa `Dropout` con una tasa de **0.5**, y la capa de salida `softmax` final, ajustada a **3 o 7 neuronas** según el modelo.

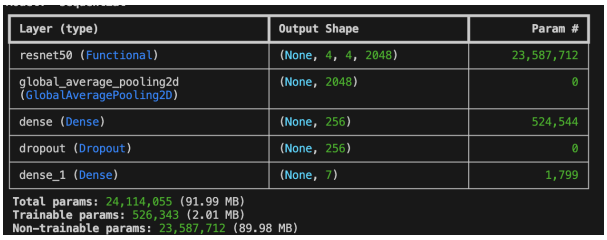
El uso de `GlobalAveragePooling2D` tuvo un impacto significativo en la complejidad del modelo. Como se detalla en el resumen arrojado por el sistema en las **Figuras 6.12 y 6.13**, la base ResNet50 contiene **23,587,712 parámetros no entrenables**. Gracias a esta capa, el número de parámetros entrenables en el clasificador fue de solo **aproximadamente 525,000**. Esta cifra representa una **reducción de más del 55 %** en comparación con los 1.18 millones de parámetros entrenables del modelo VGG16, que utilizaba una capa `Flatten`.



Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 4, 4, 2048)	23,587,712
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 256)	524,544
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 3)	771

Total params: 24,113,027 (91.98 MB)
Trainable params: 525,315 (2.00 MB)
Non-trainable params: 23,587,712 (89.98 MB)

Figura 6.12: Parámetros del modelo ResNet50 (3 clases).



Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 4, 4, 2048)	23,587,712
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 256)	524,544
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 7)	1,799

Total params: 24,114,055 (91.99 MB)
Trainable params: 526,343 (2.01 MB)
Non-trainable params: 23,587,712 (89.98 MB)

Figura 6.13: Parámetros del modelo ResNet50 (7 clases).

Configuración de Hiperparámetros y Entrenamiento Los hiperparámetros se ajustaron con el objetivo de lograr un entrenamiento más estable:

- **Optimizador:** Se utilizó `Adam` con una tasa de aprendizaje (*learning rate*) reducida de `1e-4`.
- **Función de Pérdida:** Para el modelo de 3 clases se usó `categorical_crossentropy`. Para el modelo de 7 clases, se introdujo una mejora: se utilizó la misma función pero con **suavizado de etiquetas (*label smoothing*) de 0.1**, una técnica de regularización que previene que el modelo se vuelva excesivamente confiado en sus predicciones.
- **Métrica de Evaluación:** `accuracy`.
- **Tamaño del Lote (`batch_size`):** 32.
- **Número de Épocas (`epochs`):** 40.

Nuevamente, es importante subrayar que el enfoque fue la **extracción de características**. Las capas del modelo ResNet50 permanecieron congeladas, por lo que **no se realizó un proceso de ajuste fino (*fine-tuning*)**.

6.2.3.3. Evaluación y selección del modelo O.E 3

Con el fin de seleccionar el modelo más eficiente, se establecieron como métricas clave la exactitud (accuracy), precisión, recuperación (recall) y F1-score. También se elaboraron matrices de confusión para observar el comportamiento del sistema ante cada clase emocional, identificando aciertos y confusiones más frecuentes.

Se utilizó un esquema de validación mediante partición del 15 % del total de datos, asegurando una distribución estratificada. Los resultados fueron registrados por época, graficando la evolución de la precisión y la pérdida para evaluar estabilidad, convergencia y sobreajuste.

En la clasificación de 3 clases, el modelo CNN personalizado fue el que logró mayor precisión y estabilidad. Su rendimiento fue consistente a lo largo de las épocas, destacándose en la diferenciación

entre emociones.

En la clasificación de 7 clases, VGG16 y ResNet50 mostraron una desventaja frente a la CNN personalizada, nuevamente. Se lograron clasificaciones más precisas incluso en emociones con gestos similares, como "enojo" y "disgusto" por lo cual el modelo CNN personalizado se termino escogiendo como modelo predilecto para el contexto de este trabajo.

6.2.3.4. Desarrollo de la interfaz del usuario O.E 4

Se implementó una interfaz gráfica de usuario (GUI) utilizando la biblioteca de código abierto Kivy para Python. Esta elección permitió desarrollar una interfaz diseñada específicamente para una posible ayuda a los niños con Trastorno del Espectro Autista (TEA) a conectar con su entorno. Su propósito es facilitar el reconocimiento de las emociones de las personas que los rodean, actuando como un puente entre la realidad y el mundo emocional del niño para que pueda comprender mejor las interacciones que se dan a su alrededor.

El diseño de la interfaz priorizó fundamentalmente el minimalismo. Cada decisión de diseño se guio por la pregunta: *¿Es este elemento esencial para la tarea de reconocimiento de emociones, o impone una carga cognitiva innecesaria?*. Este enfoque busca eliminar sistemáticamente cualquier elemento estético o funcional no esencial que pueda actuar como una barrera para el aprendizaje, fundamentado en el perfil neurocognitivo y sensorial de las personas con TEA.

Un principio fundamental en el diseño para usuarios con TEA es el reconocimiento de su procesamiento sensorial atípico. Muchas personas con TEA experimentan hiper o hiporreactividad a los estímulos sensoriales [69]. En el contexto de una interfaz de usuario, esto significa que los elementos visuales estándar, como colores brillantes, objetos o patrones, pueden ser percibidos con una intensidad igual o incluso superior a la del contenido principal. Un entorno digital que para un usuario neurotípico puede ser atractivo o dinámico, para un usuario autista puede convertirse rápidamente en abrumador. Esta sobrecarga sensorial no es una cuestión de preferencia, sino una respuesta fisiológica genuina que puede desencadenar una intensa ansiedad y estrés, un estado que a menudo se describe como una "tormenta sensorial". El cerebro autista a menudo procesa la información visual de manera diferente, asimilando más detalles simultáneamente sin los mecanismos de filtrado típicos. Esta falta de un mecanismo de filtrado neurológico es un concepto crítico para el diseño de interfaces, pues implica que cada elemento visual en la pantalla, ya sea funcional o meramente decorativo, compite por la atención del usuario en igualdad de condiciones [70]

Arquitectura de la interfaz: Separación Backend y Frontend Una decisión arquitectónica clave fue separar la lógica de la interfaz de su diseño visual. Esta práctica promueve la modularidad y mantenibilidad del código.

- **Backend (Lógica en `emotion_app.py`):** Este archivo Python alberga el núcleo computacio-

nal. Es responsable de toda la lógica de procesamiento: la inicialización de la cámara y los clasificadores, la carga de los modelos de TensorFlow, el procesamiento de cada fotograma de video, la ejecución de las predicciones y la gestión del estado general de la interfaz.

- **Frontend (Diseño en `emotionapp.kv`):** Este archivo utiliza el lenguaje declarativo de Kivy para definir la totalidad de la interfaz gráfica. Describe la jerarquía de widgets (botones, etiquetas, etc.), su disposición en la pantalla (layouts), y sus propiedades visuales (color, tamaño, fuente).

La conexión entre ambos archivos es uno de los puntos fuertes de Kivy. A través del sistema de IDs y el enlace de eventos (*event binding*), como por ejemplo `on_press`, las acciones realizadas en el frontend (`.kv`) invocan directamente métodos definidos en el backend (`.py`), y este último puede actualizar las propiedades de los widgets del frontend en tiempo real.

Análisis Detallado del Backend (`emotion_app.py`) El script de Python gestiona el estado y el flujo de la interfaz a través de varios métodos y atributos clave:

- **Inicialización (`__init__`) y Atributos de Clase:** Al crearse la instancia de la interfaz, se inicializan sus atributos de estado principales:
  - `self.face_cascade`: Carga el clasificador en cascada Haar (`haarcascade_frontalface_default.xml`) de OpenCV, un algoritmo basado en características que es rápido y eficiente para la detección de objetos.
  - `self.capture`: Inicia la conexión con la cámara del dispositivo (`cv2.VideoCapture(0)`).
  - `self.model`, `self.model_path`, `self.emotion_labels`: Atributos inicializados como vacíos que almacenarán el modelo de Keras cargado, su ruta y las etiquetas de emoción correspondientes.
  - `self.detection_active`: Un booleano que actúa como interruptor para iniciar o detener el proceso de detección.
- **Carga Dinámica de Modelos (`load_model`):** Este método es invocado desde el widget `Spinner` de la interfaz. Tal como se puede observar en la figura 6.14, al seleccionar un modelo, el método recibe su nombre y ejecuta la siguiente lógica:
  - Obtiene la ruta del archivo `.h5` desde un diccionario `MODELS`.
  - Utiliza un bloque `try...except` para una carga robusta. Si ocurre un error (ej. archivo no encontrado), se notifica al usuario en la interfaz.
  - Si la carga es exitosa, el modelo se almacena en `self.model` usando `tf.keras.models.load_model()`.
  - **Lógica Condicional:** Se inspecciona el nombre del archivo (`self.model_path`). Si contiene la cadena "7clases", el atributo `self.emotion_labels` se puebla con la lista de siete emociones; de lo contrario, se puebla con la lista de tres. Esto permite que el programa maneje ambos tipos de modelos de forma flexible.

```
def load_model(self, model_name):
    if self.detection_active:
        self.toggle_detection() # Si la deteccion esta activa, la detenemos

    model_path = MODELS.get(model_name)
    if model_path is None:
        self.model = None
        self.ids.active_model_label.text = "Ningun modelo cargado"
        return

    try:
        self.ids.active_model_label.text = f"Cargando: {model_name}..."
        self.model = tf.keras.models.load_model(model_path)
        self.model_path = model_path
        self.ids.active_model_label.text = f"Modelo Activo: {model_name}"

        # Actualizamos las etiquetas segun el modelo cargado
        if "7clases" in self.model_path:
            self.emotion_labels = ['happy', 'sad', 'neutral', 'surprise', 'fear',
   'disgust', 'anger']
        else:
            self.emotion_labels = ['Happy', 'Sad', 'Neutral']

        print(f"Modelo '{model_name}' cargado. Etiquetas: {self.emotion_labels}")

    except Exception as e:
        self.ids.active_model_label.text = "Error al cargar el modelo"
        print(f"Error: {e}")
```

Figura 6.14: Fragmento de código del método `load_model`

- **Bucle Principal de Detección (`update`):** Este es el corazón de la funcionalidad en tiempo real. Es invocado repetidamente por `Clock.schedule_interval(self.update, 1.0 / 30.0)`. Este comando le instruye a Kivy que intente ejecutar el método `update` 30 veces por segundo (el cual se muestra en la figura 6.16) y se explica graficamente el diagrama de flujo en la figura 6.15, estableciendo una tasa de refresco objetivo de 30 FPS (fotogramas por segundo). La tasa de FPS real puede ser inferior dependiendo de la carga computacional y el hardware. En cada ejecución, se realiza el siguiente pipeline:

  1. **Captura de Frame:** Se lee un fotograma de la cámara.
  2. **Detección de Rostro:** El frame se convierte a escala de grises y se pasa al clasificador Haar. El método `detectMultiScale` devuelve una lista de rectángulos con los rostros encontrados. Para optimizar el rendimiento, el código está diseñado para procesar solo el primer rostro detectado en cada fotograma.
  3. **Preprocesamiento Condicional del Rostro:** El rostro detectado se recorta y re-

dimensiona a 100x100 píxeles. En un paso crítico para la replicabilidad, se aplica un preprocesamiento que depende del modelo activo. Si el nombre del modelo contiene "ResNet50", se usa la función `resnet50_preprocess`; en los demás casos (VGG16 y CNN personalizada), se aplica una normalización simple dividiendo por 255.0. Esto asegura que los datos de entrada para la inferencia coincidan con los datos de entrenamiento de cada modelo.

4. **Inferencia:** El rostro preprocesado se pasa al método `self.model.predict()`, que devuelve un vector de probabilidades.
5. **Actualización de la Interfaz:** Se determina el índice de la emoción con la probabilidad más alta usando `np.argmax`. El valor de esta probabilidad (la "certeza" del modelo) se extrae del vector de predicción y se convierte a un valor porcentual. Finalmente, ambos datos se utilizan para actualizar las propiedades de los widgets del frontend mediante sus IDs; por ejemplo, se actualiza la propiedad `text` de la etiqueta (`self.ids.emotion_label.text = ...`) y la propiedad `value` de la barra de certeza (`self.ids.probability_bar.value = ...`).
6. **Renderizado del Video:** El frame de OpenCV (un array de NumPy) se convierte a una textura de Kivy (`kivy.graphics.texture.Texture`) y se asigna al widget de imagen para mostrar el video en vivo.

