Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

Reconocimiento de Expresiones Faciales Mediante Redes Neuronales Convolucionales Ligeras

TRABAJO RECEPCIONAL PARA OBTENER EL GRADO DE

MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: **VÍCTOR RAÚL CÁRDENAS GIL**

Asesor: **MS. VÍCTOR HUGO MARTÍNEZ SÁNCHEZ**

Tlaquepaque, Jalisco. Julio 2025

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Department of Electronics, Systems, and Informatics

Master's Degree in Computer Science

Facial Expression Recognition Using Lightweight Convolutional Neural Networks

RECEPTIONAL PAPER SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF COMPUTER SCIENCE

Presented by: **VÍCTOR RAÚL CÁRDENAS GIL**

Advisor: **MS. VÍCTOR HUGO MARTÍNEZ SÁNCHEZ**

Tlaquepaque, Jalisco. July 2025

Agradecimientos

Expreso mi más profundo agradecimiento a mi asesor, M.S. Víctor Hugo Martínez Sánchez, por su constante guía y apoyo a lo largo de este proceso.

Al Dr. Iván Esteban Villalón Turrubiates, quien fungió tanto como coordinador de la Maestría durante mi periodo como estudiante como revisor de este Trabajo de Obtención de Grado, siempre atento, generoso y dispuesto a brindar su apoyo.

Al Dr. Víctor Hugo Ortega Guzmán, revisor de ese trabajo, por su valioso tiempo, comentarios reflexivos y contribuciones académicas.

Un agradecimiento especial a todas las personas que generosamente ofrecieron su tiempo e imagen para participar como sujetos del presente sistema de reconocimiento de expresiones faciales, así como a los psicólogos que brindaron su valiosa retroalimentación para enriquecer su proyección hacia contextos clínicos y educativos.

También agradezco a la comunidad *open-source*, cuya generosidad al compartir herramientas, recursos y conocimientos hizo posible —e incluso inspirador— el desarrollo de este trabajo.

Finalmente, este trabajo está dedicado a todas las personas que enfrentan valientemente desafíos relacionados con la salud mental, así como a quienes dedican su tiempo, energía y recursos para brindarles apoyo. Su fortaleza y compromiso otorgan a este trabajo su sentido más profundo.

Figure 1 *Una Figura Humana Resiliente: Manos Tensas, Músculos Contraídos y Una Mano Amiga que Brinda Calma.*

Este dibujo es una reinterpretación personal de una fotografía que vi en internet, cuyo autor original, lamentablemente, no he podido identificar a pesar de varios intentos. La imagen mostraba una figura humana en una pose muy similar a la figura principal de mi boceto, en la que el juego de luces, sombras y gestos transmitía una profunda sensación de lucha interior y resiliencia. Esta obra no pretende atribuirse la autoría creativa del concepto original, sino rendir un homenaje respetuoso a su impacto emocional a través de mi propia interpretación visual.

Acknowledgments

I extend my deepest gratitude to my advisor, M.S. Víctor Hugo Martínez Sánchez, for his continuous guidance and support throughout this journey.

To Dr. Iván Esteban Villalón Turrubiates, who served both as coordinator of the Master's program during my time as a student and as a reviewer for the current work, always being attentive, generous, and supportive.

To Dr. Víctor Hugo Ortega Guzmán, my reviewer, for his valuable time, thoughtful feedback, and academic contributions to this work.

A special thanks to all the individuals who generously offered their time and image to participate as subjects in the present facial expression recognition system, as well as to the psychologists who provided valuable feedback to enrich its projection toward clinical and educational contexts.

I also wish to thank the broader open-source community, whose generosity in sharing tools, resources, and knowledge made this work not only possible, but truly inspiring.

Lastly, this work is dedicated to the many individuals who face mental health challenges, and to those who devote their time, energy, and resources to supporting them. Your strength and commitment give this work the deepest meaning.

Figure 2 *A Resilient Human Figure: Clenched Hands, Tense Muscles, and a Calming Helping Hand.*

This drawing is a personal reinterpretation of a photograph I once encountered on the internet, whose original author I have unfortunately been unable to identify despite several attempts. The reference featured a human figure in a pose closely resembling the main figure in my sketch, where the interplay of light, shadow, and gesture conveyed a profound sense of inner struggle and resilience. This piece does not intend to claim creative ownership over the original concept, but rather to pay respectful tribute to its emotional resonance through my own visual interpretation.

Dedicatoria

A mi madre, Olga, por su incansable insistencia de que todo se puede, y por habernos dado todo siempre. Mamá, viéndote a ti creo en Dios, creo en la vida y en el amor. Te ama, tu hijo.

A mi padre, Raúl, por su atento consejo y por inculcarme el hábito de preguntarme cómo dar lo mejor de mí.

A mi hermana, Denisse, por su decidido esmero de vivir la vida en el más profundo sentido. Viéndote, quiero viajar, leer, respirar y hacer todo con plena conciencia.

Y finalmente, a ti, Ursula. Todos estos años has sido mi mejor amiga, mi compañera de vida y mi razón. Porque fueron años llenos de retos que, a tu lado, siempre encontraron inspiración, amor y alegría. Me llenas el corazón de agua dulce.

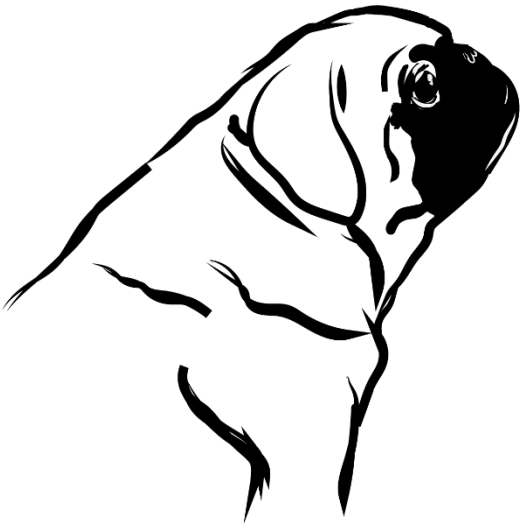

Figure 3 *Godot: Atento, Esperando, Acompañando.*

Dedication

To my mother, Olga, for her unwavering belief that anything is possible, and for always giving us everything. Mom, seeing you makes me believe in God, in life, and in love. With all my love — your son.

To my father, Raúl, for his thoughtful advice and for instilling in me the habit of giving my best.

To my sister, Denisse, for your determined devotion to living life in its fullest sense. Watching you makes me want to travel, read, breathe, and do everything with deep awareness.

And finally, to you, Ursula. All these years, you have been my best friend, my life partner, and my reason. These were years full of challenges that, by your side, always found inspiration, love, and joy. You taught both of us to never give up — and good luck always found us. And it always will.

Figure 4 *Godot: Watchful, Waiting, Present.*

Abstract

Facial Expression Recognition (FER) is an active research area within Artificial Intelligence (AI) with increasing relevance in real-world applications [1][2][3][4][5]. This work explores the development of a deep learning-based FER system focused on achieving competitive performance using lightweight architectures that are suitable for environments with limited computational resources.

While high-capacity models were initially explored, their computational requirements exceeded the available hardware, prompting a shift in focus toward lightweight alternatives. The final system was built around ResNet-18 [6] and trained using transfer learning on a hybrid dataset comprising real-world, AI-generated, and publicly available images from MMI [7], OULU-CASIA [8], EFE [9], FERD [10] and AffectNet [11].

Experimental results showed that the proposed ResNet-18 model achieved a mean accuracy of 91.74% ± 0.40% (n=3), with a maximum observed accuracy of 92.27%. EfficientNet [12] and MobileNetV3 [13] were also evaluated and achieved competitive accuracy levels; however, their training curves plateaued early, suggesting unstable learning and limited convergence compared to ResNet-18.

Although the system was not deployed in real-world scenarios, its compact design and strong results on a diverse, resolution-consistent dataset indicate its potential for future application in low-resource settings.

Resumen

El reconocimiento de expresiones faciales (FER, por sus siglas en inglés) es un área de investigación activa dentro de la inteligencia artificial (IA), con una relevancia creciente en aplicaciones del mundo real [1][2][3][4][5]. Este trabajo explora el desarrollo de un sistema FER basado en aprendizaje profundo, enfocado en alcanzar un desempeño competitivo utilizando arquitecturas ligeras, adecuadas para entornos con recursos computacionales limitados.

Si bien en un inicio se exploraron modelos de alta capacidad, sus requerimientos computacionales excedieron las capacidades del hardware disponible, lo que motivó a un cambio de enfoque hacia alternativas más ligeras. El sistema final fue construido en torno a ResNet-18 [6] y entrenado mediante aprendizaje por transferencia sobre un conjunto de datos híbrido que incluye imágenes del mundo real, imágenes generadas por IA e imágenes provenientes de conjuntos públicos como son MMI [7], OULU-CASIA [8], EFE [9], FERD [10] y AffectNet [11].

Los resultados experimentales mostraron que el modelo propuesto basado en ResNet-18 alcanzó una precisión media de 91.74% ± 0.40% (n=3), con una precisión máxima de 92.27%. También se evaluaron EfficientNet [12] y MobileNetV3 [13] los cuales lograron niveles competitivos de precisión; sin embargo, sus curvas de entrenamiento se estabilizaron prematuramente, lo que sugiere un aprendizaje inestable y una convergencia limitada en comparación con ResNet-18.

El diseño ligero del modelo y sus resultados sobre un diverso conjunto de datos indican su potencial para futuras aplicaciones en contextos con recursos limitados.

Table of Contents

Table of Figures

Table of Tables

Table of Abbreviations

Table 1 Common Abbreviations Used Throughout the Document.

This table presents a list of abbreviations and their corresponding full terms as used throughout the work.

Abbreviation	**Full Term**
AI	Artificial Intelligence
ANN	Artificial Neural Network
CAGR	Compound Annual Growth Rate
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DL	Deep Learning
FC	Fully Connected (layer)
FER	Facial Expression Recognition
FN	False Negative
FP	False Positive
FT	Fine-Tuning
GPU	Graphics Processing Unit
LMIC	Low- and Middle-Income Countries
MAD	Multi-Attention Dropping
ML	Machine Learning
MSAD	Multi-head Self-Attention Dropping
NLP	Natural Language Processing
RAM	Random Access Memory
ReLU	Rectified Linear Unit
ResNet	Residual Network
SE	Squeeze-and-Excitation
SSD	Solid State Drive
TL	Transfer Learning
TN	True Negative
TP	True Positive
UX	User Experience
ViT	Vision Transformer

01. INTRODUCTION

Facial Expression Recognition (FER) is a growing field within the Artificial Intelligence (AI) domain. It is centered on creating machines able to interpret human emotions through facial cues. FER combines advances in computer vision, machine learning, and neural networks. This project aims to train, and evaluate a deep learning FER system capable of identifying seven emotional states—the six basic human emotions proposed by Paul Ekman: anger, disgust, fear, happiness, sadness, and surprise [14] in addition to neutrality.

The motivation behind this project is rooted in the increasing relevance of emotionally intelligent systems across industries such as education, health, customer experience [15], and human-computer interaction. In particular, systems capable of classifying emotional states through facial cues offer valuable support in the field of mental health [5]. They can help monitor emotional well-being, detect signs of emotional distress, and support therapy or intervention tools—especially in digital or remote settings. In regions with limited access to professional psychological care, automated emotion recognition tools could play a supportive role in monitoring well-being and enabling early interventions.

The approach integrates transfer learning, a modular *MLFlow* experimentation framework, and real-world evaluation scenarios using consumer-grade cameras (e.g., Dell G5 webcam and iPhone 12 mini). While the system's main model (ResNet-18 [6]) was selected for its accuracy and efficiency, additional lightweight architectures such as MobileNetV3 [13] were also evaluated for potential deployment on mobile and embedded devices.

This multi-model strategy supports the overarching goal of developing FER systems that are both scalable and practical across varying computational constraints. The final system demonstrates high classification performance on a hybrid, curated dataset and includes functionality for real-time inference and visualization. These features suggest the system's potential utility in applications such as emotion monitoring, educational tools, and mental health support platforms.

01.1 Background

***Brief**: This section provides an overview of the foundational concepts underpinning the development of the proposed emotion recognition system. It outlines the evolution of FER as a research domain, the role of Convolutional Neural Networks (CNNs) in this field, its interdisciplinary relevance, and the primary challenges faced in real-world applications.*

01.1.1 Facial Expression Recognition (FER)

FER consists in an interdisciplinary field that merges computer vision and psychological theory to analyze human emotional states. FER models aim to classify facial expressions into discrete emotional categories [14] further providing actionable data for various applications.

Recent advances in AI have significantly transformed the development of FER systems. Traditional methods used more manual approaches such as geometric relationships or texture descriptors like

Local Binary Patterns [16]. These methods struggled with generalization in real-world conditions—some of these being varying lighting, face occlusions, and subtle emotional cues or micro-expressions [16]. Recent research on FER has focused on improving models through different architectures mainly focused on CNN and Transformers [16].

01.1.2 Convolutional Neural Networks (CNNs)

CNNs, a key category of deep learning models, are capable of extracting layered visual representations directly from facial image data [17]. This processing enables the network to understand varying levels of abstraction: initial layers focus on simple visual features like edges and textures, middle layers identify actual facial elements such as eyes and mouth, and the deepest layers capture complex expressions and subtle emotional indicators [17].

This hierarchical learning enables the model to identify elaborate facial features—such as fine muscle movements or subtle asymmetries—that are often critical in distinguishing between similar expressions [17]. As a result, these models are particularly well-suited for FER tasks, as they do not rely on handcrafted features but instead learn directly from the data in a structured and scalable manner [16].

01.1.3 Interdisciplinary Applications of FER

FER not only serves commercial applications like marketing, UX design [15], and entertainment, but also holds strong potential in mental health monitoring. By detecting subtle expressions of distress, emotional withdrawal, or mood changes, FER tools can complement psychological assessments or power digital mental health interventions [18].

01.1.4 Real-World Challenges

Deploying FER in real-world environments introduces several challenges: occlusions (e.g., glasses, masks), varying lighting conditions, and non-standard expression dynamics [19]. To address these, the system was trained and tested using a mixture of spontaneous and acted expressions from diverse sources.

Now that the foundational concepts surrounding FER have been established, it becomes essential to reflect on why integrating these components into a unified system is both relevant and necessary. The following section explores the economic, social, and technological rationale that justifies the development of projects like this, especially in settings where scalable, emotionally aware systems can offer meaningful support.

01.2 Justification

***Brief**: Developing an AI-based emotion recognition system for mental health applications is justified from economic, social, and technological perspectives. This section delves into the key opportunity*

areas within each of these domains to highlight the relevance and potential impact of the proposed system.

01.2.1 Economic Justification

Automated emotion recognition is emerging as a promising, cost-efficient alternative to traditional approaches for monitoring mental health. By minimizing the need for constant human involvement, these systems offer a more scalable way to support diagnostics and early detection. According to [20], the global market for emotion recognition technologies was valued at $21.7 billion in 2021 and is expected to grow to $136.2 billion by 2031—representing an impressive compound annual growth rate (CAGR) of 20.5% [20].

To give that figure some context: this expected growth outpaces the pharmaceutical industry, forecasted to grow at a CAGR of 7.1% between 2022 and 2031 [21], and even surpasses the overall AI sector, presumed at 19.2% CAGR from 2025 to 2034 [22]. These comparisons reflect just how rapidly emotion recognition is gaining traction—not just in healthcare, but also in fields like human-computer interaction, automotive technologies, and security.

Beyond market trends, there are strong financial and societal reasons to invest in this technology. Untreated or late-diagnosed mental health issues carry a steep global price. The World Economic Forum [23] reported that in 2010, such conditions cost the world economy around $2.5 trillion—$1.7 trillion from lost productivity and $0.8 trillion from direct medical expenses. Alarmingly, these costs are projected to reach $6 trillion by 2030, surpassing the combined financial burden of cancer, diabetes, and chronic respiratory diseases [23].

By contributing to early monitoring strategies, facial emotion recognition systems could help mitigate both personal suffering and broader economic consequences associated with delayed access to mental health care.

01.2.2 Social Justification

Mental health disorders such as depression, anxiety, and emotional dysregulation often manifest through subtle changes in facial expression [24]. Despite growing technological advances, mental health diagnoses still largely depend on self-reported symptoms and face-to-face evaluations [24]. These methods can be limiting [25]—they often require significant time, may not be readily accessible to everyone, and can carry social stigma that discourages individuals from seeking help. Considering that more than 970 million people around the world are currently living with mental health conditions [23], there is a growing need for non-invasive, objective monitoring tools.

This system offers a discreet and judgment-free way to track emotional well-being, lowering psychological and logistical barriers to care. By enabling deployment through smartphones or cloud platforms, the system ensures greater reach, particularly for rural or underserved populations. Ultimately, it promotes early detection, reduces stigma, and empowers users to take a proactive role in managing their emotional health.

> ‘’People with severe mental
>
> health conditions die
>
> 10 to 20 years
>
> earlier than the general
>
> population.” [23]

01.2.3 Technological Justification

This project utilizes ResNet-18, a deep CNN architecture known for its balance of accuracy and computational efficiency [26]. Its relatively low parameter count (11.2M) and inference time (~0.0055s/image) make it a suitable candidate for real-time deployment, even in mid-tier hardware environments, unlike more recent transformer-based models [16].

In addition to the main model, lightweight convolutional architectures (MobileNetV3-small, MobileNetV3-large, EfficientNetB0) were included in the experimental framework. Although their training curves remained unstable and failed to demonstrate consistent convergence, they delivered competitive accuracy scores, demonstrating promise for edge deployment scenarios where inference speed and resource constraints are critical.

The experimentation framework was supported by MLflow, which allowed structured hyperparameter tuning, reproducibility, and version tracking. The training pipeline included modern training strategies such as cosine annealing and early stopping to ensure convergence while avoiding overfitting, even with a limited number of epochs.

Although this system has not yet been integrated into clinical workflows, its design aligns with the broader objective of making emotionally aware systems more accessible. It contributes to ongoing work in affective computing by showing how deep learning models can be trained on hybrid datasets—including synthetic and real-world facial expressions—to approximate emotional recognition capabilities. Future iterations may expand these foundations toward more formal psychological validation or mental health integration.

While the economic, social, and technological dimensions discussed in this section illustrate the promising potential of emotion recognition systems, it is important to recognize that these opportunities are framed within a broader set of global mental health challenges. Understanding the scale, complexity, and limitations of current emotional monitoring practices is essential for situating the scope and relevance of the system proposed in this work. The following section explores this

context in more detail, outlining the key issues that motivate the development of scalable, non-invasive, and accessible tools for emotional insight.

01.3 Problem Statement

***Brief**: This section presents key statistics and evidence that underscore the global potential of implementing AI-based tools to support mental health.*

01.3.1 Mental Health: Current Challenges

The increasing global burden of mental health conditions has brought attention to the need for complementary technological solutions that can support early monitoring and emotional well-being. While clinical evaluation remains the gold standard, current monitoring methods—such as self-reporting—are often subjective and not easily scalable. This is especially problematic in regions with limited mental health infrastructure or where cultural stigma may deter individuals from seeking help [23].

According to the World Health Organization [23], mental health services represent, on average, only 2% of national healthcare budgets, with less than 20% of that funding allocated to community-based services [23]. These limitations are exacerbated by a global shortage of trained professionals, particularly in low-income countries where the mental health workforce is severely under-resourced [23].

This context creates an opportunity for alternative tools that, while not replacing professional care, can offer preliminary insights and emotional awareness in non-intrusive ways. Lightweight and scalable systems, such as the FER model proposed in this work, can operate on widely available devices and contribute to improving emotional literacy and awareness, especially in underserved communities.

The present system does not aim to diagnose or treat mental health conditions but instead contributes to the broader ecosystem of digital tools that may support emotional self-monitoring. Its low hardware requirements and real-time capabilities are aligned with the goal of increasing accessibility to emotion-aware technologies in everyday environments [3].

Given these structural and logistical barriers to timely emotional health support, it becomes necessary to explore solutions that can operate within existing constraints and extend access through technological means. The next section outlines the objectives that guided this project, emphasizing both its technical scope and its potential contribution to expanding digital emotional health tools.

01.4 Objectives

***Brief**: This section outlines the general and specific objectives of the present work. The focus is on the development and evaluation of a compact DL model for FER, emphasizing technical feasibility, model performance, and deployment considerations in low-resource environments.*

01.4.1 General Objective:

To design, train, and evaluate a FER system based on a pretrained ResNet-18 neural network, using a hybrid dataset composed of curated and public facial images with consistent resolution, to achieve strong classification performance while maintaining a compact architecture suitable for future deployment in resource-constrained environments.

01.4.2 Specific Objectives:

- To fine-tune the ResNet-18 model using a hybrid dataset composed of publicly available and manually curated facial expression images.
- To implement a preprocessing pipeline aimed at improving model robustness against real-world variability.
- To assess the model's performance using standard classification metrics such as accuracy, precision, recall, and F1-score.
- To implement a modular experimentation framework for systematic tuning and architecture comparison.
- To test real-time performance using laptop and mobile cameras and evaluate model behavior in regular conditions.
- To analyze the potential societal and economic impact of deploying emotion recognition technology as part of a scalable digital mental health solution.

Having outlined the goals of this work, it is now pertinent to examine the specific technical strategies implemented to achieve them. While the project does not claim to introduce groundbreaking advances in architecture or clinical validation, it does propose a carefully structured system that integrates established machine learning methods with design decisions tailored for realistic usage conditions. The following section describes the core technical contributions of the project.

01.5 Technological Innovation

Brief: *The following section details the key technical innovations that define this project, including a transfer learning strategy, a structured preprocessing pipeline, and the use of a synthetic dataset to enhance training efficiency and model generalization.*

01.5.1 ResNet-18 Adaptation

The system fine-tunes a pretrained ResNet-18 model to perform emotion classification tasks with high accuracy, while maintaining relatively low computational demands. By employing transfer learning, the model adapts effectively to the specific characteristics of emotional expressions across varied datasets, avoiding the need for extensive data collection or high-end hardware. This makes the system suitable for real-time or embedded deployment scenarios.

01.5.2 Preprocessing Pipeline

To address real-world variability in facial image data, the project incorporates a preprocessing pipeline. This includes image normalization, resizing, and augmentation techniques including brightness adjustments, flipping, and noise insertion— looking to mimic natural conditions as well as a mobile camera quality and output image. These steps help mitigate challenges posed by occlusions, lighting variations, and pose distortions, enhancing the model's generalization across uncontrolled environments.

01.5.3 Manually Curated and AI-Synthesized Data

A key innovation lies in the manual curation and AI-based augmentation of the training dataset. In addition to incorporating images from established sources like AffectNet [11], and Oulu-CASIA [8], the dataset was enriched with synthetic facial expressions generated using large-scale AI image generators (e.g., *Gemini*, *Meta*, *Stable Diffusion [27], Copilot*), and manually searched images from the web, books and cartoons. This hybrid dataset accounts for variations in age, gender, race, emotion intensity, and expression ambiguity—helping the model become more resilient and inclusive in real-world applications.