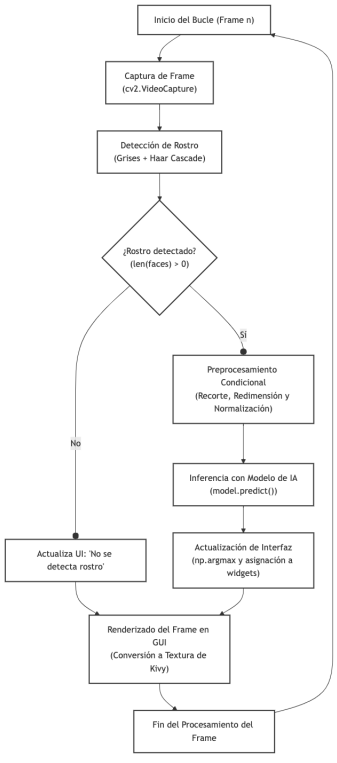


Figura 6.15: Diagrama de flujo detallado de procesamiento por fotograma

```
def update(self, dt):
    ret, frame = self.capture.read()
    if not ret: return

    frame = cv2.flip(frame, 1)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = self.face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors
   =5, minSize=(50, 50))

    if len(faces) > 0:
        (x, y, w, h) = faces[0]
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
        face_img = frame[y:y + h, x:x + w]
        face_resized = cv2.resize(face_img, (IMG_SIZE, IMG_SIZE))

        face_to_predict = face_resized.astype('float32')

        # self.model_path para la logica condicional
        if 'ResNet50' in self.model_path:
            face_processed = resnet50_preprocess(face_to_predict)
        elif 'VGG16' in self.model_path:
            face_processed = face_to_predict / 255.0
        else:
            face_processed = face_to_predict / 255.0

        face_expanded = np.expand_dims(face_processed, axis=0)
        prediction = self.model.predict(face_expanded)[0]
        emotion_index = np.argmax(prediction)

        if emotion_index < len(self.emotion_labels):
            emotion_text = self.emotion_labels[emotion_index]
            probability = prediction[emotion_index] * 100
            self.ids.emotion_label.text = f"Emocion: {emotion_text}"
            self.ids.probability_bar.value = probability
        else:
            self.ids.emotion_label.text = "Error: Indice de emocion"
            self.ids.probability_bar.value = 0
    else:
        self.ids.emotion_label.text = "Emocion: No se detecta rostro"
        self.ids.probability_bar.value = 0

    buf = cv2.flip(frame, 0).tobytes()
    texture = Texture.create(size=(frame.shape[1], frame.shape[0]), colorfmt='bgr'
   )
    texture.blit_buffer(buf, colorfmt='bgr', bufferfmt='ubyte')
    self.ids.video.texture = texture
```

Figura 6.16: Fragmento del codigo Update

Análisis Detallado del Frontend (`emotionapp.kv`) El archivo `.kv` define la estructura visual y la interacción.

- **Estructura y Layouts:** Se utiliza un `FloatLayout` como contenedor principal para permitir la superposición de elementos, como la etiqueta de bienvenida sobre el área de video. En la parte inferior, un `BoxLayout` organiza los controles (etiquetas, barra de certeza, menú desplegable y botón) de manera estructurada. El diseño es deliberadamente minimalista para centrar la atención en la interacción del usuario y la retroalimentación de la emoción.
- **Enlace de Eventos:** La interacción se logra mediante el enlace directo de eventos de los widgets a los métodos de la clase principal de Python, tal como se muestra en la figura 6.17:
  - **Spinner:** El evento `on_text` de este widget se activa cada vez que el usuario selecciona una opción. La instrucción `on_text: root.load_model(self.text)` llama al método `load_model` en el script de Python y le pasa el texto del elemento seleccionado como argumento.
  - **Button:** De forma similar, el evento `on_press` del botón "Iniciar/Detener" está enlazado directamente al método `root.toggle_detection()`, actuando como el interruptor principal del programa.

```
Spinner:
    id: model_spinner
    text: 'Seleccionar Modelo'
    values: list(app.MODELS.keys())
    on_text: root.load_model(self.text)
    size_hint_x: 0.6

Button:
    text: 'Iniciar/Detener'
    on_press: root.toggle_detection()
    size_hint_x: 0.4
```

Figura 6.17: Fragmento de codigo Kivy

Para finalizar y resumir toda esta información, es determinante recalcar que la arquitectura adquiere relevancia cuando se realiza una interacción en un contexto real, donde se presionan los botones de la interfaz. A continuación, se muestra la versión final e interactiva de la interfaz en la figura 6.18 y se detalla el flujo de ejecución para las dos interacciones principales del usuario con el fin de sintetizar lo mencionado anteriormente y ver su relevancia cuando se esta ejecutando la interfaz.

Primero se inicia la interfaz, para esto se tomó la decisión de que solo se ejecutara el código de python y que se empezará la interacción, se determino no crear un ejecutable local y en vez de eso se subieron los archivos necesarios y los codigos a GitHub. Una vez iniciada la interfaz (desde el codigo) se verá en pantalla los siguientes botones:

El enlace del repositorio de GitHub se encuentra disponible en el siguiente enlace: https://github.com/msgonzalez31/SISTEMA-INTERFAZ-DE-RECONOCIMIENTO-DE-EMOCIONES

Botón "Iniciar/Detener" Cuando el usuario presiona este botón, se ejecuta el método toggle_detection() en el código de Python.

- **En el archivo** `.kv`**:** El botón tiene la instrucción `on_press: root.toggle_detection()`. Esto significa "cuando me presionen, llama a la función toggle_detection() que está en el código principal".
- **En el archivo** `.py`**:** El método `toggle_detection()` actúa como un interruptor. Si la detección está detenida, inicia un bucle que llama a la función `update()` 30 veces por segundo (`Clock.schedule_interval`). Si la detección ya está en marcha, detiene ese bucle (`self.update_event.cancel()`), pausando el análisis de video.

Menú Desplegable "Seleccionar Modelo" Tal como se muestra en la figura 6.19, el usuario puede seleccionar un modelo del menú desplegable y se ejecuta el método `load_model(self, model_name)` en el código de Python.

- **En el archivo** `.kv`**:** El menú `Spinner` tiene la instrucción `on_text: root.load_model(self.text)`. Esto significa "cuando el texto de mi selección cambie, llama a la función `load_model` y pásale mi nuevo texto como argumento".
- **En el archivo** `.py`**:** El método `load_model()` recibe el nombre del modelo y busca la ruta del archivo `.h5` correspondiente en el diccionario `MODELS`. Luego, usa `tf.keras.models.load_model()` para cargar el modelo de inteligencia artificial en la memoria y ajusta las etiquetas de las emociones (`emotion_labels`) según el modelo sea de 3 o 7 clases.

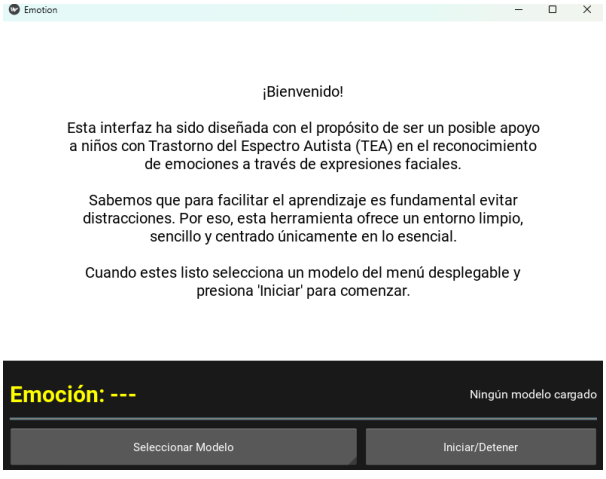


Figura 6.18: Interfaz inicial

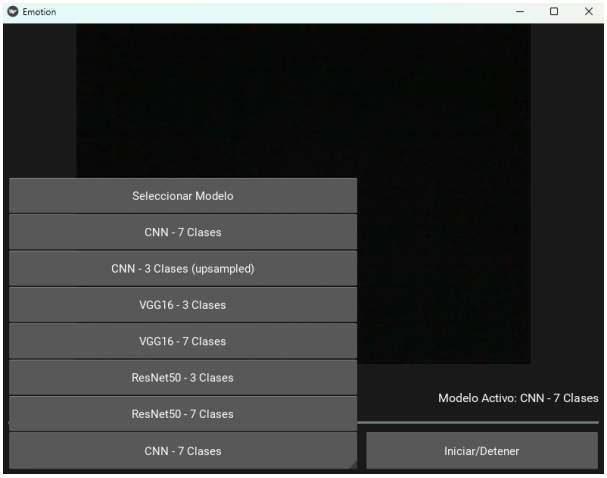


Figura 6.19: Selección de modelo

Capítulo 7

Resultados y Discusión

7.1. Resultados y Discusión de Modelos - 3 Clases

Este capítulo presenta los resultados obtenidos tras el entrenamiento de tres modelos distintos para la clasificación de emociones (*Happy, Sad, Neutral*) a partir de imágenes faciales. Los modelos evaluados incluyen: una red neuronal convolucional (CNN) personalizada, un modelo basado en aprendizaje por transferencia con VGG16, y otro con ResNet50.

7.1.1. Modelo CNN Personalizado (3 Clases)

En esta sección se presentan y analizan los resultados obtenidos por el modelo de Red Neuronal Convolucional (CNN) personalizado, entrenado para la clasificación de tres expresiones faciales básicas: felicidad (happy), tristeza (sad) y neutralidad (neutral). La evaluación del rendimiento del modelo se realiza a través del análisis de sus curvas de aprendizaje, la matriz de confusión y el reporte de clasificación detallado.

7.1.1.1. Análisis de las Curvas de Aprendizaje (Precisión y Pérdida)

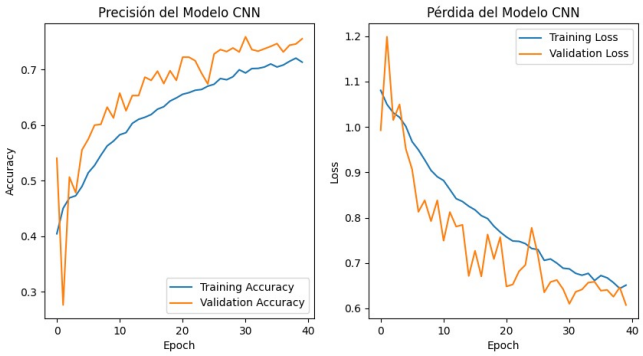


Figura 7.1: Graficas CNN personalizada 3 clases

Las curvas de aprendizaje, mostradas en la figura 7.1, ofrecen una visión fundamental del comportamiento del modelo durante las 40 épocas de entrenamiento.

- **Curva de Precisión (Accuracy):** El gráfico de la izquierda muestra que tanto la precisión de entrenamiento (*Training Accuracy*) como la de validación (*Validation Accuracy*) siguen una

tendencia ascendente clara, lo que indica que el modelo está aprendiendo patrones relevantes de los datos a medida que avanzan las épocas. Un comportamiento destacable es que la curva de validación se mantiene consistentemente por encima o muy cerca de la curva de entrenamiento. Esto puede parecer contraintuitivo, pero sugiere que el modelo generaliza bien y que las técnicas de regularización implementadas (como podría ser Dropout) están cumpliendo su función, siendo más estrictas durante la fase de entrenamiento que durante la validación.

- **Curva de Pérdida (Loss):** El gráfico de la derecha corrobora estos hallazgos. La pérdida de entrenamiento (*Training Loss*) y la de validación (*Validation Loss*) disminuyen progresivamente. La curva de pérdida de validación, aunque más volátil, sigue la tendencia decreciente de la curva de entrenamiento y, en general, se mantiene por debajo de esta. Este comportamiento es un fuerte indicador de que el modelo no está sufriendo de un sobreajuste (*overfitting*) severo, ya que la pérdida en los datos no vistos no comienza a aumentar. La volatilidad en la curva de validación es esperable y refleja la variabilidad inherente en el subconjunto de datos de validación.

7.1.1.2. Análisis de la Matriz de Confusión

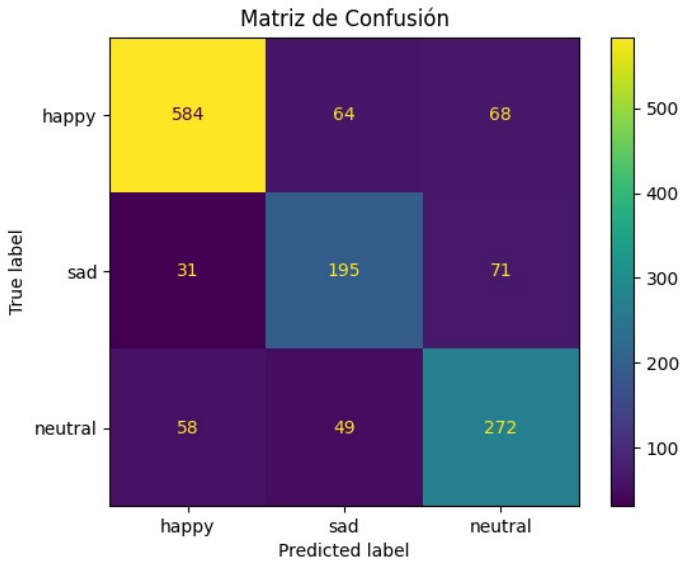


Figura 7.2: Matriz de confusión CNN 3 clases

La matriz de confusión, presentada en figura 7.2, permite un análisis cualitativo detallado de los aciertos y errores del modelo para cada una de las tres clases.

- **Alto Rendimiento en la Clase 'Happy':** El modelo demuestra un rendimiento sobresaliente al clasificar la expresión de felicidad. De un total de 716 imágenes de esta clase, predijo correctamente 584, logrando una alta tasa de verdaderos positivos.

- **Confusión entre 'Sad' y 'Neutral':** La principal debilidad del modelo reside en la distinción entre las clases 'sad' y 'neutral'. Se observa que 71 de las 297 imágenes de 'sad' fueron incorrectamente clasificadas como 'neutral', y 49 de las 379 imágenes de 'neutral' fueron clasificadas como 'sad'. Esta confusión es comprensible desde una perspectiva visual, ya que las expresiones de tristeza sutil y de neutralidad pueden compartir características faciales muy similares, lo que representa un desafío mayor para el modelo.

7.1.1.3. Análisis del Reporte de Clasificación

Cuadro 7.1: Métricas por clase del modelo CNN personalizado (3 clases)

Clase	Precisión	Recall	F1-score	Soporte
Happy	0.87	0.82	0.84	716
Sad	0.63	0.66	0.64	297
Neutral	0.66	0.72	0.69	379
Accuracy			0.76	1392
Macro avg	0.72	0.73	0.72	1392
Weighted avg	0.76	0.76	0.76	1392

El reporte de clasificación, mostrado en el cuadro 7.1, cuantifica el rendimiento del modelo utilizando métricas estándar:

- **Precisión General (Accuracy):** El modelo alcanzó una precisión global del **76 %** sobre el conjunto de validación, lo que indica que clasificó correctamente tres de cada cuatro imágenes aproximadamente. Este es un resultado sólido para un problema de clasificación de tres clases.
- **Métricas por Clase:**
  - **Happy:** Presenta las mejores métricas, con una precisión de **0.87**, una sensibilidad (*recall*) de **0.82** y una puntuación F1 de **0.84**, confirmando que es la clase mejor reconocida.
  - **Sad:** Muestra el rendimiento más bajo, con una puntuación F1 de **0.64**, reflejando la dificultad del modelo para distinguir esta clase.
  - **Neutral:** Obtiene un rendimiento intermedio con una puntuación F1 de **0.69**.
- **Promedios (Averages):** El promedio ponderado (*weighted avg*) de 0.76 es igual a la precisión global. El promedio macro (*macro avg*) de 0.72, que trata a todas las clases por igual, es ligeramente inferior, reflejando el menor rendimiento en las clases 'sad' y 'neutral'.

7.1.1.4. Síntesis de los Resultados

En resumen, el modelo CNN personalizado para 3 clases demuestra un rendimiento general competente con una precisión del **76 %**. Su principal fortaleza es la identificación inequívoca de la expresión de "felicidad". Su principal área de mejora es la diferenciación entre las expresiones más sutiles y visualmente similares de "tristeza" y "neutralidad". El análisis de las curvas de aprendizaje confirma que el entrenamiento fue robusto y no presentó signos de sobreajuste significativo, validando la arquitectura y el proceso de entrenamiento implementados.

.

7.1.2. Modelo VGG16 (Transfer Learning)

En esta sección se evalúa el rendimiento del modelo basado en la arquitectura VGG16, adaptado mediante *transfer learning* para la clasificación de las tres expresiones faciales (felicidad, tristeza y neutralidad). Al igual que con el modelo anterior, el análisis se basa en las curvas de aprendizaje, la matriz de confusión y las métricas de clasificación.

7.1.2.1. Análisis de las Curvas de Aprendizaje (Precisión y Pérdida)

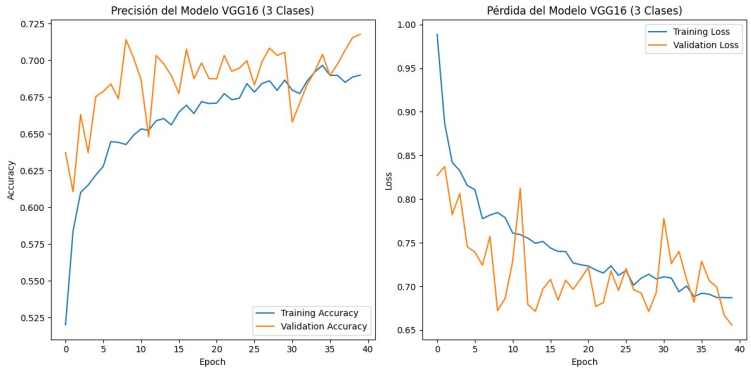


Figura 7.3: Graficas VGG16 3 clases

Las gráficas de entrenamiento para el modelo VGG16, visibles en la figura 7.3, revelan características importantes sobre su proceso de aprendizaje.

- **Curva de Precisión (Accuracy):** Se observa una tendencia general ascendente tanto para la precisión de entrenamiento como para la de validación, alcanzando un valor final en torno al 72 %. Sin embargo, a diferencia del modelo CNN personalizado, la curva de validación del VGG16 es notablemente más volátil. Las fluctuaciones pronunciadas de una época a otra sugieren que el entrenamiento fue menos estable. A pesar de la inestabilidad, se repite el fenómeno donde la precisión de validación es consistentemente superior a la de entrenamiento, indicando una vez más el efecto positivo de las técnicas de regularización. En resumen, aunque el modelo VGG16 aprende exitosamente, su proceso de entrenamiento es menos estable

- **Curva de Pérdida (Loss):** El gráfico de pérdida confirma la inestabilidad del entrenamiento. La pérdida de validación muestra picos y valles muy marcados, aunque la tendencia general es decreciente. No se aprecian signos de un sobreajuste (*overfitting*) severo, pero la falta de estabilidad es un punto en contra en comparación con el modelo CNN personalizado.

7.1.2.2. Análisis de la Matriz de Confusión

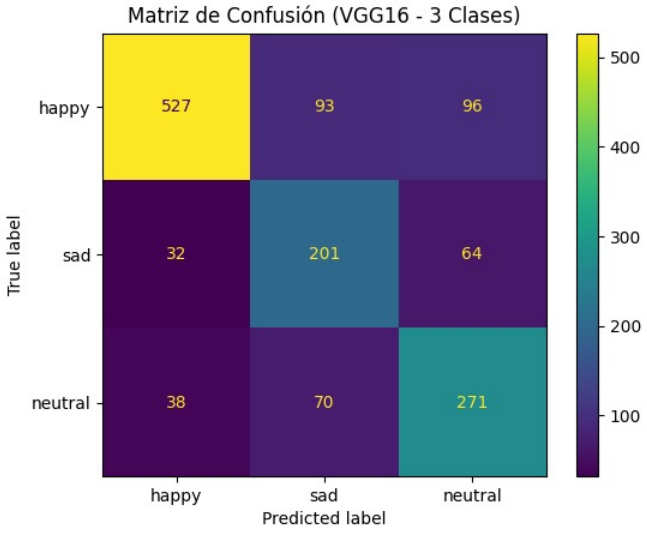


Figura 7.4: Matriz de confusión VGG16 3 clases

La matriz de confusión del modelo VGG16, presentada en la figura 7.4, muestra un patrón de aciertos y errores similar al del modelo anterior.

- **Rendimiento en la Clase 'Happy':** El modelo sigue siendo muy competente para identificar la felicidad, clasificando correctamente 527 instancias.

- **Confusión Persistente entre 'Sad' y 'Neutral':** La principal fuente de error sigue siendo la misma: la confusión entre tristeza y neutralidad. Este patrón, al ser casi idéntico al del primer modelo, refuerza la hipótesis de que esta dificultad no es exclusiva de una arquitectura, sino que es una característica inherente al desafío que presentan estas dos clases visualmente similares en el dataset.

7.1.2.3. Análisis del Reporte de Clasificación

Cuadro 7.2: Métricas por clase del modelo VGG16 (3 clases)

Clase	Precisión	Recall	F1-score	Soporte
Happy	0.88	0.74	0.80	716
Sad	0.55	0.68	0.61	297
Neutral	0.63	0.72	0.67	379
Accuracy			0.72	1392
Macro avg	0.69	0.71	0.69	1392
Weighted avg	0.74	0.72	0.72	1392

El reporte de clasificación, mostrado en el cuadro 7.2, cuantifica el rendimiento y permite una comparación directa con el modelo CNN personalizado.

- **Precisión General (Accuracy):** El modelo VGG16 logró una precisión global del **72 %**. Este resultado, aunque respetable, es **inferior al 76 %** obtenido por el modelo CNN personalizado.
- **Métricas por Clase:**
  - **Happy:** Obtiene una puntuación F1 de **0.80**. Curiosamente, su precisión (0.88) es muy alta, pero su sensibilidad o *recall* (0.74) es notablemente más baja que la del modelo CNN.
  - **Sad:** Con una puntuación F1 de **0.61**, esta sigue siendo la clase más difícil de clasificar y su rendimiento es inferior al del modelo CNN (0.64).
  - **Neutral:** Alcanza una puntuación F1 de **0.67**, también ligeramente por debajo del rendimiento del modelo CNN (0.69).

7.1.2.4. Síntesis y Comparación de los Resultados

El modelo VGG16, a pesar de su complejidad y su base pre-entrenada en ImageNet, obtuvo un rendimiento inferior al del modelo CNN personalizado, con una precisión final del **72 %** frente al **76 %**. Ambos modelos demostraron ser muy eficaces en la identificación de la clase "happy" y ambos tuvieron dificultades similares para diferenciar entre "sad" y "neutral", lo que apunta a un desafío intrínseco de los datos.

La mayor volatilidad durante el entrenamiento y el rendimiento ligeramente inferior en las métricas clave sugieren que la arquitectura más profunda de VGG16 no se tradujo en una ventaja para este problema. Es posible que una red más pequeña y diseñada a medida, como la CNN personalizada, sea más adecuada para especializarse en las características específicas de las expresiones faciales de este dataset.

7.1.3. Modelo ResNet50 (Transfer Learning)

Se concluye el análisis de los modelos para tres clases con la evaluación de la arquitectura ResNet50. Este modelo incorporó una serie de mejoras metodológicas con respecto a la implementación de VGG16, incluyendo el uso de la función de preprocesamiento específica de ResNet50, la sustitución de la capa `Flatten` por `GlobalAveragePooling2D` para reducir el número de parámetros, y la utilización de una tasa de aprendizaje más baja (`1e-4`) en el optimizador Adam. Estos cambios estaban destinados a mejorar la estabilidad y el rendimiento del entrenamiento.

7.1.3.1. Análisis de las Curvas de Aprendizaje (Precisión y Pérdida)

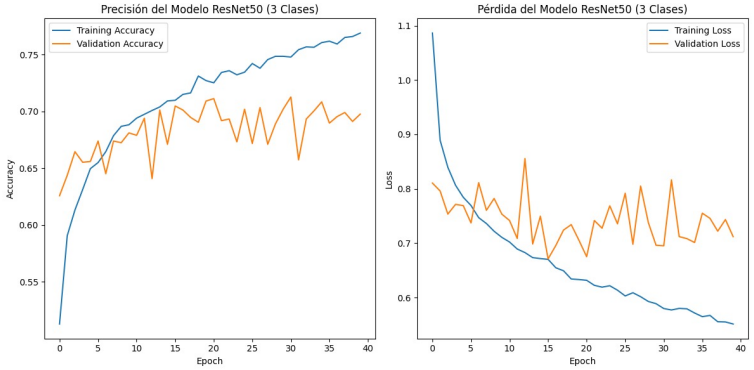


Figura 7.5: Graficas ResNet50 3 clases

Las curvas de aprendizaje del modelo ResNet50, mostradas en la figura 7.5, presentan un comportamiento significativamente diferente al de los modelos anteriores y revelan un desafío clave.

- **Curva de Precisión (Accuracy) y Pérdida (Loss):** A diferencia de los modelos anteriores, las curvas del ResNet50 muestran una clara divergencia entre el entrenamiento y la validación. La precisión de entrenamiento (*Training Accuracy*) asciende de manera constante hasta casi el 80 %, mientras que la precisión de validación (*Validation Accuracy*) se estanca y fluctúa erráticamente en torno al 70 %. De forma análoga, la pérdida de entrenamiento (*Training Loss*) desciende de forma continua y pronunciada, mientras que la pérdida de validación (*Validation Loss*) se mantiene alta y volátil.

- **Evidencia de Sobreajuste (Overfitting):** Este comportamiento es un indicio clásico de sobreajuste. El modelo está memorizando de manera muy efectiva los datos de entrenamiento, pero no logra generalizar ese aprendizaje al conjunto de datos de validación, que es nuevo para él. A pesar de las mejoras técnicas implementadas para mitigar este riesgo (como la tasa de aprendizaje reducida), la profunda complejidad de la arquitectura ResNet50 parece haber provocado que el modelo se ajuste en exceso a los datos de entrenamiento.

7.1.3.2. Análisis de la Matriz de Confusión

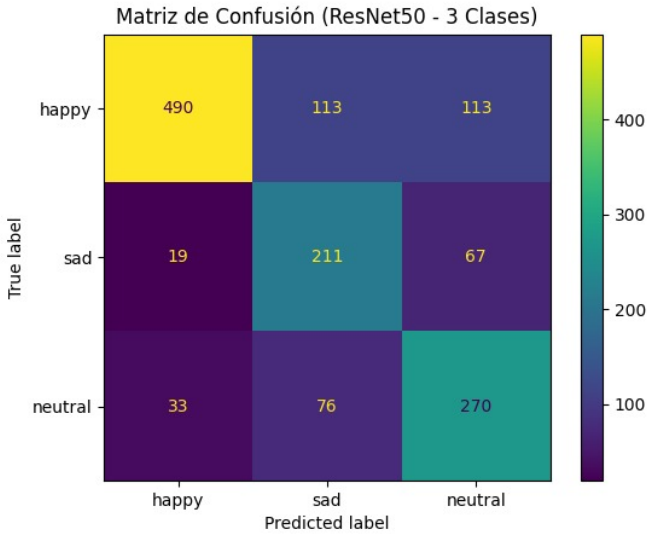


Figura 7.6: Matriz ResNet50 3 clases

La matriz de confusión, visible en la figura 7.6, refleja cómo el sobreajuste afectó el rendimiento predictivo del modelo.