Although the synthetic data has limitations in terms of realism and labeling accuracy, it provided a pragmatic solution to data scarcity, allowing for the training of a performant model within resource constraints.

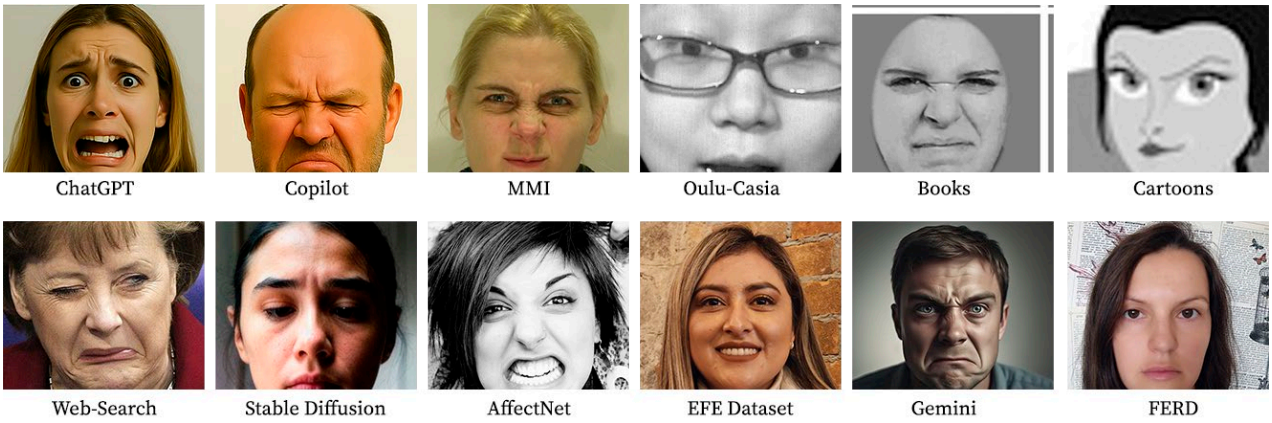

Figure 5 *Representative Samples from our Hybrid FER Dataset Showing Different Emotion Classes—captions below image show the images source.*

Dataset comprises 35,887 images from multiple sources including AffectNet [11], MMI [7], OULU-CASIA[8] , and AI-synthesized samples. All images are normalized to 256×256 pixels.

The strategies implemented in this project reflect a deliberate and pragmatic approach to building a functional FER system under constrained conditions. These decisions aimed to balance training feasibility with real-world applicability. In order to contextualize these contributions within the broader landscape of FER research, the following section reviews the current state of the art in FER.

02. STATE OF THE ART

Brief: *The following section examines the state-of-the-art approaches that inform and contextualize the methodological choices.*

02.1 Latest Techniques in FER

Recent developments in FER have primarily focused on enhancing the precision and adaptability of models, especially when applied to varied and challenging datasets. Among the most influential directions in the field are the growing use of ensemble learning techniques and the widespread integration of Transformer-based models. These methods have achieved leading results on multiple FER benchmarks and are actively shaping the evolution of performance expectations in the domain. [19][16].

02.2 Ensemble Methods

Ensemble learning has emerged as a widely adopted approach for improving model reliability and reducing the limitations often encountered in individual architectures. By taking advantage of the complementary capabilities of multiple models, researchers look to boost performance consistency and minimize systemic bias. One effective strategy has been the usage of CNNs with auxiliary components, such as attention layers or external classification modules [16].

An example is the work by [16], introducing a dual-attention CNN framework. The first attention mechanism operates at a fine-grained level, analyzing spatial facial features through a grid-based method, while the second incorporates a Transformer module to capture abstract semantic patterns. This combined architecture attained perfect accuracy on the CK+ dataset—despite the absence of supplemental training data—demonstrating the effectiveness of unifying both localized and high-level feature extraction [16].

02.3 Transformer Architectures

Initially developed for Natural Language Processing (NLP) tasks [28], Transformer models have demonstrated strong performance in computer vision, including FER tasks. Vision Transformers (ViTs) interpret images as sequences of patches, allowing them to model long-range dependencies across different facial regions—a task that CNNs often struggle with [16].

Several FER systems based on Transformers have reported near-flawless classification outcomes on widely used benchmark datasets [16]. For instance, Aouayeb et al. [16] introduced a Vision Transformer (ViT) model enhanced with Squeeze-and-Excitation (SE) blocks, achieving a remarkable 99.8% accuracy on the CK+ dataset. In another study, Xue et al. [16] developed the TransFER model, which incorporates ViT modules alongside Multi-Attention Dropping (MAD) and Multi-head Self-Attention Dropping (MSAD) to capture differential relevance across local facial regions [16].

Among the most advanced solutions to date is FER-former, proposed by [29] which combines embedding strategies, self-attention layers, and domain-guided supervision techniques. This approach

reached top-tier accuracy scores on several datasets—90.96% on FER+, 91.30% on RAF-DB, and 62.1% on SFEW 2.0—outperforming both conventional CNNs and less complex ViT-based alternatives [29].

02.4 Comparison and Practical Impact

Both ensemble methods and Transformer-based models have propelled FER to new heights. Ensemble frameworks are particularly effective at addressing class imbalances and improving generalization, while Transformers capture nuanced, global facial dependencies with high fidelity [16]. Hybrid models such as FER-former [29] demonstrate that the combination of CNN-based feature extraction and Transformer-based sequence modeling is currently the most effective approach for FER.

The reviewed literature demonstrates that FER has rapidly evolved through the integration of advanced architectures such as Vision Transformers and ensemble-based approaches [16]. These systems represent the cutting edge of what is currently achievable in emotion recognition tasks. However, they also tend to demand significant computational resources and large-scale datasets, making them less practical for certain real-world applications. As mentioned, this work takes a more constrained but focused approach—prioritizing feasibility, and reproducibility. To better understand the conceptual underpinnings of this approach, the following section introduces the theoretical foundations that guided the development of the system.

03. THEORETICAL FRAMEWORK

Brief: *This section offers a hierarchical overview of the core concepts and technologies that underpin the development of the proposed system. It begins with broad topics such as AI and ML, and progressively narrows down to the specific methodologies employed in this work, focusing on CNNs and the ResNet architecture.*

03.1 Artificial Intelligence (AI)

Artificial Intelligence encompasses the creation of computational systems that can carry out functions traditionally associated with human reasoning [30], such as identifying patterns, making informed decisions, and adapting through learning. In this work, AI is leveraged to enable machines to detect and interpret facial cues, allowing them to deduce an individual's emotional state. This capability forms the conceptual base from which more specialized domains like machine learning and neural networks operate [31].

03.2 Machine Learning (ML)

Machine Learning, a branch within AI, concentrates on the design of algorithms that extract knowledge from data to support prediction and decision-making tasks [17]. In this project, a supervised learning framework is employed—meaning the model is trained using labeled datasets of facial expressions, each corresponding to a specific emotion (e.g., joy, sadness, anger). The following sections describe in detail the neural models used to implement these strategies as well as specificities about transfer learning.

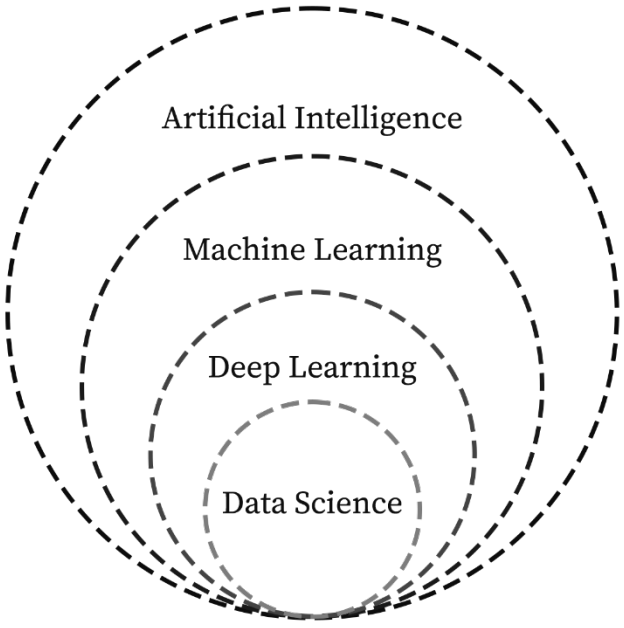

Figure 6 *Hierarchical Relationship between AI, ML, And DL Methodologies.*

The present FER approach utilizes deep learning techniques, specifically CNNs with transfer learning, positioned within the broader ML framework. Adapted from [32]).

03.3 Artificial Neural Networks and Deep Learning

Artificial Neural Networks (ANNs) are systems which emulate the way the human brain works.[17]. Like the human brain, ANNs are conformed by multiple layers of neurons which contribute to generate the computational equivalent of reasoning. Each neuron processes incoming data by applying weighted computations—essentially solving a mathematical equation— followed by the application of a non-linear function. This architecture allows ANNs to extract complex patterns and high-level features from input data, positioning them as one of the dominant architectures in the field of image classification and recognition [17].

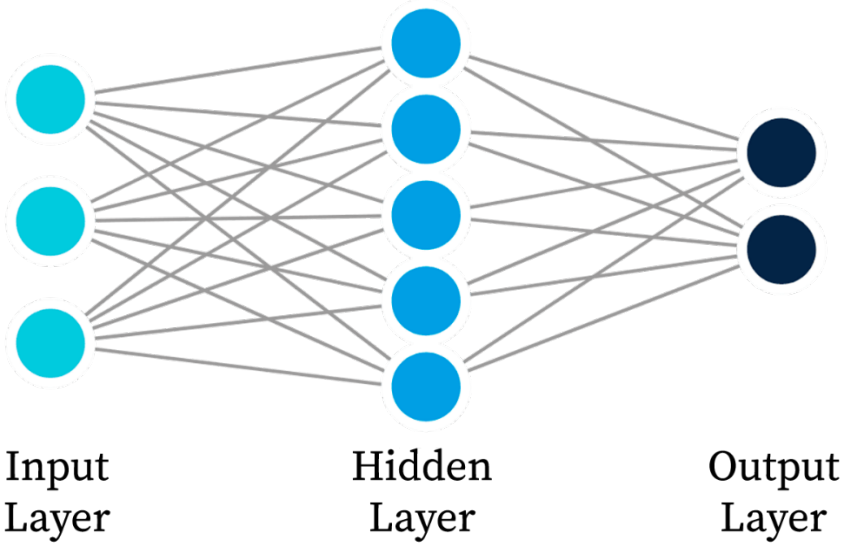

Figure 7 *Fundamental Architecture of an ANN Showing Input Layer, Hidden Layers with Weighted Connections, and Output Layer with Activation Functions.*

This foundational structure is extended in CNNs for visual feature extraction in our FER system. Adapted from [33].

Deep Learning (DL) is a subdivision of ML focused on the use of neural networks with many hidden layers [17] —specifically, more than two [31]. These neurons are capable of learning elaborate, high-level patterns from raw data inputs [17]. A popular DL use in terms of handling visual data are CNNs, recognized for their effectiveness in obtaining valuable information from images [31][34][35][18].

The next section outlines the structure and function of CNNs, emphasizing their critical role in advancing facial emotion recognition systems.

03.4 Convolutional Neural Networks

CNNs are one category of DL designed to handle structured data, particularly images arranged in two-dimensional grids. As outlined by [36], CNNs construct multi-level feature representations from spatial data through core blocks like convolutional layers, pooling layers, and dense (fully connected) layers—all trained using backpropagation.