- **Degradación en la Clase 'Happy':** A diferencia de los dos modelos anteriores, que sobresalían en la identificación de la felicidad, el modelo ResNet50 muestra un rendimiento notablemente inferior en esta clase. Logró solo 490 predicciones correctas y confundió un gran número de imágenes felices (113 como 'sad' y 113 como 'neutral').

- **Confusión Generalizada:** Si bien la confusión entre 'sad' y 'neutral' persiste (67 y 76 errores, respectivamente), el modelo ahora también comete errores sustanciales al clasificar la clase 'happy'. Esto sugiere que, al memorizar los datos de entrenamiento, el modelo aprendió patrones demasiado específicos que no se correspondían bien con las características generales de las expresiones en el conjunto de validación.

7.1.3.3. Análisis del Reporte de Clasificación

Cuadro 7.3: Métricas por clase del modelo ResNet50 (3 clases)

Clase	Precisión	Recall	F1-score	Soporte
Happy	0.90	0.68	0.78	716
Sad	0.53	0.71	0.61	297
Neutral	0.60	0.71	0.65	379
Accuracy			0.70	1392
Macro avg	0.68	0.70	0.68	1392
Weighted avg	0.74	0.70	0.71	1392

El reporte de clasificación final, mostrado en la cuadro 7.3, confirma que el modelo ResNet50 fue el de menor rendimiento de los tres.

- **Precisión General (Accuracy):** El modelo alcanzó una precisión global del **70 %**, situándose por debajo del **72 %** de VGG16 y del **76 %** de la CNN personalizada.
- **Métricas por Clase:**
  - **Happy:** La puntuación F1 para esta clase cayó a **0.78**, principalmente debido a una baja sensibilidad (*recall*) de **0.68**. Esto cuantifica lo observado en la matriz de confusión: el modelo falló en identificar casi un tercio de todas las imágenes de felicidad.
  - **Sad y Neutral:** Estas clases también mostraron un rendimiento inferior, con puntuaciones F1 de **0.61** y **0.65** respectivamente, las más bajas o entre las más bajas de toda la comparativa.

7.1.3.4. Síntesis y Conclusión Comparativa Final

A pesar de las mejoras técnicas implementadas en su código (preprocesamiento adecuado, `GlobalAveragePooling2D` y una tasa de aprendizaje optimizada), el modelo ResNet50 resultó ser el menos efectivo de los tres evaluados para este problema, con una precisión final del **70 %** y claros signos de sobreajuste.

La comparativa final de los tres modelos para la clasificación de 3 clases arroja una conclusión clara:

- **Mejor Modelo:** La **CNN personalizada** fue el modelo con el mejor rendimiento general (**76 %** de precisión), el entrenamiento más estable y el mejor balance en sus métricas de clasificación.

- **Modelos Pre-entrenados:** Los modelos de *transfer learning* (VGG16 y ResNet50), a pesar de su reconocida potencia en otras tareas, no lograron superar a la arquitectura más simple. Su enorme complejidad parece haber sido una desventaja, llevándolos a una mayor inestabilidad (VGG16) o a un claro sobreajuste (ResNet50), incluso con hiperparámetros ajustados.

7.1.4. Discusión entre los tres modelos (3 Clases)

El análisis cuantitativo de los resultados arroja una conclusión clara: el modelo CNN personalizado superó a las dos arquitecturas pre-entrenadas, no solo en términos de precisión final, sino también en la estabilidad y eficiencia de su proceso de aprendizaje, para ser más exactos, mostró una mejora en la precisión de 4 y 6 puntos porcentuales sobre VGG16 y ResNet50 respectivamente, lo que representa una mejora relativa en el rendimiento del 5.56 % y 8.57 %.

Para facilitar una comparación directa, la Tabla 7.4 resume las métricas de rendimiento clave obtenidas por cada modelo en el conjunto de validación.

Modelo	**Accuracy**	**F1 (Happy)**	**F1 (Sad)**	**F1 (Neutral)**
CNN Personalizada	76 %	0.84	0.64	0.69
VGG16	72 %	0.80	0.61	0.67
ResNet50	70 %	0.70	0.61	0.65

Cuadro 7.4: Tabla comparativa del rendimiento de los tres modelos para el problema de 3 clases.

Como se evidencia en la Tabla 7.4, el modelo CNN personalizado no solo alcanzó la mayor precisión general (76 %), sino que también obtuvo la puntuación F1 más alta para cada una de las tres clases individuales. Esto indica un rendimiento superior y más balanceado.

Más allá de las métricas finales, el comportamiento durante el entrenamiento reveló diferencias fundamentales:

- **Estabilidad del Entrenamiento:** El modelo CNN personalizado exhibió las curvas de aprendizaje más estables y suaves, sugiriendo un proceso de convergencia robusto. En contraste, tanto VGG16 como ResNet50 mostraron una alta volatilidad en sus curvas de validación, lo que indica inestabilidad y una mayor dificultad para generalizar de manera consistente a lo largo de las épocas.
- **Patrones de Error:** Es notable que los tres modelos compartieron la misma dificultad principal: la confusión entre las clases ‘sad‘ y ‘neutral‘. Esto sugiere que la ambigüedad visual entre estas dos expresiones es un desafío inherente a los datos mismos, más que una debilidad de una arquitectura en particular.

Los resultados sugieren que para la tarea específica de clasificar estas expresiones faciales, una arquitectura más pequeña y especializada (la CNN personalizada) fue capaz de aprender un conjunto de características más relevante y eficiente.

7.2. Resultados y Discusión de Modelos - 7 Clases

En esta segunda etapa se amplió el desafío de clasificación para incluir siete emociones: felicidad (Happy), tristeza (Sad), neutralidad (Neutral), sorpresa (Surprise), miedo (Fear), disgusto o asco (Disgust) y enojo (Anger). Los tres modelos fueron nuevamente entrenados con conjuntos balanceados mediante oversampling y aumento de datos. A continuación se presenta el análisis individual de cada modelo con base en las matrices de confusión, curvas de entrenamiento y resultados cuantitativos.

7.2.1. Modelo VGG16 (7 Clases)

Este modelo fue implementado con aprendizaje por transferencia congelando las capas del modelo base VGG16 entrenado en ImageNet. Posteriormente se añadieron capas densas adaptadas a las siete emociones del problema.

7.2.1.1. Análisis de las Curvas de Aprendizaje (Precisión y Pérdida)

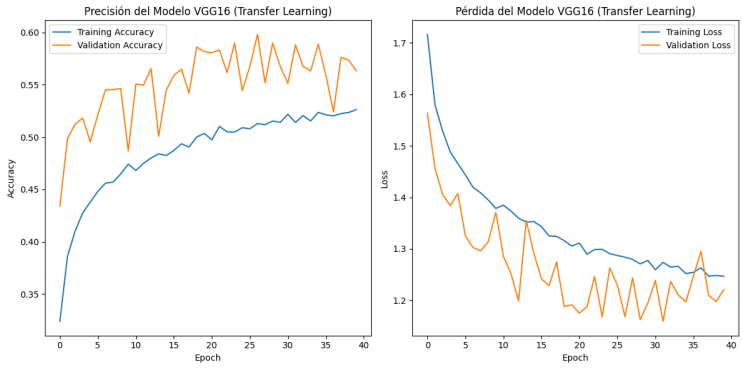


Figura 7.7: Gráficas VGG16 7 clases

- **Curva de Precisión (Accuracy):** Como se puede observar en la figura 7.7 la precisión de entrenamiento muestra una tendencia ascendente constante, alcanzando aproximadamente un 53 %. Por otro lado, la precisión de validación, aunque también con una tendencia general al alza, es extremadamente volátil. Varía de manera significativa entre épocas, con picos que llegan hasta el 60 % y valles cercanos al 50 %. A pesar de esta inestabilidad, la precisión de validación es consistentemente superior a la de entrenamiento, lo que sugiere un efecto positivo de la regularización (como *dropout*) que previene un sobreajuste severo. No obstante, la precisión final, que ronda el **56 %**, indica que el modelo tiene dificultades para generalizar en un problema con 7 clases.

- **Curva de Pérdida (Loss):** El gráfico de pérdida de la figura 7.7 refuerza la observación de

inestabilidad. La pérdida de validación presenta picos y valles muy pronunciados, aunque la tendencia general es decreciente. Esta volatilidad indica que el modelo no converge de manera estable, probablemente debido a la alta complejidad de diferenciar entre 7 emociones y a la similitud visual entre algunas de ellas. No se evidencia un sobreajuste claro, ya que la pérdida de validación no se incrementa de forma sostenida por encima de la de entrenamiento.

7.2.1.2. Análisis de la Matriz de Confusión

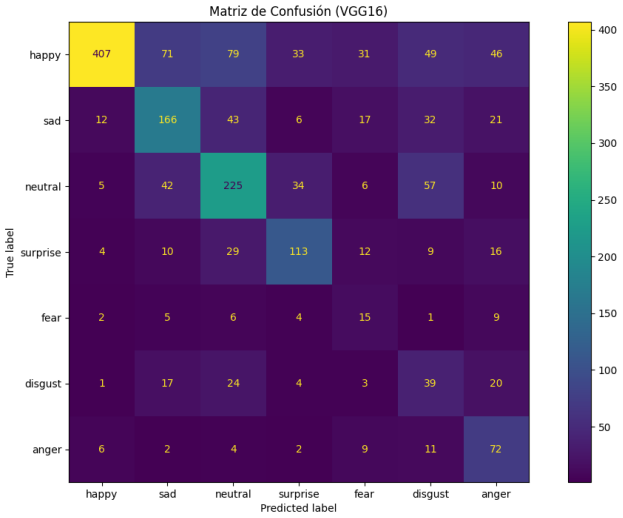


Figura 7.8: Matriz de confusión VGG16 7 clases

La matriz de confusión del modelo VGG16 para 7 clases, presentada en la figura 7.8, detalla los aciertos y, más importante aún, los patrones de error del modelo.

- **Rendimiento en Clases Clave:** La clase **'happy'** es la que mejor se identifica, con 407 predicciones correctas, aunque sigue teniendo confusiones con 'neutral' y 'sad'. La clase **'neutral'** también muestra un número considerable de aciertos (225), pero es la que más confusión genera, siendo incorrectamente predicha como 'sad' en 42 ocasiones y confundida en menor medida con casi todas las demás clases.
- **Confusiones Significativas:** Se observan varias confusiones problemáticas:
  - Una fuerte confusión entre **'sad'** y **'neutral'**.
  - La clase **'fear'** (miedo) y **'disgust'** (disgusto) son las peor clasificadas. El modelo las confunde frecuentemente con 'sad', 'neutral' y 'anger', lo que indica una dificultades para extraer las características distintivas de estas expresiones que a su vez estan relacionadas a un soporte inferior en comparación con el número de imágenes de otras clases.

- La clase **'surprise'** (sorpresa) también se confunde con 'happy' y 'neutral'.

La matriz sugiere que el modelo ha aprendido a reconocer las expresiones más pronunciadas ('happy') pero no es tan preciso con las emociones más sutiles o que comparten rasgos faciales similares, como 'sad', 'fear', y 'disgust', confundiéndolas a menudo con un estado 'neutral', este problema se relaciona con el número de imágenes con la que se valida cada clase.

7.2.1.3. Análisis del Reporte de Clasificación

Como se observa en el reporte de clasificación (cuadro 7.5), el cual cuantifica el rendimiento del modelo y confirma las debilidades observadas en la matriz de confusión.

Cuadro 7.5: Métricas por clase del modelo VGG16 (7 clases)

Clase	Precisión	Recall	F1-score	Soporte
Happy	0.93	0.57	0.71	716
Sad	0.53	0.56	0.54	297
Neutral	0.55	0.59	0.57	379
Surprise	0.58	0.59	0.58	193
Fear	0.16	0.36	0.22	42
Disgust	0.20	0.36	0.25	108
Anger	0.37	0.68	0.48	106
Accuracy			0.56	1841
Macro avg	0.47	0.53	0.48	1841
Weighted avg	0.66	0.56	0.59	1841

Métricas por Clase:

- **Happy:** Es la clase con el mejor rendimiento, con una puntuación F1 de **0.71**. Su precisión (0.93) es muy alta, lo que indica que cuando el modelo predice 'happy', es muy probable que esté en lo cierto. Sin embargo, su sensibilidad o *recall* (0.57) es considerablemente más bajo, lo que significa que no identifica un 43 % de las imágenes de felicidad.
- **Fear y Disgust:** Estas son las clases con el peor desempeño, con puntuaciones F1 de solo **0.22** y **0.25** respectivamente. Sus valores de precisión y *recall* son bajos, confirmando que el modelo tiene problemas para distinguirlas correctamente.
- **Sad, Neutral, Surprise, Anger:** Estas clases presentan un rendimiento intermedio, con puntuaciones F1 que se mueven en el rango de **0.48** a **0.58**, lo que demuestra una capacidad limitada para todas ellas.

7.2.2. Modelo ResNet50 (7 Clases)

Utilizando nuevamente transferencia de aprendizaje, ResNet50 fue aplicado con capas densas adaptadas a siete salidas. Se analizará el comportamiento del modelo durante el entrenamiento, su patrón de aciertos y errores, y sus métricas de rendimiento para obtener una visión completa de su eficacia.

7.2.2.1. Análisis de las Curvas de Aprendizaje (Precisión y Pérdida)

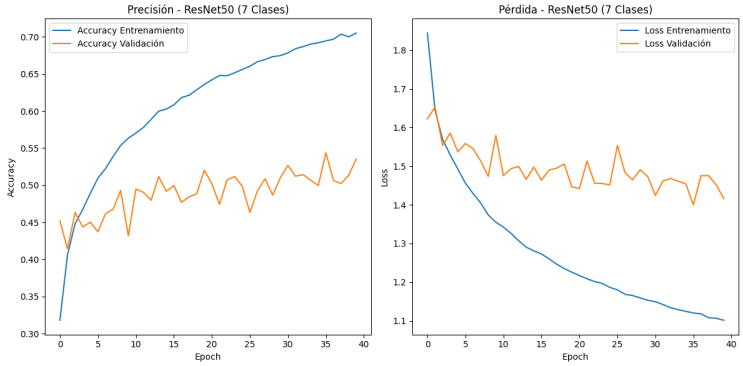


Figura 7.9: Gráficas ResNet50 7 clases

Las gráficas de entrenamiento del modelo ResNet50, mostradas en la figura 7.9, revelan un claro problema de sobreajuste (*overfitting*).

- **Curva de Precisión (Accuracy):** La precisión de entrenamiento muestra una mejora constante y fluida, superando el 70 % al final de las épocas. En contraste, la precisión de validación se estanca rápidamente alrededor del 50-55 % y exhibe una alta volatilidad. La brecha creciente entre la curva de entrenamiento y la de validación es un indicador inequívoco de que el modelo está memorizando los datos de entrenamiento pero no logra generalizar su aprendizaje a datos no vistos.

- **Curva de Pérdida (Loss):** El gráfico de pérdida confirma el diagnóstico de sobreajuste. Mientras la pérdida de entrenamiento desciende de manera consistente hasta un valor cercano a 1.1, la pérdida de validación se estanca e incluso tiende a aumentar ligeramente en las últimas épocas, manteniéndose cambiante alrededor de 1.5. Esta divergencia es la manifestación clásica del sobreajuste.

7.2.2.2. Análisis de la Matriz de Confusión

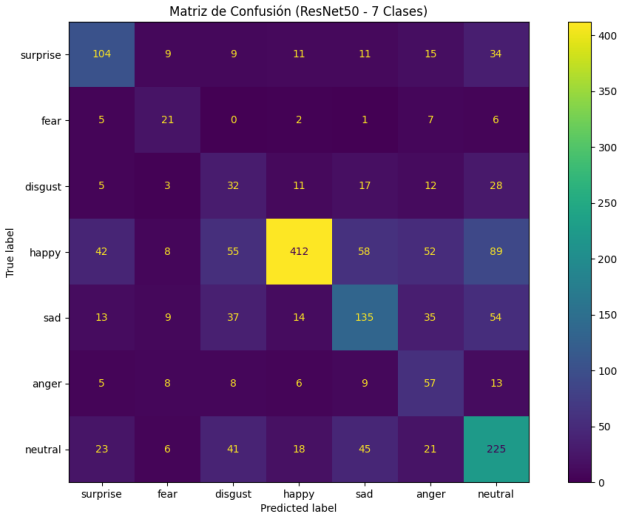


Figura 7.10: Matriz de confusión ResNet50 7 clases

La matriz de confusión del modelo ResNet50, presentada en la figura 7.10, detalla cómo se distribuyen las predicciones y qué clases son las más problemáticas.

- **Rendimiento en Clases Dominantes:** El modelo demuestra una competencia notable para identificar la clase 'happy', acertando en 412 ocasiones. De manera similar, 'neutral' y 'sad' también tienen un número de aciertos considerable (225 y 135, respectivamente). Esto sugiere que el modelo se especializa en las clases con más muestras o características más distintivas.
- **Confusiones Significativas:** Los errores del modelo siguen un patrón claro:
  - Las clases 'fear' (miedo) y 'disgust' (disgusto) son las peor clasificadas, con solo 21 y 32 aciertos respectivamente. Son confundidas masivamente con otras emociones, principalmente 'sad', 'neutral' y 'happy'
  - La clase 'happy' es una fuente común de error; muchas otras emociones como 'anger' (enojo) y 'disgust' son incorrectamente clasificadas como 'happy'.
  - La clase 'neutral', atrae una cantidad significativa de predicciones incorrectas de casi todas las demás categorías, especialmente de 'happy' (89 errores) y 'sad' (54 errores).

7.2.2.3. Análisis del Reporte de Clasificación

El reporte de clasificación cuantifica el rendimiento observado, confirmando las fortalezas y debilidades del modelo (cuadro 7.6).

Cuadro 7.6: Métricas por clase del modelo ResNet50 (7 clases)

Clase	Precisión	Recall	F1-score	Soporte
Surprise	0.53	0.54	0.53	193
Fear	0.33	0.50	0.40	42
Disgust	0.18	0.30	0.22	108
Happy	0.87	0.58	0.69	716
Sad	0.49	0.45	0.47	297
Anger	0.29	0.54	0.37	106
Neutral	0.50	0.59	0.54	379
Accuracy			0.54	1841
Macro avg	0.45	0.50	0.46	1841
Weighted avg	0.61	0.54	0.56	1841

Métricas por Clase:

- **Precisión General:** El modelo ResNet50 alcanzó una precisión global del 54 %. Este resultado es modesto y ligeramente inferior al obtenido por VGG16 (56 %), lo que indica que su mayor complejidad no se tradujo en un mejor rendimiento.

- **Happy:** Es la clase con mejor rendimiento, con una puntuación F1 de 0.69. Su alta precisión (0.87) indica que sus predicciones de 'happy' son fiables, pero su bajo *recall* (0.58) muestra que ignora más del 40 % de las imágenes felices.

- **Anger y Disgust:** Estas clases tienen el peor desempeño, con puntuaciones F1 de 0.37 y 0.22 respectivamente. La precisión de 'disgust' (0.18) es extremadamente baja, lo que significa que la mayoría de sus predicciones para esta clase son incorrectas.

- **Fear:** También muestra un rendimiento muy bajo, con una puntuación F1 de 0.40

- **Surprise, Sad, Neutral:** Presentan un rendimiento intermedio, con puntuaciones F1 de 0.53, 0.47 y 0.54.

7.2.3. Modelo CNN Personalizada (7 Clases)

Este modelo fue diseñado y entrenado con técnicas de aumento de datos. Su arquitectura incluye múltiples capas convolucionales, pooling y una capa densa final con activación softmax. (se porfundiza más en la sección 6.2.3.2)

7.2.3.1. Análisis de las Curvas de Aprendizaje (Precisión y Pérdida)

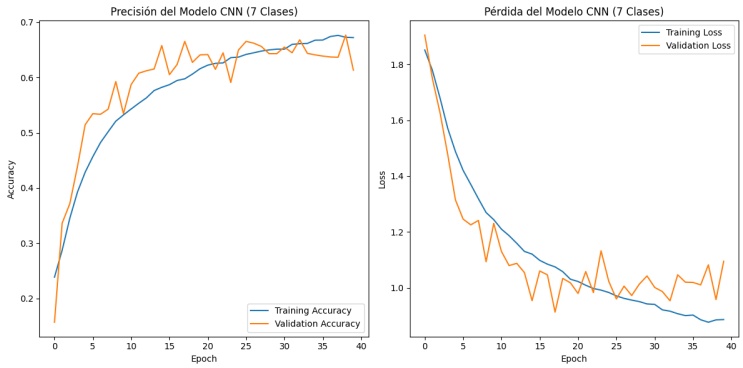


Figura 7.11: Gráficas CNN 7 clases

Tal como se puede apreciar en la figura 7.11, las gráficas de aprendizaje para la CNN personalizada son notablemente diferentes a las de los 2 anteriores modelos pre-entrenados y sugieren un comportamiento de entrenamiento mucho más efectivo.

- **Curva de Precisión (Accuracy):** Se observa un comportamiento ejemplar. Tanto la precisión de entrenamiento como la de validación, siguen una trayectoria ascendente muy similar, finalizando ambas en un rango cercano al 64-68 %. Aunque la curva de validación muestra cierta inestabilidad, no existe la brecha amplia y sostenida que se vio en los otros modelos. Esto es un fuerte indicio de que el modelo evitó el sobreajuste y logró generalizar bien.

- **Curva de Pérdida (Loss):** El gráfico de pérdida confirma lo anteriormente explicado. Ambas curvas, la de entrenamiento y la de validación, descienden a lo largo de las épocas. La ausencia de una divergencia clara entre las dos curvas refuerza la idea de que el modelo está aprendiendo patrones útiles de los datos en lugar de simplemente memorizarlos.

7.2.3.2. Análisis de la Matriz de Confusión

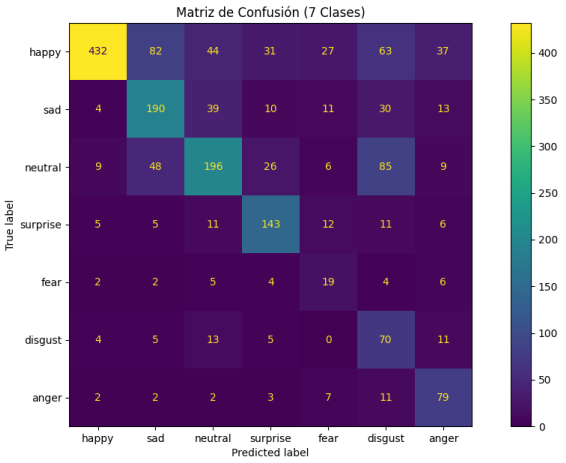


Figura 7.12: Matriz de confusión CNN 7 clases

- **Rendimiento en Clases Dominantes:** Tal como se observa en la figura 7.12 la diagonal principal es notablemente más fuerte y distribuida que en los casos anteriores. Clases como 'happy' (432), 'sad' (190), 'neutral' (196) y 'surprise' (143) son identificadas con una alta tasa de acierto.

- **Confusiones Significativas:**

  - La debilidad más grande del modelo es la clase 'fear' (miedo), que solo se identifica correctamente en 19 ocasiones. La confusión para esta clase se distribuye entre varias otras emociones, indicando que sus características distintivas no fueron aprendidas.
  - Existe una confusión considerable entre las clases 'neutral' y 'disgust', donde 85 imágenes neutrales fueron clasificadas incorrectamente como disgusto.
  - La clase 'happy' también es confundida con 'sad' (82 veces), lo que podría indicar problemas con imágenes de sonrisas sutiles o ambiguas.

7.2.3.3. Análisis del Reporte de Clasificación

Se puede evidenciar que en el reporte de métricas (cuadro 7.7) se valida cuantitativamente el éxito de la estrategia de entrenamiento, mostrando un rendimiento superior y más equilibrado.

Cuadro 7.7: Métricas por clase del modelo CNN (7 clases)

Clase	Precisión	Recall	F1-score	Soporte
Happy	0.95	0.64	0.77	716
Sad	0.52	0.70	0.60	297
Neutral	0.61	0.58	0.60	379
Surprise	0.69	0.74	0.71	193
Fear	0.30	0.40	0.34	42
Disgust	0.35	0.47	0.40	108
Anger	0.46	0.82	0.59	106
Accuracy			0.64	1841
Macro avg	0.55	0.62	0.57	1841
Weighted avg	0.71	0.64	0.66	1841

Métricas por Clase:

- **Precisión General:** El modelo CNN personalizado alcanzó una precisión global del 64 %. Este es el mejor resultado entre los tres modelos probados para la tarea de 7 clases, superando el 56 % de VGG16 y el 54 % de ResNet50. El 64 % de precisión se logra sobre el conjunto de validación (`X_val`), el cual, como se realizó en el código, fue separado antes del balanceo y se mantuvo intacto.
- **Happy y Surprise:** Con F1-scores de 0.77 y 0.71, respectivamente y una precisión de 0.95 para 'happy', el modelo es muy fiable cuando predice estas emociones.
- **Sad, Neutral y Anger:** Muestran un rendimiento sólido y muy competente, con puntuaciones F1 de 0.60, 0.60 y 0.59 respectivamente. Es destacable el alto *recall* (0.82) de la clase 'anger', lo que significa que el modelo es muy bueno para encontrar esta emoción.
- **Fear y Disgust:** Siguen siendo las clases más débiles, con puntuaciones F1 de 0.34 y 0.40 respectivamente. A pesar de ser bajas, estas métricas son superiores a las obtenidas por los otros modelos para estas mismas clases.

7.2.4. Discusión entre los tres modelos (7 Clases)

Al escalar el problema a la clasificación de siete emociones, la complejidad de la tarea aumentó significativamente, lo que puso a prueba la capacidad de generalización y la robustez de cada una de las tres arquitecturas implementadas. A pesar del incremento en la dificultad, el análisis comparativo de los resultados revela una conclusión consistente con la observada en el problema de 3 clases: el modelo CNN personalizado demostró ser superior a las arquitecturas de transfer learning, tanto en métricas cuantitativas como en comportamiento cualitativo durante el entrenamiento.