In a CNN, convolutional layers apply sets of trainable filters that move across the input image to capture localized visual patterns such as edges or textures [17]. Next come pooling layers, reducing the dimension of the feature maps to lower computational costs while also promoting spatial invariance [37]. At deeper levels, fully connected layers consolidate the extracted information and associate it with output targets, such as emotion classes or numeric predictions [37].

In this project, a CNN model called ResNet-18 is utilized. This architecture includes residual links that support the smooth flow of gradients throughout the training process. These shortcut paths are designed to reduce the impact of the vanishing gradient problem, helping the network train more reliably and efficiently as its depth increases [6].

03.5 Layers for Neural Networks

Neural networks—such as the one developed in this work—are composed of different types of layers that define their architecture. To better understand their function, Figure 8 will be analyzed step by step, highlighting the role and impact of each layer in the overall workflow.

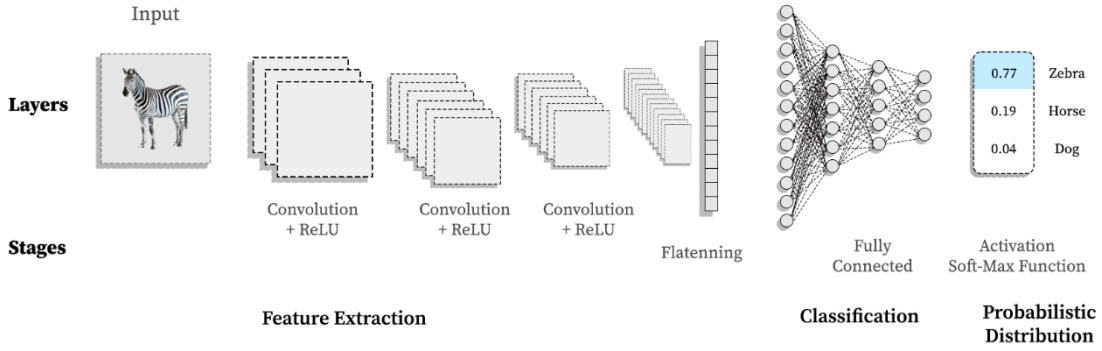

Figure 8 *General Architecture of a CNN.*

The image illustrates the typical structure of a CNN used for image classification tasks. The model performs feature extraction through convolutional and activation layers, followed by flattening and classification via fully connected layers. A softmax function at the output produces a probabilistic distribution over target classes. Adapted from [38].

03.5.1 Convolutional Layers:

These form the foundational components of CNN models. Each of these layers applies a group of learnable filters—commonly known as kernels—that slide over the image to detect spatial characteristics. These filters are trained to respond to specific local features such as lines, textures, or edges [31]. The resulting output, referred to as a feature map, highlights regions where certain patterns

are present. As the model grows deeper, it can learn increasingly complex visual abstractions, leading to a layered understanding of the input image [31].

03.5.2 Activation Layers (ReLU):

After each convolutional operation, the model applies an activation function to introduce non-linearity, a key aspect for capturing complex data relationships beyond simple linear trends [31]. Without this mechanism, the network would be limited to learning only linear transformations. One of the most frequently implemented activation functions is ReLU (Rectified Linear Unit), which transforms the input by setting any value below zero to zero, while preserving values that are positive. This non-linear transformation allows the model to better identify nuanced patterns like curves, thresholds, or interactions—particularly useful in visual data and emotional analysis contexts.

03.5.3 Pooling Layers:

Pooling layers serve to shrink the dimensions of feature maps, reducing computational overhead and minimizing the risk of overfitting. The most widely used method, max pooling, filters the strongest activation from each small region of the feature map [31]. This approach ensures that the most critical information is preserved while also increasing the network's ability to tolerate minor variations or distortions in the input data.

03.5.4 Fully Connected Layers:

Towards the final stages of the network, the complex features captured by earlier layers are condensed into a single vector and passed through one or more densely connected layers [31]. In these layers, each node connects to every output from the previous layer, allowing the model to combine all learned information into a unified representation. This final step enables the network to reason through the extracted features and determine the most likely classification outcome.

03.5.5 Softmax Output Layer:

The final component in the architecture is the softmax layer. This layer transforms the raw scores (logits) from the previous dense layer into a normalized probability output [31]. This conversion allows each output to be interpreted as the model's confidence in one of seven possible emotional states aforementioned. By ensuring that the total probability sums to one, the softmax layer produces interpretable results that reflect the network's predictions [31].

03.6 ResNet-18

Proposed by [6] this architecture is notable for introducing residual connections—shortcuts that allow gradients to skip one or more layers during training. These skip connections help address one common learning challenge called vanishing gradient. In practical terms, it offers a strong balance between computational efficiency and accuracy [26], making it a suitable option for tasks that involve limited computing resources and image classification.

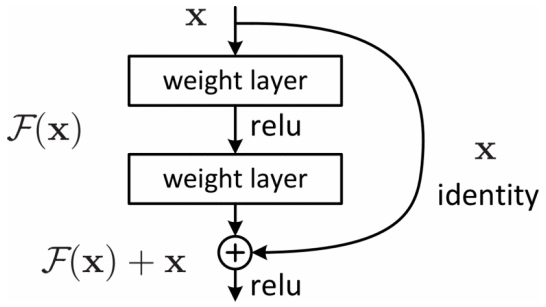

Figure 9 *Residual Block Architecture Showing Skip Connections that Enable Trainig of Deeper Networks*

Input x bypasses weight layers through identity mapping, allowing gradients to flow directly and mitigating vanishing gradient problem. This building block is core to ResNet-18 used in the FER system. Source: [6]

The figure above illustrates a residual block, which is the core building component of ResNets. A residual block is essentially a small sub-network that processes the input and then adds it back to the original input before passing the result to the next layer. This idea is known as a shortcut connection or skip connection, because it allows the original input to "skip" over a few layers and be reused later in the network[6].

Instead of learning a complete transformation from the *input x* to the output *H(x)* - also called a direct mapping - the network learns only what needs to be changed or corrected [6], which is called the residual *F(x)*. In other words, the network focuses on learning the difference between the input and the desired output [6], not the entire output itself.

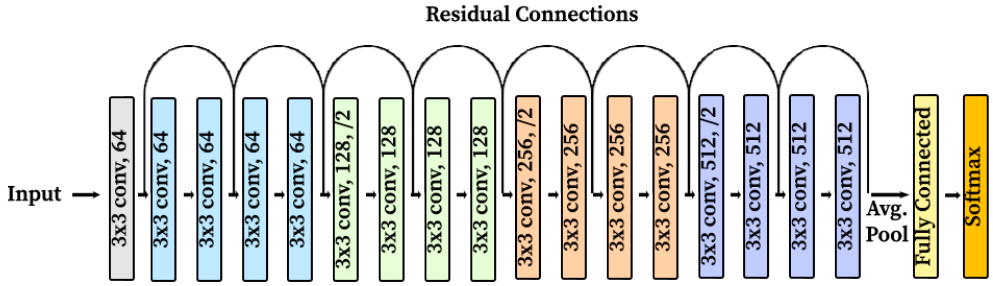

Figure 10 *Complete Resnet-18 Architecture Adapted for FER.*

Network processes 256×256 input images through four residual blocks with increasing channel depth (64→512) Dotted arrows indicate skip connections with stride=2 for spatial downsampling. Adapted from [4]

Figure 10 presents the original ResNet-18 structure, a CNN comprising 18 trainable layers. This configuration is what gives the model its name, *ResNet-18 [6]*. The architecture includes a stack of convolutional layers, residual (or shortcut) connections, a global average pooling stage, and a fully connected (FC) layer that ends with a *softmax* activation for output classification.

The model begins by receiving an input tensor—in this case, a grayscale image with one channel (referred to as input)—and processes it using a 3×3 convolutional filter with 64 channels [39]. This first layer captures local features, producing 64 feature maps. Each convolutional step typically involves a latter batch normalization and ReLU activation, though these components are not explicitly shown in the diagram.

The main body of the model contains four residual blocks, each shown in different colors (e.g., blue, green, orange, and purple) in Figure 10. Every block consists of two convolutional layers with equal output sizes. Each unit labeled 3×3 conv <n> corresponds to a 3×3 convolutional operation producing <n> output channels. For example, a block labeled 3×3 conv, 128 includes two such convolutional layers generating 128 feature maps each [4].

In the first convolutional layer of each residual block (excluding the initial one), a stride of 2 is applied, as shown by the notation /2—this operation reduces the height and width of the resulting feature maps by half. This downsampling step compresses spatial detail while increasing the model's representational depth [4].

The curved arrows represent residual or skip connections, which are a defining feature of the ResNet architecture. These connections enable input from a residual block to be directly added to its corresponding output. This bypasses the intermediate convolutional layers. This mechanism facilitates gradient flow during backpropagation, combating the vanishing gradient problem and enabling deeper network training.

After the final residual block—responsible for generating 512 feature maps —the network applies an average pooling layer (Avg Pool) across each feature map. This operation compresses the information into a single, fixed-size feature vector.

This vector then enters a fully connected (FC) layer, which functions as a classifier by converting the learned features into a vector sized according to the number of output categories. Finally, the output from the FC layer is passed to a softmax function, which converts the raw values into a probability distribution across the possible emotion classes.

In total, the architecture includes:

- 1 initial convolutional layer
- 8 residual blocks, each with 2 convolutional layers (totaling 16 conv layers)
- 1 fully connected layer

Resulting in 18 weight-bearing layers, which define the depth of ResNet-18 [4].

03.7 Transfer learning

[40] define transfer learning (TL) as "the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned". This is particularly valuable if the target task is short of labeled data. TL avoids training networks from scratch and uses pre-trained model's learned features to help faster convergence and better performance.

In this work, TL is implemented by utilizing the pre-initialized weights of the ResNet-18 model, which had been originally trained using the ImageNet [41] dataset. Rather than building the network from the ground up, the model takes advantage of previously learned feature extraction capabilities that originated from a broad and varied image corpus.

This strategy allows the model to draw on general visual features—such as shapes, edges, and textures—and repurpose them for the specific task of recognizing facial expressions. To achieve this, the early convolutional layers are preserved, while the upper layers are retrained using a specialized facial emotion dataset based on Ekman's [14] six universal emotions. This commands the model to focus on emotion-specific features, significantly reducing the amount of data and computing power needed for training.

03.8 Fine-Tuning

Fine-tuning is a focused training method used in the context of TL, where a neural network that has already been trained on a large-scale dataset is adapted further on a smaller, task-specific dataset [31][42]:

- Copying the architecture and learned weights from a pre-trained model, with the exception of the final classification layer, which is replaced to align with the new task's output classes [42] [43].
- Freezing some of the early layers (to preserve generic feature representations) and updating the remaining layers during training [42] .

[43] further clarifies this as "the most widely used approach for transfer learning when working with deep learning models," typically achieved using ImageNet-pretrained models for computer vision tasks

In this project, ResNet-18 was initially pretrained on ImageNet [44], which consists of more than 1.2 million labeled images spanning 1,000 different object classes, to leverage its robust visual feature representations. Transfer learning was then employed, followed by fine-tuning using a custom dataset based on Ekman's emotion theory [14], as previously described.

03.9 Hyperparameters

These are configurable elements of a model that directly impact its performance. As noted by [45], "these are settings not learned from the data but set prior to the training process."

03.9.1 Hyperparameter Optimization

Hyperparameter optimization —also called hyperparameter tuning— can be defined as the process of adjusting non-learnable model settings to identify the configuration that produces the most effective model performance [46].