Para ser más exactos, el modelo CNN personalizado mostró una mejora en la precisión de 8 y 10 puntos porcentuales sobre VGG16 y ResNet50 respectivamente, lo que representa una mejora relativa en el rendimiento del 14.3 % y 18.5 %.

Para facilitar una comparación directa, la Tabla 7.8 resume las métricas de rendimiento clave obtenidas por cada modelo en el conjunto de validación para el problema de 7 clases.

Modelo	**Accuracy**	**F1 (Happy)**	**F1 (Sad)**	**F1 (Neutral)**
CNN Personalizada	**64 %**	**0.77**	**0.60**	**0.60**
VGG16	56 %	0.71	0.54	0.57
ResNet50	54 %	0.69	0.47	0.54

Modelo	**F1 (Surprise)**	**F1 (Fear)**	**F1 (Disgust)**	**F1 (Anger)**
CNN Personalizada	**0.71**	**0.34**	**0.40**	**0.59**
VGG16	0.58	0.22	0.25	0.48
ResNet50	0.53	0.40	0.22	0.37

Cuadro 7.8: Tabla comparativa del rendimiento. La tabla se ha dividido en dos partes para facilitar su visualización.

Como se evidencia en la Tabla 7.8, el modelo CNN personalizado no solo alcanzó la mayor precisión general (64 %), sino que también obtuvo, en su mayoría, las puntuaciones F1 más altas en las clases individuales. Esto indica que no solo fue el modelo más preciso, sino también el más equilibrado.

Más allá de las métricas finales, el comportamiento durante el entrenamiento reveló diferencias fundamentales que explican estos resultados:

- **Estabilidad y Sobreajuste:** El modelo CNN personalizado fue el único que demostró un entrenamiento estable y una buena generalización. Sus curvas de precisión y pérdida de entrenamiento y validación se mantuvieron cercanas, evitando el sobreajuste. En contraste, tanto VGG16 como, en mayor medida, ResNet50, mostraron claros signos de sobreajuste. Sus curvas de aprendizaje divergieron significativamente, indicando que estaban memorizando los datos de entrenamiento pero fallando en generalizar a los datos no vistos, un problema que se acentuó con la mayor complejidad de las 7 clases.
- **Patrones de Error:** Si bien los tres modelos tuvieron dificultades con las clases con menor soporte como ‘fear‘ y ‘disgust‘, el modelo CNN personalizado logró un rendimiento superior incluso en estas categorías. Los modelos pre-entrenados tendieron a confundir una gama más amplia de emociones con clases dominantes como ‘happy‘ o ‘neutral‘, mientras que la CNN personalizada, aunque no perfecta, mostró una distribución de errores ligeramente más contenida.

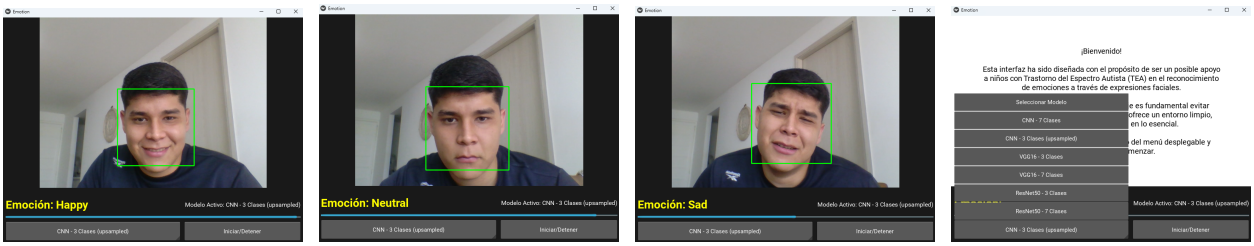
7.3. Interfaz funcional: Reconocimiento de emociones en tiempo real

En esta sección se presentan los resultados cualitativos de la aplicación desarrollada, poniendo a prueba los tres modelos entrenados para el problema de 3 clases. Tal como se estableció en la sección de consideraciones bioéticas, las pruebas se realizaron con los autores del proyecto con el único fin de demostrar la funcionalidad de la herramienta y obtener evidencia tangible de su operatividad.

La interfaz permite seleccionar dinámicamente cualquiera de los modelos entrenados a través de un menú desplegable (tal como se profundiza en la sección 6.2.3.4 y se muestra en las figuras 6.17 y 6.18 de dicha sección). La Figura 7.13d muestra el entorno inicial de la aplicación, donde se ha seleccionado el modelo CNN personalizado antes de iniciar la detección.

7.3.1. Pruebas con Modelos de 3 Clases

Modelo CNN Personalizada La Figura 7.13 muestra la ejecución del modelo CNN personalizado. El rendimiento en tiempo real fue notablemente robusto y consistente con los resultados cuantitativos. Las tres emociones (‘happy‘, ‘sad‘ y ‘neutral‘) fueron reconocidas de manera fluida y precisa, sin necesidad de exagerar o forzar las expresiones faciales. Este comportamiento práctico es un reflejo directo de su superioridad en las métricas, habiendo alcanzado la precisión más alta (76 %) y las puntuaciones F1 más balanceadas (0.84 para ‘happy‘, 0.64 para ‘sad‘ y 0.69 para ‘neutral‘). La estabilidad de su entrenamiento, sin signos de sobreajuste, se tradujo en un modelo fiable y generalizable en un entorno de uso real.

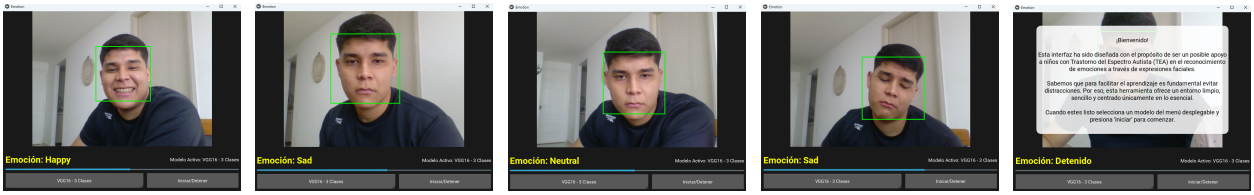


(a) Happy (b) Neutral (c) Sad (d) Selección

Figura 7.13: Resultados de la detección en tiempo real con el modelo CNN Personalizado (3 clases).

Modelo VGG16 La prueba del modelo VGG16 (Figura 7.14) a parte de evidenciar que se puede cambiar de modelo en medio de un reconocimiento, tal como se ve en la figura 7.14e, evidenció en la práctica las debilidades identificadas en el análisis cuantitativo. Mientras que las expresiones ‘happy‘ y ‘sad‘ fueron reconocidas correctamente, el modelo mostró una clara dificultad con la expresión ‘neutral‘, confundiéndola consistentemente con ‘sad‘ (Figura 7.14b). Este error es coherente con la confusión persistente entre ’sad’ y ’neutral’ observada en su matriz de confusión. Adicionalmente, se notó que la barra de confianza del modelo raramente alcanzaba el 100 %, oscilando en torno al

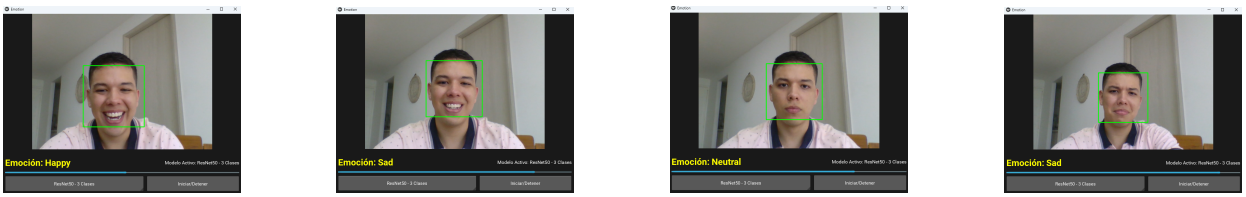
50-60 %, lo que indica una menor certeza en sus predicciones, en línea con sus métricas F1 inferiores (0.61 para 'sad' y 0.67 para 'neutral').



(a) Happy (b) Neutral->Sad (c) Neutral (d) Sad (e) Selección

Figura 7.14: Resultados de la detección en tiempo real con el modelo VGG16 (3 clases).

Modelo ResNet50 El modelo ResNet50 (Figura 7.15) fue el que presentó mayores desafíos durante la prueba funcional, validando los resultados de su análisis. Las expresiones 'neutral' y 'sad' fueron reconocidas con una eficacia razonable. Sin embargo, el modelo tuvo una marcada dificultad para identificar la emoción 'happy', tendiendo a clasificarla incorrectamente como 'sad' (Figura 7.15b). Para lograr una detección correcta, fue necesario forzar o exagerar notablemente la sonrisa (Figura 7.15a). Este comportamiento se alinea perfectamente con las métricas obtenidas, donde la clase 'happy' tuvo una baja sensibilidad (*recall*) de 0.68, y con el análisis de la matriz de confusión, en la cual se evidenció una confusión generalizada en esta clase (acertó en 490 imagenes pero falló en 226).



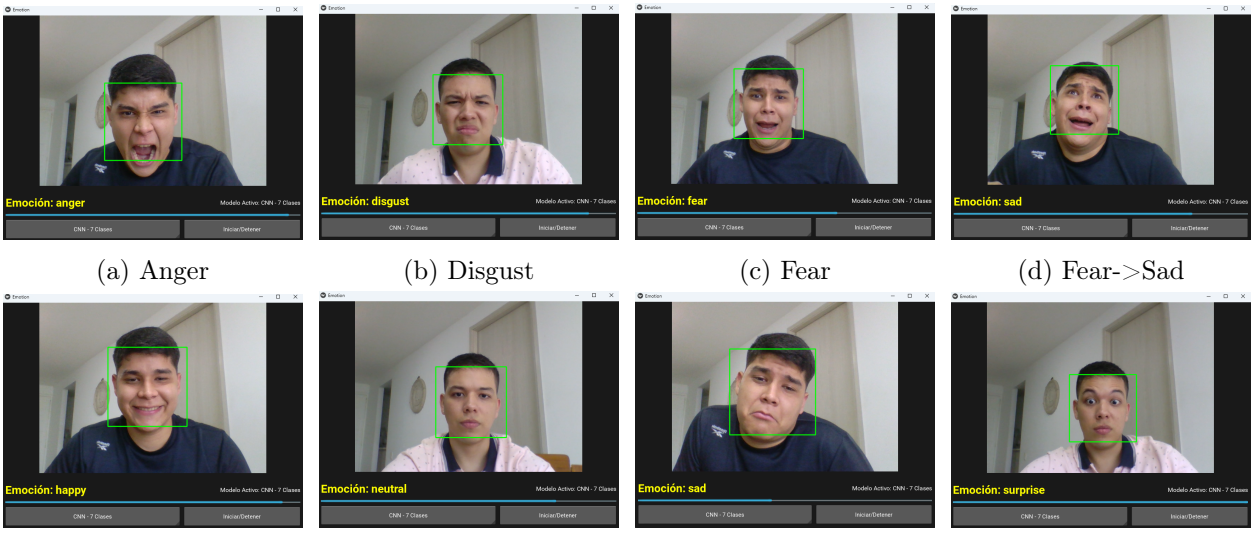
(a) Happy (Forzado) (b) Happy->Sad (c) Neutral (d) Sad

Figura 7.15: Resultados de la detección en tiempo real con el modelo ResNet50 (3 clases).

7.3.2. Pruebas con Modelos de 7 Clases

Al escalar el problema a siete emociones, la complejidad para los modelos aumenta considerablemente. En esta fase de pruebas funcionales, se evaluó cómo cada arquitectura manejaba la tarea de distinguir entre un rango más amplio y sutil de expresiones faciales (son 7 posibles emociones). La interfaz permitió cambiar dinámicamente entre los modelos de 7 clases para observar su rendimiento en tiempo real bajo las mismas condiciones.

Modelo CNN Personalizada El modelo CNN personalizado, que demostró ser el más robusto en el análisis cuantitativo, mantuvo un rendimiento superior en la prueba funcional (Figura 7.16). La mayoría de las emociones fueron identificadas de manera correcta y con una alta confianza, reflejada en la barra de progreso. El único inconveniente notable se presentó con la emoción 'fear', la cual el modelo tendió a confundir con 'sad' (Figura 7.16d). Este error práctico es totalmente coherente con las métricas obtenidas, donde 'fear' fue la clase con la puntuación F1 más baja (0.34), y con la matriz de confusión, que mostró que su principal fuente de error era, en efecto, la clasificación incorrecta como 'sad'. A pesar de esta debilidad específica, su capacidad para distinguir correctamente las otras seis clases valida su estatus como el modelo más equilibrado y efectivo.

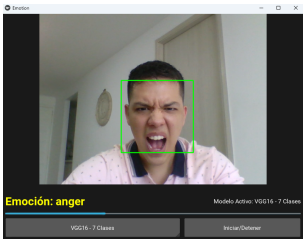


(a) Anger (b) Disgust (c) Fear (d) Fear->Sad

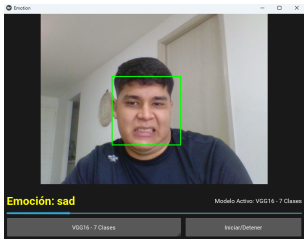
(e) Happy (f) Neutral (g) Sad (h) Surprise

Figura 7.16: Resultados de la detección en tiempo real con el modelo CNN Personalizado (7 clases).

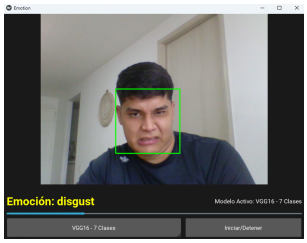
Modelo VGG16 La ejecución del modelo VGG16 (Figura 7.17) reflejó la inestabilidad y el rendimiento modesto observados en su análisis. El modelo presentó confusiones significativas con las clases 'fear' y 'disgust', las cuales tendía a clasificar erróneamente como 'sad' (Figuras 7.17d y 7.17b). Esto se corresponde directamente con los resultados cuantitativos, que mostraron puntuaciones F1 extremadamente bajas para 'fear' (0.22) y 'disgust' (0.25). Adicionalmente, se observó que la confianza del modelo era inconsistente: mientras que para 'anger', 'sad' y 'surprise' la barra de certeza era relativamente alta, para 'happy' y 'neutral' esta apenas alcanzaba el 30-40 %, indicando una baja confianza incluso en predicciones correctas.



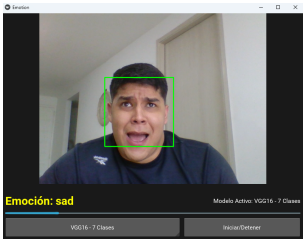
(a) Anger



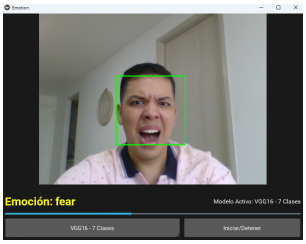
(b) Disgust->Sad



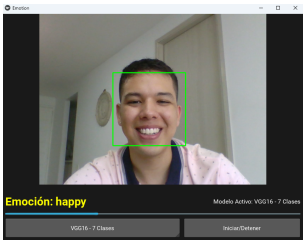
(c) Disgust



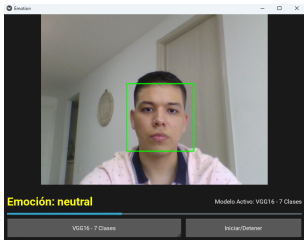
(d) Fear->Sad



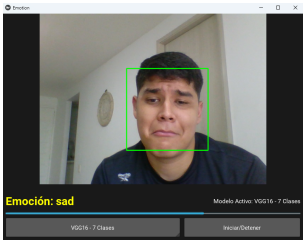
(e) Fear



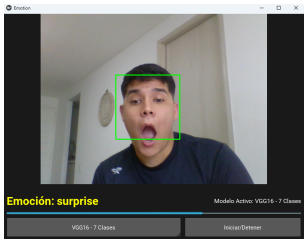
(f) Happy



(g) Neutral



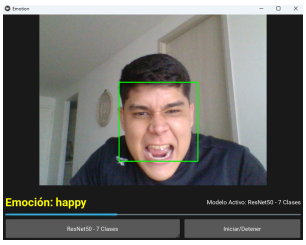
(h) Sad



(i) Surprise

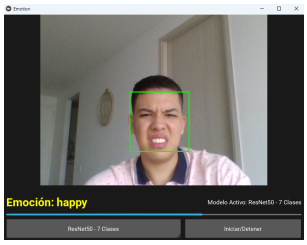
Figura 7.17: Resultados de la detección en tiempo real con el modelo VGG16 (7 clases).

Modelo ResNet50 El modelo ResNet50, que ya había mostrado signos de sobreajuste severo en su análisis, confirmó ser el menos fiable en la prueba práctica (Figura 7.18). Presentó confusiones generalizadas: ‘anger‘ y ‘disgust‘ fueron incorrectamente clasificadas como ‘happy‘ (Figuras 7.18a y 7.18c), y ‘fear‘ fue clasificada como ‘anger‘. Este comportamiento es una manifestación directa de su bajo rendimiento en las métricas, con puntuaciones F1 de solo 0.37 (‘anger‘) y 0.22 (‘disgust‘). Las únicas emociones que el modelo reconoció consistentemente fueron ‘happy‘, ‘neutral‘, ‘sad‘ y ‘surprise‘. No obstante, incluso en estas predicciones correctas, la confianza del modelo era baja (en torno al 40-50 %), con la excepción de ‘surprise‘, que sí mostró una certeza alta. Esto confirma que el sobreajuste impidió al modelo aprender características generalizables para la mayoría de las emociones.

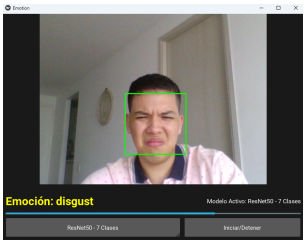


(a) Anger->Happy

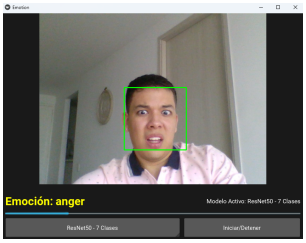
(b) Anger



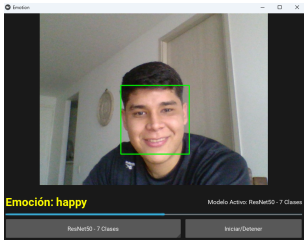
(c) Disgust->Happy



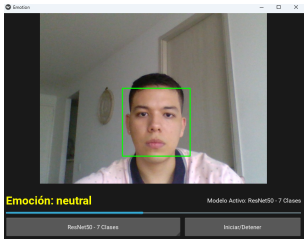
(d) Disgust



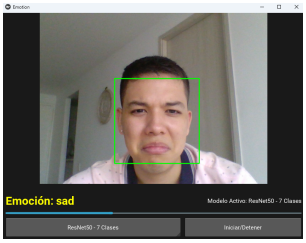
(e) Fear->Anger



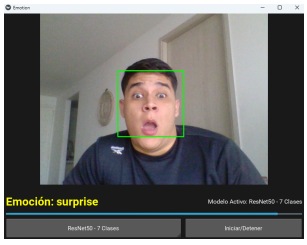
(f) Happy



(g) Neutral



(h) Sad



(i) Surprise

Figura 7.18: Resultados de la detección en tiempo real con el modelo ResNet50 (7 clases).

Capítulo 8

Conclusiones

8.1. Conclusiones

Este trabajo de grado se propuso desarrollar y evaluar modelos para el reconocimiento de expresiones faciales para determinar emociones en tiempo real, con el fin de establecer una base tecnológica para una posible herramienta de asistencia para niños con Trastorno del Espectro Autista (TEA). Tras una rigurosa fase de pruebas y desarrollo, se concluye que los objetivos planteados se cumplieron satisfactoriamente, validando la viabilidad del enfoque propuesto.

La conclusión central extraída de la evaluación comparativa es que, para este problema específico, una **Red Neuronal Convolucional (CNN) diseñada a medida superó consistentemente** a las arquitecturas de *transfer learning* más complejas como **VGG16** y **ResNet50**. En la tarea de clasificación de 7 clases, el modelo personalizado alcanzó una precisión del **64 %**, mostrando una mejora de 8 y 10 puntos porcentuales sobre VGG16 (56 %) y ResNet50 (54 %) respectivamente. El análisis de las curvas de aprendizaje demostró que, mientras los modelos pre-entrenados mostraron una marcada tendencia al sobreajuste y a la inestabilidad, la CNN personalizada logró una mejor generalización.

Se validó la aplicabilidad de los algoritmos mediante el desarrollo de una interfaz que realizara la detección en tiempo real. La interfaz de esta aplicación, cuyo acceso y uso se detallan en la metodología, permite la carga dinámica de cualquiera de los modelos entrenados para su prueba inmediata. Su rendimiento fue probado exitosamente por los autores en condiciones de alta luminosidad, demostrando que los algoritmos son eficientes y capaces de operar fuera de un entorno de laboratorio. Si bien se observaron algunas confusiones, especialmente en las clases con menor soporte en el dataset, la funcionalidad principal de identificar emociones en tiempo real se cumplió, sentando las bases para futuras mejoras.

En respuesta directa a los requerimientos de la población objetivo (niños con TEA), el diseño de la interfaz se centró deliberadamente en el minimalismo y la eliminación de estímulos sensoriales innecesarios. Lejos de ser una simple ventana con texto, esta decisión de diseño, fundamentada en la evidencia sobre la hipersensibilidad sensorial en personas con TEA, es una característica central del proyecto. Se priorizó la claridad y la funcionalidad esencial sobre cualquier elemento estético que pudiera generar una sobrecarga sensorial, asegurando que la herramienta fuera conceptualmente aplicable al contexto planteado.

Finalmente, este proyecto establece una base sólida para futuras investigaciones. La metodología, documentada con diagramas de flujo y fragmentos de código, garantiza la replicabilidad del trabajo. Las futuras líneas de acción podrían enfocarse en la ampliación del conjunto de datos para mejorar el rendimiento en las clases minoritarias y explorar el ajuste fino (*fine-tuning*) de modelos complejos con técnicas de regularización más avanzadas.

En síntesis, este trabajo de grado y los resultados mostrados demuestra la viabilidad de un sistema de asistencia emocional y, crucialmente, establece que para problemas de nicho como este, una solución especializada puede ser más efectiva que arquitecturas de propósito general.

Capítulo 9

Trabajos futuros

De cara al futuro, este trabajo plantea diversas oportunidades de expansión y profundización. Una de las más atractivas es la posibilidad de implementar el sistema de reconocimiento de emociones en un dispositivo portátil, como unas gafas inteligentes. Este avance permitiría una interacción más directa, intuitiva y discreta, brindando a las personas con TEA una herramienta que reconozca emociones en tiempo real mientras interactúan con su entorno cotidiano.

Adicionalmente, se propone enriquecer el modelo con información proveniente del lenguaje verbal y no verbal, integrando señales como el tono de voz, pausas, gestos y posturas corporales. Esta retroalimentación multimodal permitiría alcanzar un nivel de comprensión emocional más cercano al comportamiento humano, abriendo la puerta al reconocimiento de matices como la ironía, la exageración emocional o la burla, aspectos fundamentales para una interpretación más realista y empática.

Otra línea de trabajo relevante consiste en aumentar la diversidad y calidad de las bases de datos utilizadas para el entrenamiento, incluyendo aún más rostros de distintas edades, etnias y condiciones particulares, lo que fortalecería la generalización del modelo. Así mismo, explorar modelos más robustos y específicos como Transformers visuales o redes neuronales adaptativas podría llevar el sistema a un nuevo nivel de precisión y eficiencia.

Finalmente, se considera importante desarrollar módulos de personalización dentro de la aplicación funcional, permitiendo que se ajuste a las necesidades individuales del usuario. Esto podría incluir configuraciones adaptativas, niveles de sensibilidad, y la integración de sistemas de respuesta que brinden retroalimentación amigable, instrucciones o incluso juegos interactivos que fomenten el aprendizaje emocional.

Estas proyecciones no solo enriquecerían la aplicación tecnológica, sino que contribuirían a construir herramientas más inclusivas, humanas y eficaces para la comunidad TEA y otros grupos que requieran apoyo en la interpretación de emociones faciales.

Bibliografía

[1] S. B. Mukherjee, "Autism spectrum disorders - diagnosis and management," *Indian Journal of Pediatrics*, vol. 84, no. 5, pp. 393–398, 2017. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28101829/

[2] G. P.-C. M. I. F.-A. Inmaculada March-Miguez, Maite Montagut-Asunción, "Intervención en habilidades sociales de los niños con trastorno de espectro autista: Una revisión bibliográfica," *Papeles del Psicólogo*, 2018. [Online]. Available: https://www.redalyc.org/journal/778/77855949009/html/

[3] M. G. O. M. Gustavo Celis Alcala, "Trastorno del espectro autista (tea)," *Revista de la Facultad de Medicina (México)*, vol. 65, no. 1, pp. 7–20, 2022. [Online]. Available: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0026-17422022000100007&lng=es&nrm=iso

[4] F. I. for Integrated Circuits IIS. (2019) Erik - anforderungsanalyse: Entwicklung einer roboterplattform zur unterstützung neuer interaktionsstrategien in der therapie von kindern mit eingeschränkten sozio-emotionalen fähigkeiten. [Online]. Available: https://www.scs.fraunhofer.de/de/publikationen/studien/studie-erik-anforderungsanalyse.html

[5] M. K.-D. S. M. Ariane V. S. Buescher, Zuleyha Cidav, "Costs of autism spectrum disorders in the united kingdom and the united states," *JAMA Pediatrics*, vol. 168, no. 8, pp. 721–728, 2014. [Online]. Available: https://jamanetwork.com/journals/jamapediatrics/fullarticle/1879723

[6] W. H. Organization, "Medidas integrales y coordinadas para gestionar los trastornos del espectro autista: Informe de la secretaría," *67.ª Asamblea Mundial de la Salud*, vol. A67/17, 2014, punto 13.4 del orden del día provisional. [Online]. Available: https://www.who.int

[7] F. D. C. A. B. E. S. A. R. M. C. A. Frolli, G. Savarese and M. C. Ricci, "Children on the autism spectrum and the use of virtual reality for supporting social skills," *Children*, vol. 9, p. 181, 2022. [Online]. Available: https://doi.org/10.3390/children9020181