As noted by [47], some of the most effective hyperparameters for tuning in CNNs include the number of convolutional layers, learning rate, dropout rate, optimizer, and number of epochs. In this work, weight decay and data augmentation were also considered, as several studies have demonstrated their relevance in contributing to model performance optimization [48] [38].

03.9.2 Data Augmentation

These terms refer to a set of techniques used to artificially increase in number and diversify a dataset to improve training performance. This is achieved by applying transformations—horizontal flip, vertical, flip, noise induction— to existing data instances, resulting in slightly modified versions of the original images. These variations help improve model generalization, particularly in scenarios with limited labeled data [48].

03.9.3 Weight Decay

A regularization technique used to mitigate overfitting [38]. Based on the L2 norm, its purpose is to prevent any single weight from having disproportionate influence on the model's performance and the overall optimization process [38]. In simpler terms, this hyperparameter helps the model avoid over-relying on specific features that may not be genuinely informative for distinguishing between emotions.

For example, without weight decay, the model might focus excessively on the shape of the mouth to determine if someone is happy. However, weight decay encourages the model to distribute importance more evenly—considering not only the mouth but also other relevant regions like the eyes, cheeks, or a wrinkled nose. This leads to more generalizable representations.

The formula for weight decay can be expressed as [38]:

$$L_{\text{total}} = L_{\text{original}} + \lambda \sum_{i} w_i^2 \quad (1)$$

- L_{total} is the total loss with regularization
- L_{original} is the original loss function (in this case, cross entropy)
- λ is the weight decay coefficient (the factor or penalization)
- w_i^2 are the individual weights of the model

This addition to the loss function encourages the model to keep the weights small, which contributes to more stable and generalizable learning.

03.9.4 Dropout

Another regularization technique used to prevent overfitting by randomly "dropping" a given amount or neurons per pass during training. This prevents complex co-dependencies among neurons and aims to force each neuron to learn more complex features [49].

In this project, a dropout layer with p = 0.3 is applied before the final classification layer. During each training batch, 30% of the neurons are randomly disabled. This enforces pattern recognition across multiple facial regions (e.g. eyes, mouth, eyebrows), reducing over-reliance on a single feature like mouth curvature alone for emotion detection.

03.9.5 Learning Rate

A crucial hyperparameter that defines the step size at which a model updates weights and biases during training. It directly influences both how quickly the model learns and how stable the learning process is [50].

When set too high, the model may apply overly large adjustments, causing it to overshoot optimal values. This can lead to instability, oscillations in the loss function, or failure to converge. Otherwise, a very low learning rate results in slow progress, as the model makes only minimal updates at each step. In such cases, it may take excessive time to converge or even become trapped in suboptimal configurations [50]

An appropriately chosen learning rate enables the model to progress toward optimal solutions efficiently and reliably—balancing speed and stability throughout the training process [50].

03.10 Evaluation Metrics for Classification Models

These tools help assess the model's accuracy across different emotion categories and offer insight into areas that may require improvement.

03.10.1 Relationships between Predicted and Actual Classes

When speaking about classification tasks, many performance metrics are derived from the relationships between predicted and actual class labels. To analyze a model's performance, it is common to organize outcomes into four fundamental categories [51]:

03.10.1.1 True Positive (TP):

This refers to instances where the model correctly predicts the positive class [51]. In FER, for example, a true positive occurs when the model predicts "happy" and the actual label is also "happy."

03.10.1.2 False Positive (FP);

This occurs when the model incorrectly predicts the positive class [51]. Continuing with the same example, a false positive would happen if the model predicts "happy" when the actual expression is "neutral."

03.10.1.3 True Negative (TN):

This represents cases where the model correctly predicts the negative class [51]. For instance, if the model predicts an expression is not "happy" and the true label is also not "happy," it counts as a true negative.

03.10.1.4 False Negative (FN):

This is when the model fails to identify the positive class, predicting a different category instead [51]. For example, a false negative occurs if the model predicts "sad" when the true expression is "happy."

These four categories form the basis for calculating essential classification metrics such as accuracy, precision, recall, F1-score, and others, which provide deeper insight into the model's strengths and weaknesses.

03.10.2 Confusion Matrix

One of the most used evaluation tools is the confusion matrix, which provides a summary of the classifier's predictions [52]. It is a tabular representation that shows how well the model performs by comparing predicted labels with actual ones: columns typically represent the predicted classes, while rows correspond to the true labels. This matrix not only reveals how many predictions were correct but also highlights where the model tends to confuse one emotion with another.

In the case of this project, the confusion matrix summarizes how the emotion recognition model performs across the six basic emotions proposed by Ekman [14]. Each row in the matrix corresponds to a true emotion label (e.g., happiness, sadness, anger), while each column indicates how often the model assigned that label. For example, if the model often misclassifies 'fear' as 'surprise,' the

confusion matrix will highlight that specific error—providing valuable feedback for model refinement.

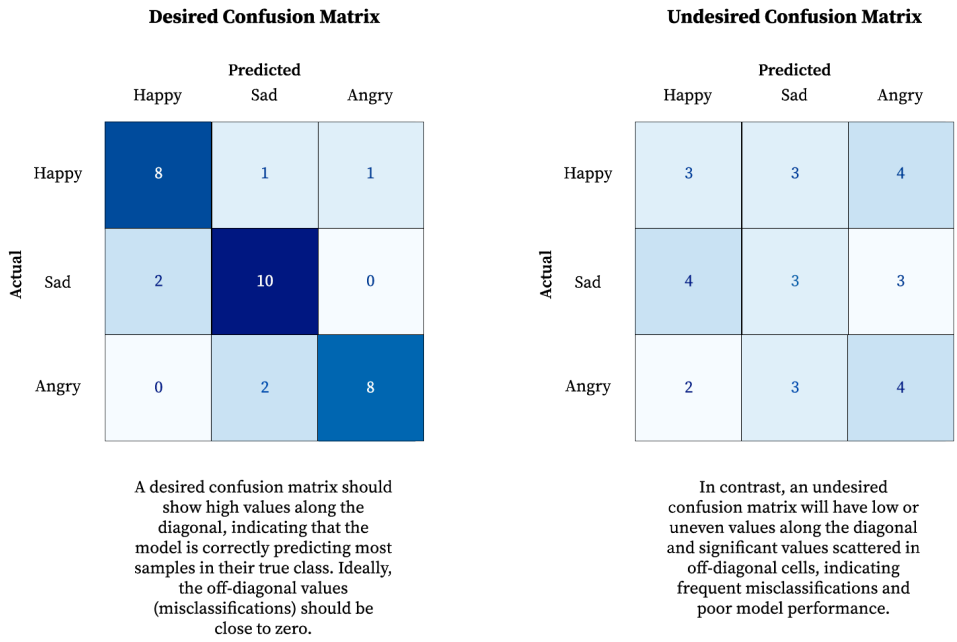

Figure 11 *Illustration of Ideal Vs. Problematic Confusion Matrix Patterns in Emotion Classification.*

(Left) Desired: Strong diagonal indicating correct classifications with minimal off-diagonal confusion. (Right) Undesired: Scattered predictions indicating poor class discrimination, particularly problematic between similar emotions like fear/surprise.

03.10.3 Accuracy and Error Rate

- Accuracy: The proportion of total correct predictions [31]:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad (2)$$

- Error Rate: The proportion of incorrect predictions.

$$Error\ Rate = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{P} + \mathrm{N}} \quad (3)$$

03.10.4 Recall

- Recall: Measures the model's ability to detect positive instances [31]:

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (4)$$

03.10.5 Precision and F1-Score

- Precision: The proportion of correctly predicted positive observations[31]:

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5)$$

- F1-Score: Harmonic mean of precision and recall [31]:

$$F1 = \frac{2 \ * \text{Precision} \ * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

03.10.6 Epoch-Based Curves

An epoch-based loss or accuracy curve, often referred to simply as a "training curve" [53] in DL literature, plots the value of the loss function and accuracy against the number of training epochs. This visualization allows researchers to [54]:

- Track model learning dynamics over time
- Detect signs of underfitting (high loss that does not decrease) or overfitting (training loss decreases while validation loss stagnates or increases)
- Make informed decisions about early stopping or adjustments to hyperparameters

The loss and accuracy curves typically include two separate plots:

- Training, which indicates how well the model fits the training data.
- Validation, which reflects generalization performance on unseen data.

A properly behaving curve shows both training and validation loss decreasing and converging as epochs increase. In the case of accuracy, this curve follows the same logic however opposite-wise, accuracy should be increasing at a steady rate. Divergence between the two—especially if validation loss rises while training loss falls—indicates overfitting.

03.10.7 Classification Report

A classification report is a standard evaluation tool in multiclass classification tasks that summarizes key performance metrics—precision, recall, f1-score, and support—for each class, along with overall averages [55].

Having reviewed the foundational principles that support the design and training of the current architecture —it becomes essential to introduce the tools and libraries that enabled their practical implementation. The following section outlines the key software components that supported dataset handling, model training, experiment tracking, and evaluation, forming the operational backbone of the system's development.

04. TOOLS AND LIBRARIES

Brief: *This section introduces the additional tools that, alongside Python, played a key role in the development process.*

04.1 NumPy

NumPy serves as the core library for numerical operations in Python. It provides an efficient N-dimensional array structure (ndarray) and a broad set of universal functions (ufuncs) that support tasks such as array manipulation, linear algebra, random number generation, and more[56]. It makes vectorized computation up to 50× faster than native Python lists [57]. NumPy is the backbone of scientific compute workflows, providing the core data structure and efficient primitives upon which deep learning frameworks are built.

04.2 Pandas

Pandas enhances NumPy by providing DataFrame and *Series* constructs designed for flexible and efficient handling of structured data [58]. A DataFrame represents two-dimensional, labeled data with support for column types, label-based alignment, powerful descriptive statistics and methods for quick data inspection. This makes it ideal for managing dataset metadata, class labels, experiment logs, and evaluation results in a tabular and human-readable format.

04.3 Scikit-Learn

Scikit-Learn provides a consistent interface and extensive library of tools for traditional machine learning tasks and data validation workflows [59]. Notably, it includes functions such as *train_test_split*, performance metrics like classification_report, and confusion_matrix for error analysis in classification.

04.4 PyTorch and timm

PyTorch is widely recognized as a leading tool for deep learning. One of its key features is a dynamic computation graph that is built in real time during forward propagation, allowing for smooth integration with Python-based debugging tools. This adaptability makes it particularly well-suited for research applications, rapid prototyping, and models that require input sizes to vary dynamically [60].

timm (“PyTorch Image Models”), created by Ross Wightman, is a comprehensive library that maintains state-of-the-art (SOTA) image model architectures [61], pretrained weights, and training utilities. In this project, timm provides ready-to-use pretrained backbones (e.g., ResNet, EfficientNet, MobileNet), and model utilities, streamlining benchmarking and enabling efficient experimentation without building architectures from scratch.

04.5 MLflow

MLflow is an open-source platform designed to simplify the entire machine learning lifecycle by gathering all the different steps into one unified process capable of experiment tracking, reproducible runs, model packaging, and deployment [62].

Key Features:

- Experiment Tracking: Possibility to log various parameters, metrics, code versions, and artifacts like model weights or plots. Enhances comparison and visualization of model performance across different runs [63].
- Reproducible Runs: Each execution (or "run") records metadata such as runtime, source code version, environment, and evaluation metrics, ensuring experiments can be replicated and audited [63].
- Model Packaging & Registry: Models can be saved in standard formats and versioned, assisting integration into production or deployment workflows [63].