[8] Z. K. S. El-Salahi and R. Vohora, "Experiences of inclusive school settings for children and young people on the autism spectrum in the uk: a systematic review," *Journal Title*, vol. X, no. X, pp. X–X, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s40489-023-00405-2

[9] G. Paraskevi, "The inclusion of children with autism in the mainstream school classroom. knowledge and perceptions of teachers and special education teachers," *Journal Title*, vol. X, no. X, pp. X–X, Year. [Online]. Available: https://www.scirp.org/journal/paperinformation?paperid=111978

[10] J. Murray, "Practical teaching strategies for students with autism spectrum disorder: A review of the literature," *Journal Title*, vol. X, no. X, pp. X–X, Year. [Online]. Available: https://files.eric.ed.gov/fulltext/EJ1230708.pdf

[11] L. Petersson-Bloom and M. Holmqvist, "Strategies in supporting inclusive education for autistic students—a systematic review of qualitative research results," *Autism Dev Lang Impair.*, vol. 7, p. 23969415221123429, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9620685/

[12] A. P. et al., "Teaching children with autism spectrum disorder desire-based emotion prediction and cause," *J Autism Dev Disord.*, vol. 16, no. 3, pp. 826–836, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10480360/

[13] J. A. W. M. C. Cappadocia and D. Pepler, "Bullying experiences among children and youth with autism spectrum disorders," *J Autism Dev Disord.*, vol. 42, no. 2, pp. 266–277, 2012. [Online]. Available: https://doi.org/10.1007/s10803-011-1241-x

[14] R. A. K. D. L. Forrest and S. Stroope, "Autism spectrum disorder symptoms and bullying victimization among children with autism in the united states," *J Autism Dev Disord.*, vol. 50, no. 2, pp. 560–571, 2020. [Online]. Available: https://doi.org/10.1007/s10803-019-04282-9

[15] S. K. X. H. B. L. N. Xiao, K. Shinwari and J. Qi, "Effects of equine-assisted activities and therapies for individuals with autism spectrum disorder: Systematic review and meta-analysis," *Int. J. Environ. Res. Public Health*, vol. 20, no. 3, p. 2630, 2023. [Online]. Available: https://doi.org/10.3390/ijerph20032630

[16] A. Kershaw. (2023) Horseback riding as autism therapy. [Online]. Available: https://www.autismspeaks.org/expert-opinion/equine-therapy-autism#:~:text=Horses%20calm%20riders%20with%20autism,and%20gently%20discourage%20negative%20behaviors

[17] J. E. Spector and N. N. Singh, "Challenges and issues for autistic children in schools," *Verywell Health*, 2015.

[18] L. J. Rudy, "Why is school so challenging for autistic children? 8 reasons most people aren't aware of," 2023, updated on Nov. 13, 2023.

[19] J. Murray, "Practical teaching strategies for students with autism spectrum disorder: A review of the literature," *BU Journal of Graduate Studies in Education*, vol. 7, no. 2, p. 71, 2015.

[20] L. M. Little, "Peer victimization of children with asperger syndrome and high-functioning autism: Problems and solutions," *The American Journal of Orthopsychiatry*, vol. 72, no. 1, p. 57–68, 2002.

[21] K. C. P. H. K. K. Y. I. Hwang, C. L. Ball and P. A. Fisher, "Autism spectrum disorder symptoms and bullying victimization among children with autism in the united states," *Journal of Autism and Developmental Disorders*, vol. 50, no. 2, p. 560–571, 2019.

[22] Cureus, "Correlations between the development of social anxiety and individuals with autism spectrum disorder: A systematic review," *Cureus*, vol. 15, no. 9, p. e44841, 2023.

[23] U. Rao and L. A. Chen, "Characteristics, correlates, and outcomes of childhood and adolescent depressive disorders," *Dialogues Clin Neurosci*, vol. 11, no. 1, p. 45–62, 2009.

[24] E. C. M. Tandon and J. Luby, "Internalizing disorders in early childhood: a review of depressive and anxiety disorders," *Child Adolesc Psychiatr Clin N Am*, vol. 18, no. 3, p. 593–610, 2009.

[25] J. E. Lainhart and S. E. Folstein, "Affective disorders in people with autism: a review of published cases," *J Autism Dev Disord*, vol. 24, no. 5, p. 587–601, 1994.

[26] S. B. e. a. O. T. Leyfer, S. E. Folstein, "Comorbid psychiatric disorders in children with autism: interview development and rates of disorders," *J Autism Dev Disord*, vol. 36, no. 7, p. 849–861, 2006.

[27] A. M. S. R. Skinner, C. Ng and T. Walters, "A patient with autism and severe depression: medical and ethical challenges for an adolescent medicine unit," *Med J Aust*, vol. 183, no. 8, p. 422–424, 2005.

[28] N. M. Rubio, "Paul ekman: biografía y aportes de este estudioso de las emociones," 2020. [Online]. Available: https://psicologiaymente.com/biografias/paul-ekman

[29] P. Ekman, "An argument for basic emotions. cognition and emotion," vol. 6(3–4),, p. 169–200., 2008. [Online]. Available: https://doi.org/10.1080/02699939208411068

[30] M. Wrobel and M. Piórkowska, "Basic emotions," 2017. [Online]. Available: https://www.researchgate.net/publication/318447136_Basic_Emotions

[31] Wikipedia, "Paul ekman," 2025. [Online]. Available: https://es.wikipedia.org/w/index.php?title=Paul_Ekman&oldid=166436208

[32] L. F. Barrett, "Are emotions natural kinds? perspectives on psychological science," *Sage Journals*, vol. 1, pp. 28–58, 2006. [Online]. Available: https://doi.org/10.1111/j.1745-6916.2006.00003.x

[33] H. A. Uljarevic, M., "Recognition of emotions in autism: A formal meta-analysis. j autism dev disord," *Journal of Autism and Developmental Disorders*, vol. 43, p. 1517–1526, 2012. [Online]. Available: https://doi.org/10.1007/s10803-012-1695-5

[34] M. A. Lozier LM, Vanmeter JW, "Impairments in facial affect recognition associated with autism spectrum disorders: A meta-analysis. development and psychopathology." *Cambridge University Press*, vol. 26, pp. 933–945, 2014. [Online]. Available: https://doi.org/10.1017/S0954579414000479

[35] Y. G. S. M. K. D. V. L. Ofer Golan, Emma Ashwin and S. Baron-Cohen, "Enhancing emotion recognition in children with autism spectrum conditions: An intervention using animated vehicles with real emotional faces. j autism dev disord," *Journal of Autism and Developmental Disorders*, vol. 40, p. 269–279, 2010. [Online]. Available: https://doi.org/10.1007/s10803-009-0862-9

[36] J. E. Lainhart and J. Piven, "Diagnosis, treatment, and neurobiology of autism in children," *Curr Opin Pediatr*, vol. 7, p. 392–400, 1995.

[37] C. R. Woodard and J. Chung, "Feasibility of a play-based intervention set for toddlers with autism," *Res Dev Disabil*, vol. 80, p. 24–34, 2018.

[38] D. S. S. J. P. McCleery, N. A. Elliott and C. A. Stefanidou, "Motor development and motor resonance difficulties in autism: relevance to early intervention for language and communication skills," *Front Integr Neurosci*, vol. 7, p. 30, 2013.

[39] Correction and Republication, "Prevalence and characteristics of autism spectrum disorder among children aged 8 years - autism and developmental disabilities monitoring network, 11 sites, united states, 2012," *MMWR Morb Mortal Wkly Rep*, vol. 67, p. 1279, 2018.

[40] S. S. A. B. T. A. M.-B. L. D. P. S. S. R. S. J. Lane, Z. Mailloux and R. C. Schaaf, "Neural foundations of ayres sensory integration®," *Brain Sci*, vol. 9, 2019.

[41] R. C. W. A. J. Marino, E. N. Fletcher and S. E. Anderson, "Amount and environmental predictors of outdoor playtime at home and school: a cross-sectional analysis of a national sample of preschool-aged children attending head start," *Health Place*, vol. 18, p. 1224–1230, 2012.

[42] F. Castelli, "Understanding emotions from standardized facial expressions in autism and normal development," *Autism*, vol. 9, p. 428–449, 2005. [Online]. Available: https://doi.org/10.1177/1362361305056082

[43] J. P. G. Lorenzo, A. Lledó and R. Roig, "Design and application of an immersive virtual reality system to enhance emotional skills for children with autism spectrum disorders," *Comput. Educ.*, vol. 98, p. 192–205, 2016. [Online]. Available: https://doi.org/10.1016/j.compedu.2016.03.018

[44] H. F. A. Schwarze and B. Niehaves, "Advantages and propositions of learning emotion recognition in virtual reality for people with autism," in *Proceedings of the 27th European Conference on Information Systems (ECIS)*, Stockholm Uppsala, Sweden, 2019.

[45] N. H. K. Kim, P. Geiger and M. Rosenthal, "The virtual reality emotion sensitivity test (v-rest): Development and construct validity," in *Proceedings of the Association for Behavioral and Cognitive Therapies (ABCT) Conference*, San Francisco, CA, USA, 2010.

[46] M. G. W. J. N. H. N. M. L. S. M. S. K. Kim, M. Z. Rosenthal and P. Mundy, "A virtual joy-stick study of emotional responses and social motivation in children with autism spectrum disorder," *J. Autism Dev. Disord.*, vol. 45, p. 3891–3899, 2015. [Online]. Available: https://doi.org/10.1007/s10803-014-2036-7

[47] H. M. Y. M. K. A. A. A. D. A. S. G. M. A. Rashidan, S. N. Sidek and S. A. M. Zabidi, "Technology-assisted emotion recognition for autism spectrum disorder (asd) children: A systematic literature review," *IEEE Access*, vol. 9, p. 33638–33653, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3060753

[48] M. D. L. J. M. Garcia-Garcia, V. M. R. Penichet and A. Fernando, "Using emotion recognition technologies to teach children with autism spectrum disorder how to identify and express emotions," *Universal Access in the Information Society*, vol. 21, no. 4, pp. 809–825, 2021. [Online]. Available: https://doi.org/10.1007/s10209-021-00818-y

[49] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. X, p. X, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s42979-021-00815-1

[50] Z.-J. H. A. e. a. Alzubaidi, L., "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," 2021. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00444-8

[51] [Online]. Available: https://link.springer.com/article/10.1007/s42979-021-00815-1

[52] M. Martínez-Comesaña, A. Ogando-Martínez, F. Troncoso-Pastoriza, J. López-Gómez, L. Febrero-Garrido, and E. Granada-Álvarez, "Use of optimised mlp neural networks for spatiotemporal estimation of indoor environmental conditions of existing buildings," 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360132321006429

[53] M. A. Guevara Richaud, "Retropropagación (backpropagation): qué es y cómo funciona en el entrenamiento de redes neuronales," https://antonio-richaud.com/blog/archivo/publicaciones/40-backpropagation.htm, 2024.

[54] [Online]. Available: https://datascientest.com/es/vgg-que-es-este-modelo-daniel-te-lo-cuenta-todo

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[56] [Online]. Available: https://www.researchgate.net/figure/GG16-Network-Structure-4_fig1_357516533

[57] P. Kora, C. Ooi, O. Faust, R. U, A. Gudigar, W. Y. Chan, M. Kollati, K. Swaraja, P. Pławiak, and U. Acharya, "Transfer learning techniques for medical image analysis: A review," *Biocybernetics and Biomedical Engineering*, vol. 42, pp. 79–107, 01 2022. [Online]. Available: https://www.researchgate.net/publication/357011717_Transfer_learning_techniques_for_medical_image_analysis_A_review

[58] Y. Zhou, H. Chang, Y. Lu, X. Lu, and R. Zhou, "Improving the performance of vgg through different granularity feature combinations," 10 2020. [Online]. Available: https://www.researchgate.net/publication/346306126_Improving_the_Performance_of_VGG_Through_Different_Granularity_Feature_Combinations

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[60] [Online]. Available: https://www.researchgate.net/figure/The-framework-of-the-Resnet50-The-Resnet50-model-trained-on-ImageNet-which-is_fig3_344190091

[61] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," pp. 558–567, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8954382

[62] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9134370

[63] A. K. D. W. X. Z. T. U. M. D. M. M. G. H. S. G. J. U. Alexey Dosovitskiy, Lucas Beyer and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[64] A. Tharwat, "Classification assessment methods: a detailed tutorial," 2018. [Online]. Available: https://www.researchgate.net/publication/327148996_Classification_Assessment_Methods_a_detailed_tutorial

[65] Google Developers, "Curso intensivo de aprendizaje automático - google developers," https://developers.google.com/machine-learning/crash-course/classification/thresholding?hl=es-419.

[66] D. M. W. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2020. [Online]. Available: https://arxiv.org/abs/2010.16061

[67] K. Muralidhar, "Curva de aprendizaje para identificar sobreajuste y subajuste en el aprendizaje automático," 2021. [Online]. Available: https://towardsdatascience.com/learning-curve-to-identify-overfitting-underfitting-problems-133177f38df5/

[68] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," 2020. [Online]. Available: https://arxiv.org/abs/2008.05756

[69] K. Valencia, C. Rusu, F. Botella, and E. Jamet, "A methodology to evaluate user experience for people with autism spectrum disorder," 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/22/11340

[70] "How visual stimulation affects your autistic child's daily life," 2025. [Online]. Available: https://www.kidsmentalhealth.ca/how-visual-stimulation-affects-your-autistic-childs-daily-life/