These tools and libraries provided the necessary infrastructure to design, train, evaluate, and document the FER system presented in this work. From managing data and defining model architectures to logging experimental results and analyzing performance, each component contributed to building a reproducible and adaptable pipeline. With the technical groundwork in place, the next chapter presents the methodological strategy that guided the development of the system—from dataset construction to model selection, training, and evaluation.

05. DEVELOPMENT AND METHODOLOGY

Brief: *This chapter details the methodology followed to develop the emotion recognition system. It outlines the computational setup, training strategies, model selection process, and evaluation protocols, providing a clear view of how each component contributed to achieving reliable and efficient performance.*

05.1.1 Dataset

The effectiveness of a DL model is strongly influenced by the quality and representativeness of its training data. In this project, well-known facial expression datasets are combined with manually curated samples and AI-generated data to enhance emotional coverage and improve the model's robustness.

05.1.2 Dataset Creation

The dataset consists of 35,887 facial expression images distributed across seven emotional classes (Anger, Disgust, Fear, Happy, Neutral, Sad, and Surprise). Images were sourced from the following:

- Public Datasets: AffectNet [11], MMI [7], EFE [9], FERD [10] Dataset and Oulu-CASIA [8].
- Web-Sourced Data: Additional examples manually validated for emotion class correctness.
- AI-Synthesized Data: Generated using large-scale models (Gemini, Meta, Stable Diffusion, Copilot and ChatGPT) to diversify age, ethnicity, lighting, expression strength, and occlusion.

All images were normalized to a resolution of 256×256 pixels.

An exhaustive manual review was conducted to select suitable images from the various datasets used for training and evaluation. Several datasets—particularly AffectNet [11]—contained mislabeled or low-quality samples, including images categorized under the wrong emotion, as well as blank or empty images. As a result, only a limited number of carefully curated samples were selected from each dataset to ensure the integrity and reliability of the training process.

05.1.3 Image Resolution as a Design Criterion

A key constraint in the dataset curation process was the requirement for a minimum image resolution of 256 × 256 pixels. This criterion was applied consistently across all sources—public datasets, AI-generated samples, and real-world captures—to ensure compatibility with the envisioned deployment environments: webcams and smartphone cameras, which typically operate at much higher resolutions (e.g., 720p) than datasets such as RAF-DB (100×100 px) [64] or FER2013 (48×48 px) [65].

This decision was based on two main motivations:

- Real-World Alignment: The system is designed for practical use in spontaneous or stimulus-driven emotion detection scenarios using relatively modern consumer-grade devices. Low-

resolution datasets do not reflect the image quality captured by these more capable devices, thus creating a mismatch between training and deployment conditions.

- Model Compatibility: The baseline model selected for this project—ResNet-18—benefits significantly from higher-resolution inputs. Initial informal tests suggested improved learning stability and convergence when working with images above 200×200 px. Consequently, the entire pipeline was optimized for high-resolution input, rather than adapting the model to low-resolution legacy datasets.

This resolution threshold inevitably limited compatibility with some widely-used benchmark datasets. However, it also helped maintain consistency in visual quality and enhance the model's performance in the intended deployment scenarios.

For a more robust comparison with the broader literature, future work should consider:

- Downscaling the curated dataset to 100×100 and 48×48 to simulate compatibility with RAF-DB and FER2013.
- Training and testing the same ResNet-18 configuration under those conditions.

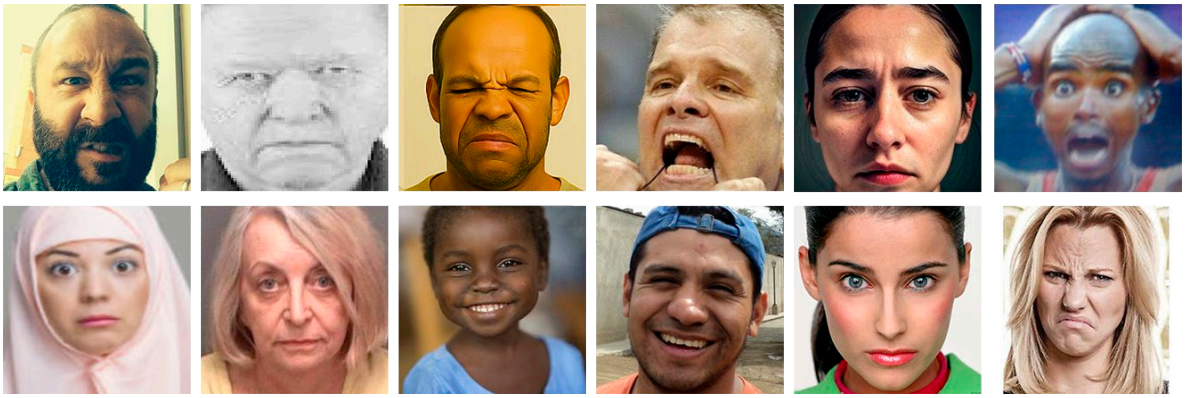

Figure 12 *Representative Samples from Curated Dataset Demonstrating Diversity in Age, Ethnicity, Lighting Conditions, And Expression Intensity.*

Images show both spontaneous and posed expressions from real-world and AI-synthesized sources.

As mentioned before, project adheres to the emotion classification framework established by Paul Ekman [14]. A neutral expression is also included in the dataset to serve as a baseline. Each of these emotional states is linked to specific facial muscle movements known as Action Units (AUs), as cataloged in Ekman's Facial Action Coding System (FACS) [66]. Below is a detailed explanation of each emotion included in the dataset:

- Anger: Anger is typically expressed through a lowered brow (AU4), glaring eyes, and tightened lips (AU23) [67][66]. This expression is characterized by tension in the face.

- Disgust: Disgust often manifests through a wrinkled nose (AU9), raised upper lip (AU10) [67] [66], and lowered eyebrows. It communicates revulsion or rejection.

- Fear: Fear is marked by raised eyebrows (AU1+2), widened eyes (AU5), and a slightly open mouth (AU25), often revealing clenched teeth [67] [66]. It reflects a sense of danger.

- Happiness: The most easily recognized emotion. It's usually shown when the corners of the mouth lift up, thanks to a muscle called the zygomatic major (AU12), which pulls the corners of the mouth upward (AU6), which causes "crow's feet" around the eyes [67][66].

- Sadness: Sadness is generally conveyed through a drooping upper eyelid (AU1), downturned corners of the mouth (AU15), and a slightly lowered head posture [67][66]. It indicates loss, disappointment, or helplessness.

- Surprise: Surprise involves raised eyebrows (AU1+2), widened eyes (AU5), and a dropped jaw (AU26) [67][66]. It reflects a reaction to sudden or unexpected stimuli, and unlike other emotions, is usually brief and quickly transitions into another emotional state.

- Neutral: A neutral expression represents the absence of any strong emotional display. It typically involves a relaxed facial posture, with no pronounced muscle activity. This category serves as a reference point for detecting deviations toward more emotionally expressive states.

Each emotion class in the dataset is visually validated to reflect these defining features, ensuring alignment with established psychological and physiological evidence.

05.1.4 Dataset Fairness Composition

To assess potential demographic bias in the model's predictions, a fairness analysis was conducted using a labeled subset of the test dataset. Specifically, 100 images were manually selected and annotated per emotion class (seven classes in total), yielding a total of 700 images. Each image was labeled with three demographic attributes based on visual inspection: gender (Female, Male), skin type (based on the Fitzpatrick scale [68] types adjusted to fit F1–F4), and age group (Young, Adult, Senior).

Table 2 *Demographic Distribution of the Fairness Evaluation Set.*

This table shows the demographic breakdown of a manually labeled subset of 700 images (100 per emotion class), categorized by gender (Female, Male), skin tone (F1–F4), and age group (Adult, Senior, Young). The table was used to support fairness analysis by evaluating whether the model's performance is consistent across diverse demographic segments.

Emotion	Total Images	Female (F)	Male (M)	F1	F2	F3	F4	Adult (A)	Senior (S)	Young (Y)
Anger	100	47	53	17	53	25	5	34	8	58
Disgust	100	51	49	36	39	18	7	35	8	57
Fear	100	57	43	44	31	11	14	37	7	56
Happy	100	58	42	26	26	25	23	20	6	74
Neutral	100	43	57	22	36	31	11	21	8	71
Sad	100	64	36	33	23	38	6	17	11	72
Surprise	100	52	48	31	33	13	23	20	3	77

The distribution across demographic categories revealed some notable imbalances. For instance, Sad and Happy classes showed a higher representation of female and young subjects, while Anger and Neutral had a more balanced gender distribution. In terms of skin tone, there was a general overrepresentation of lighter skin types (F1 and F2), particularly in Disgust and Fear, whereas darker tones (F3 and F4) were underrepresented across most classes. Age-wise, Young individuals dominated all classes, especially Surprise (77% Young) and Happy (74% Young), with Senior subjects appearing far less frequently overall.

These imbalances suggest that the current dataset may introduce demographic bias in the model's performance, potentially leading to unequal accuracy or confidence scores across groups. However, the actual impact on prediction quality across demographics has not yet been quantified and remains a critical next step.

In future iterations of this project, the following steps are recommended to strengthen fairness evaluations:

- Quantify model accuracy and confidence per demographic group.
- Augment underrepresented groups to balance training and test sets.
- Apply fairness-aware training or post-processing methods.

While these limitations do not invalidate the dataset's utility, they highlight important areas for improvement in future iterations—particularly for models intended for socially sensitive applications.

05.2 Model Definition and Comparison

05.2.1 Reasons behind ResNet18

Being one of the lighter variants of the ResNet [6] family, its choice was guided by the following considerations:

05.2.1.1 Transfer Learning Compatibility:

ResNet-18 is frequently used as a model in transfer learning [42] due to its proven effectiveness [69]. Its pre-trained weights developed using large-scale datasets like ImageNet [41], allow the model to leverage general low-level visual features—such as edges, textures, and shapes. This reuse of learned features greatly reduces the need for large quantities of labeled emotional data and supports efficient training for FER tasks while maintaining strong performance.

05.2.1.2 Computational Efficiency:

When compared to deeper architectures like ResNet-50 or ResNet-101, ResNet-18 strikes an effective balance between computational cost, model accuracy, and inference speed [70]. This efficiency makes it a practical choice for systems running on limited hardware resources, such as embedded devices or standard laptops, all while preserving reliable performance.

05.2.1.3 Modular and Adaptable Architecture:

The architecture is easy to adapt for FER by replacing the final fully connected (FC) layer [69], originally designed for 1,000 ImageNet classes [41], with a smaller output layer tailored to the seven emotion categories used in this study. This plug-and-play flexibility allows rapid experimentation and fine-tuning.

05.2.1.4 Empirical Reliability:

ResNet-18 has been extensively validated across computer vision tasks, including facial analysis and affective computing. Comparative studies [11] [26] have shown that ResNet variants perform robustly in emotion classification tasks, especially when combined with data augmentation and transfer learning [71].

05.2.2 Comparative Benchmarking

To contextualize the performance of ResNet-18, additional experiments were conducted using other compact state-of-the-art architectures:

05.2.2.1 EfficientNet-B0:

Known for its compound scaling strategy that balances depth, width, and resolution for optimized performance [12].

05.2.2.2 MobileNetV3:

Designed for mobile and edge devices, offering high accuracy at low computational cost [13].

Transformer-based models were excluded from experimentation due to their high computational demands, which exceeded the available hardware resources.

These models were accessed through the timm (PyTorch Image Models) library to ensure consistent preprocessing and training pipelines.

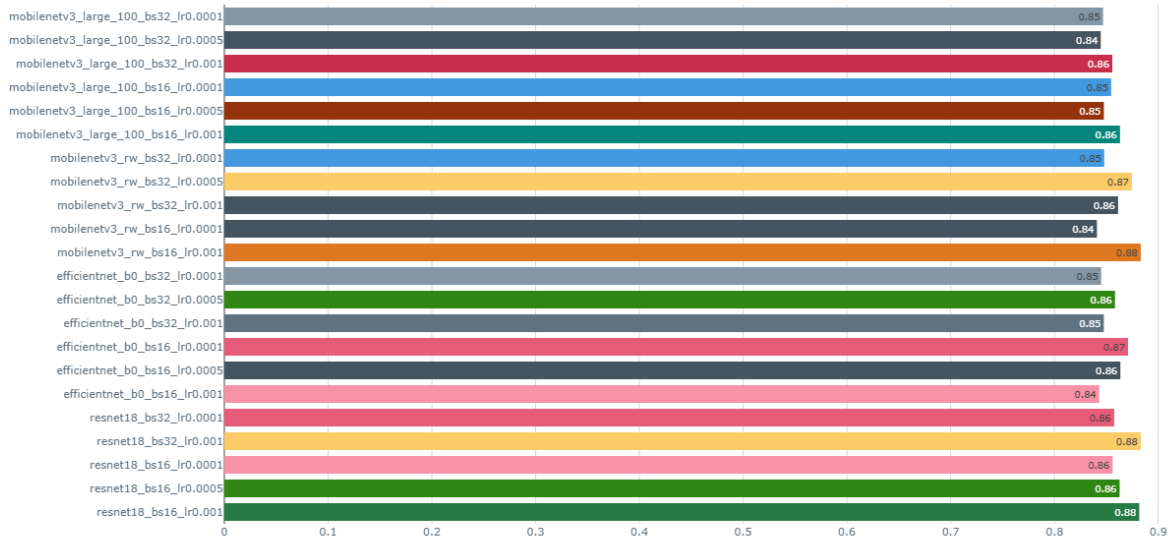

***Figure 13**. Preliminary Mlflow Experiment Tracking Interface Showing Systematic Hyperparameter Exploration Across 22 Training Runs. Including different batch sizes (16, 32) and learning rates.*

Dashboard displays validation accuracy, loss curves, and parameter combinations for different backbone architectures (ResNet-18, EfficientNet-B0, MobileNetV3), enabling reproducible model selection and performance comparison.

05.3 Training and Experimentation

This sub-section outlines the full training and fine-tuning process, beginning with the definition of the computational environment and its limitations, followed by a description of the tested models and hyperparameters, and concluding with an analysis of the experimental results and key observations.

05.3.1 Overview of Experimental Process

A total of 89 training runs were conducted and tracked using MLflow throughout the development of the system. These experiments were not conducted under a fixed benchmarking framework from the beginning; instead, they evolved iteratively, integrating increasing levels of control, logging, and model tuning.

Initial 75 runs focused on identifying general model performance under varying batch sizes and basic settings. Subsequently, data augmentation techniques were tested, followed by tuning of hyperparameters such as dropout rate and weight decay. Throughout this process, training stability and convergence were monitored via learning curves.

In the final stage, 14 targeted experiments were conducted specifically on ResNet-18 and EfficientNet-B0 using optimized augmentations and refined hyperparameter ranges. These runs produced the best-performing configuration, with ResNet-18 emerging as the most stable and robust model based on training dynamics and validation performance.

05.3.2 Hardware and Environments Used

Hardware:

- NVIDIA GTX 1650 Ti GPU (4 GB VRAM)
- Intel Core i7-10750H CPU
- 16 GB DDR4 RAM
- SSD 512 GB

Software Environment:

- Operating System: Windows 11
- Python 3.10
- CUDA 11.8, cuDNN 90100
- PyTorch 2.7.0+cu118
- timm 1.0.15
- Albumentations 2.0.8
- scikit-learn 1.6.1
- MLflow 2.22.1

Each model was trained using supervised learning, with 80/20 training-validation splits.

05.3.3 Objectives of the Experimentation

- To evaluate the performance of several lightweight CNN architectures
- To investigate how different parameters affect convergence, generalization, and training stability.

- To compare the performance of models trained with and without data augmentation techniques.
- To track and visualize training progress, as well as assess final model performance through diagnostic tools.

Each configuration was treated as a unique MLflow run, named using the following format for easy identification and grouping:

<backbone_name>_bs<batch_size>lr<learning_rate><aug|noaug>

05.3.4 Logged Parameters and Artifacts

For each training run, MLflow recorded the following:

Hyperparameters:

- Backbone architecture
- Batch size
- Weight decay
- Learning rate
- Dropout rate
- Whether data augmentation was enabled (boolean flag)

Performance Metrics:

- Accuracy and loss for both training and validation sets at each epoch
- Final validation accuracy and loss at the end of training
- Average inference time per batch on the validation set

05.3.5 Tested Experimentation Parameters

Table 3 *Summary of Training Configuration and Search Parameters.*

Overview of the experimental setup used for model selection. A total of 89 training runs were conducted, combining different backbone architectures, learning rates, weight decay values, dropout rates, and data augmentation strategies. The search was not exhaustive due to computational limitations, but sufficient diversity was achieved to observe performance trends and identify optimal hyperparameter configurations.

Parameter	Values / Options
Total training runs	89
Backbone architectures	resnet18, efficientnet_b0, mobilenetv3_rw, mobilenetv3_large_100, mobilenetv3_small
Batch sizes	32
Weight Decay	0.0, 4e-4, 5e-5, 5e-4, 6e-4
Learning rates	1e-3, 5e-4, 1e-4, 5e-5

Parameter	Values / Options
Data Augmentation	Enabled in some runs
Dropout Rate	0.00, 0.25, 0.28, 0.30, 0.32, 0.50
Epochs	Up to 30 (with early stopping)
Search Strategy	Not exhaustive due to computational constraints but multiple cross-validations among top candidates
Outcome	Sufficient variation to detect trends and select optimal model

05.3.6 Early Stopping

Early stopping was employed to prevent overfitting and preserve computational resources by exiting training once the validation performance stopped improving. While some studies use larger patience values (e.g., 10–20) [72], this work adopted a more conservative patience of 3, driven by limited computational capacity and strict time constraints. The effectiveness of this choice is supported by [72], who found that increasing patience values beyond 3 offered only marginal gains in validation accuracy, despite extending training time. As shown in their experiments

05.3.7 Evaluation Metrics

Each model was evaluated based on a set of metrics that provide both global performance and per-class diagnostic insight:

- Accuracy
- F1-Score (macro-averaged)
- Confusion Matrix
- Classification Report
- Training Curves
- Inference Time

05.3.8 Performance Visualizations

The following visual elements were generated during each training run to illustrate key results. The images presented below are included as representative examples.

05.3.8.1 Training And Validation Accuracy and Loss Curves Sample Graphs (Unsuccessful Experiment)

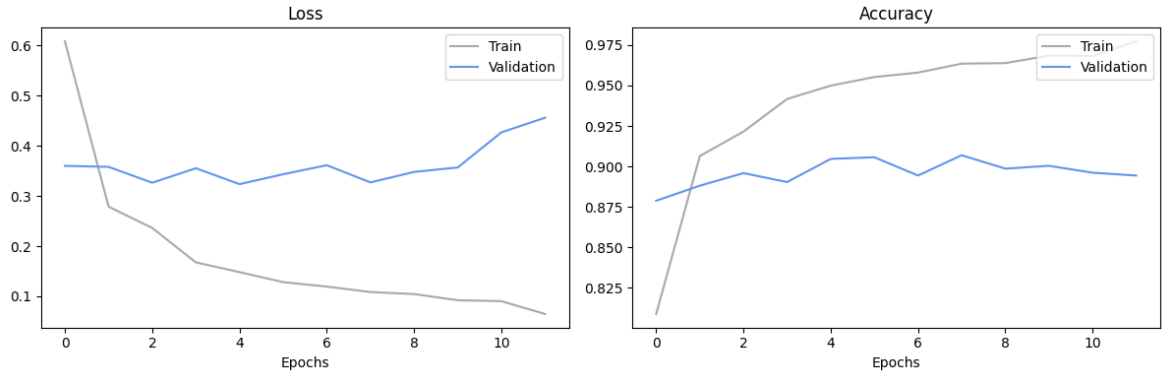

Figure 14 *Training Dynamics of an Unsuccessful Efficientnet-B0 Experiment (LR=0.0001) Showing Unstable Convergence.*

Note erratic validation accuracy fluctuations and diverging loss curves indicating overfitting. This behavior motivated the selection of ResNet-18 with more stable training characteristics.

05.3.8.2 Confusion Matrix

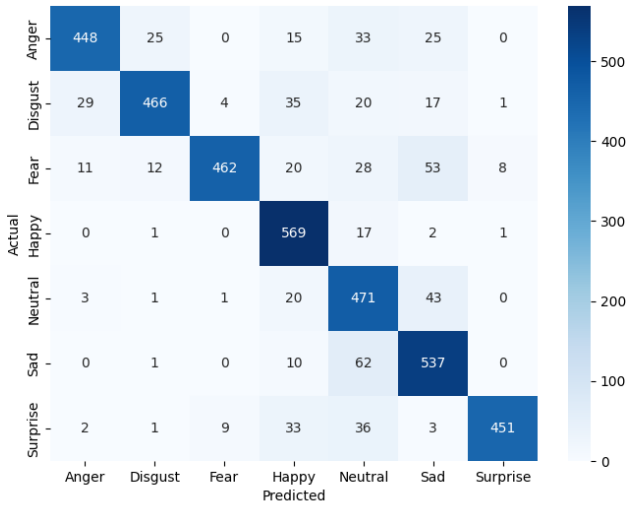

Figure 15 *Confusion Matrix for Mobilenetv3-Large Model Showing Per-Class Performance on Test Dataset.*

Strong diagonal indicates good overall performance (89.2% accuracy), with notable confusion between Surprise/Happy and Neutral/Sad, consistent with known challenges in FER literature.

05.3.8.3 Classification Report

Table 4 *Example of an EfficientNetB0 Classification Report on the Validation Set.*

Performance metrics for each emotion class, including precision, recall, and F1-score, based on a total of 3,986 validation samples. The overall accuracy reached 90%. Emotions like Happy and Surprise achieved the highest F1-scores

(0.96 and 0.95 respectively), while Neutral obtained the lowest (0.83), indicating variability in class-level performance. Macro and weighted averages were also 0.90, suggesting consistent generalization across classes.

Emotion	Precision	Recall	F1-score	Support
Angry	0.91	0.87	0.89	546
Disgust	0.89	0.91	0.90	572
Fear	0.90	0.90	0.90	594
Happy	0.95	0.97	0.96	590
Neutral	0.81	0.85	0.83	539
Sad	0.91	0.87	0.89	610
Surprise	0.95	0.96	0.95	535
Accuracy			**0.90**	**3986**
Macro avg	0.90	0.90	0.90	3986
Weighted avg	0.90	0.90	0.90	3986

05.3.8.4 Comparison of Average Validation Accuracy Across All Backbone Architectures

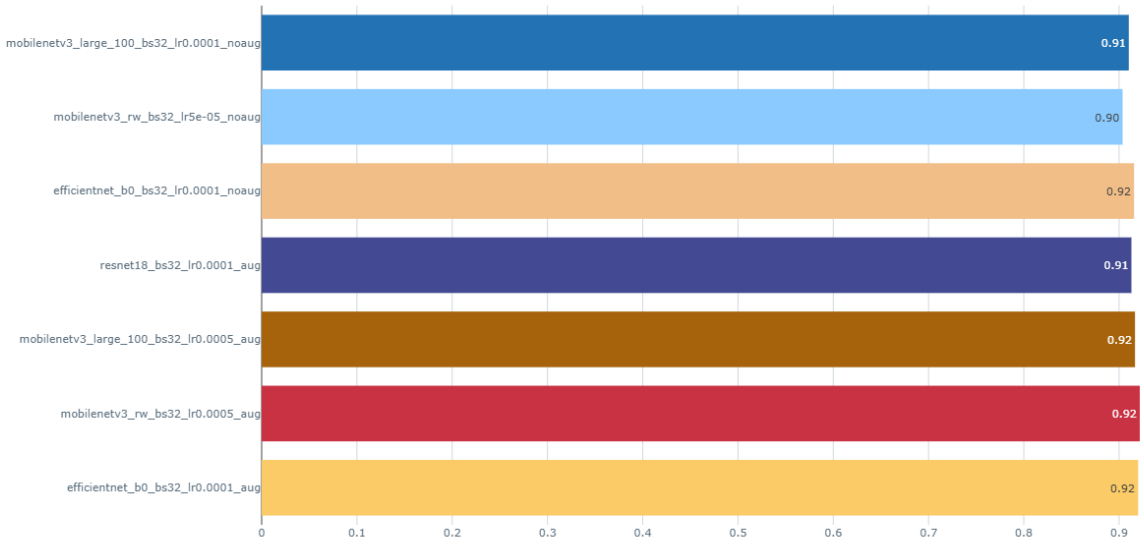

Figure 16 *An Example of a Comparative Performance Analysis Across Initial Top Seven CNN Architectures Averaged Over All Hyperparameter Configurations.*

In summary, this chapter has outlined the design and execution of an extensive development pipeline tailored to the constraints and goals of real-world FER. The strategic integration of diverse datasets, efficient architectures, and structured experimentation—supported by tools like MLflow—ensured that the selected models were not only accurate but also robust and practically deployable. The following chapter presents the results of this process, offering a comparative analysis and insights into model behavior under varying experimental conditions.

06. RESULTS AND DISCUSSION

Brief: *This section presents the evaluation results obtained from the MLflow experiments conducted using different backbone architectures. It sequentially reviews the standard performance metrics for each model and highlights key insights derived from their behavior. Finally, the best-performing model is identified and discussed in detail.*

06.1 Observations from Early and Intermediate Runs (75 Total Runs)

06.1.1 Data Augmentation

The following table was generated using pandas by analyzing the main CSV file containing the first 75 training runs. These experiments included basic data augmentation techniques—horizontal flip, random rotation, coarse dropout, and brightness variation—as well as dropout rates of 0.0, 0.3, and 0.5; learning rates of 1e-3, 5e-4, 1e-4, and 5e-5; and weight decay values of 0.0, 5e-4, and 6e-4.

Table 5 *Summary of Data Augmentation Impact on Model Performance.*

Comparison between training runs with and without data augmentation, based on averaged validation accuracy, loss, and training duration across all experiments.

Data Augmentation	Final Val Accuracy (Mean)	Final Val Accuracy (Min)	Final Val Accuracy (Max)	Final Val Loss (Mean)	Final Val Loss (Min)	Final Val Loss (Max)	Duration (min, Mean)	Duration (min, Min)	Duration (min, Max)
No	0.8944	0.8539	0.9157	0.4447	0.3706	0.5289	78.47	32.7	174.0
Yes	0.9091	0.8138	0.9235	0.3383	0.2626	0.5585	76.84	21.5	216.0

06.1.1.1 Validation Accuracy

- The mean validation accuracy is slightly higher when augmentation is used (0.9091 vs 0.8944).
- The maximum accuracy achieved is also higher with augmentation (0.9235 vs 0.9157).
- However, the minimum accuracy is a bit lower with augmentation (0.8138 vs 0.8539), indicating more variability.

06.1.1.2 Validation Loss

- The mean validation loss is lower with augmentation (0.3383 vs 0.4447), which suggests better generalization.
- The minimum loss is also lower with augmentation (0.2626 vs 0.3706).

06.1.1.3 Training Duration

- The mean training duration is lower with augmentation (76.84 min vs 78.47 min).

06.1.1.4 Summary

Early comparisons between models trained with and without data augmentation indicated a consistent improvement in generalization with augmentation applied. As a result, almost all further experiments were conducted with augmentations enabled.

06.1.2 Accuracy across different architectures

Average validation accuracy was foremost steady across model architectures. As shown in the below Figure, ResNet18 slightly outperformed more compact models such as Mobilenetv3_small. The higher capacity of ResNet-18 likely contributed to its superior ability to learn complex emotion representations, especially when combined with weight decay and dropout.

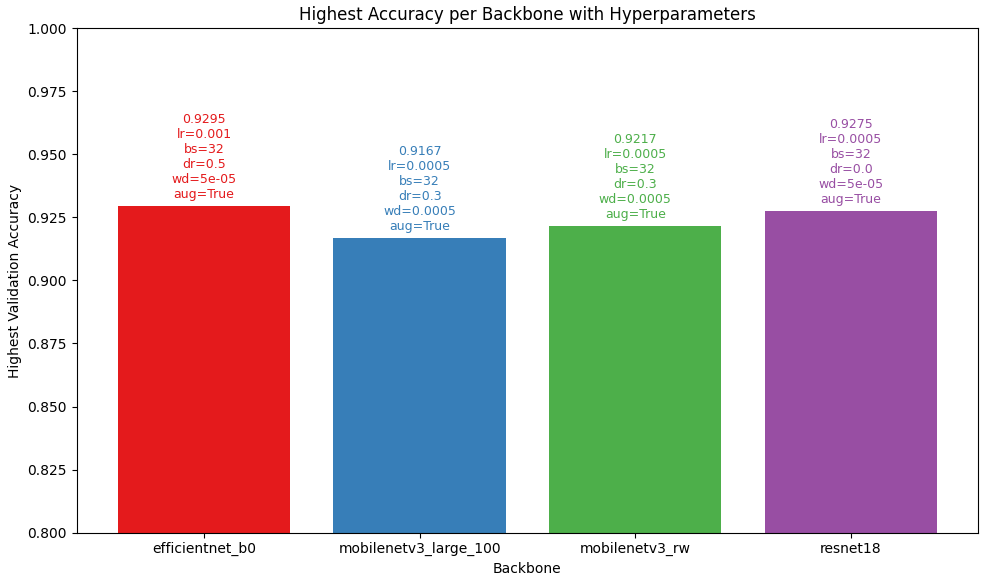

Figure 17 *Peak validation accuracy achieved by optimal hyperparameter configuration for each architecture during the first run rounds.*

EfficientNet-B0 reaches highest single performance (92.95%) but ResNet-18 shows more consistent results across runs. Bars represent single best run.

06.1.3 Statistical Summary

Table 6 *Accuracy Comparison Across Backbone Architectures during Initial 75 Runs.*

Summary of the performance of four backbone architectures evaluated during experimentation.

Backbone	Mean Accuracy	Min Accuracy	Max Accuracy
EfficientNet-B0	0.907677	0.854742	0.929503
MobileNetV3-Large-100	0.899685	0.853989	0.916708
MobileNetV3-RW	0.903067	0.875815	0.921726
ResNet18	0.908192	0.813848	0.927496

Based on Table 6, ResNet-18 achieved the highest mean accuracy (0.9082), closely followed by EfficientNet-B0 (0.9077). While EfficientNet-B0 demonstrated a slightly higher peak performance (maximum accuracy of 0.9295), ResNet-18 stood out as a strong and consistent baseline. Notably, ResNet-18 also exhibited the lowest minimum accuracy (0.8138), suggesting a higher sensitivity to specific training configurations. Nevertheless, its strong average performance reinforces its robustness across runs. However, a critical aspect to assess is whether these accuracy values resulted from stable and generalizable learning behavior, or whether they were influenced by erratic and unreliable training dynamics.

06.1.4 Learning Rate

Learning rate played a critical role in the training performance across different experiments. Its influence extended beyond final accuracy, significantly affecting the stability and smoothness of both loss and accuracy curves. Higher learning rates often led to unstable training or premature convergence, while appropriately lower rates resulted in more gradual and consistent learning. Figures 18–20 illustrate the distinct differences in accuracy progression and training dynamics resulting from various learning rate configurations.

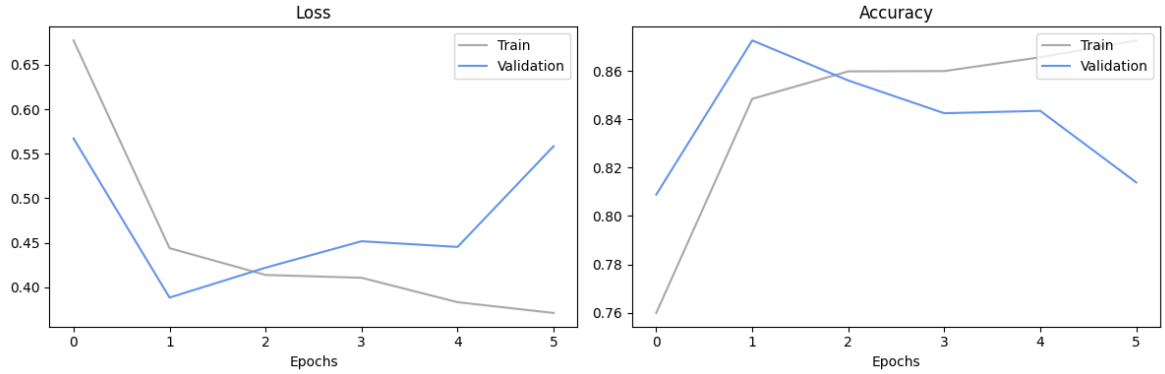

Figure 18 *Loss and Accuracy Curves for ResNet-18 with Learning Rate = 0.001.*

Training and validation loss and accuracy over six epochs for the ResNet-18 model using a learning rate of 0.001. Validation performance starts to decline after epoch 2, suggesting early signs of overfitting.

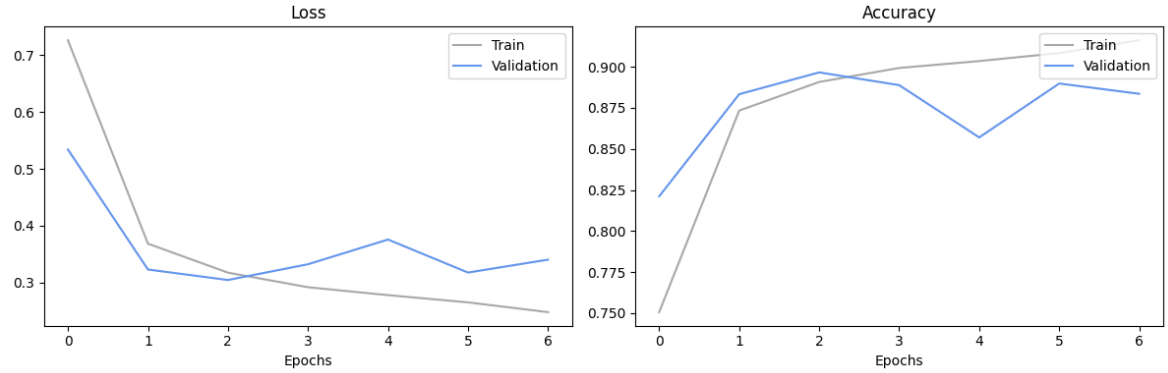

Figure 19 *Loss and Accuracy Curves for ResNet-18 with Learning Rate = 0.0005.*

The model exhibits stable learning and high accuracy; however, slight fluctuations in validation loss and accuracy suggest moderate overfitting starting after epoch 3.

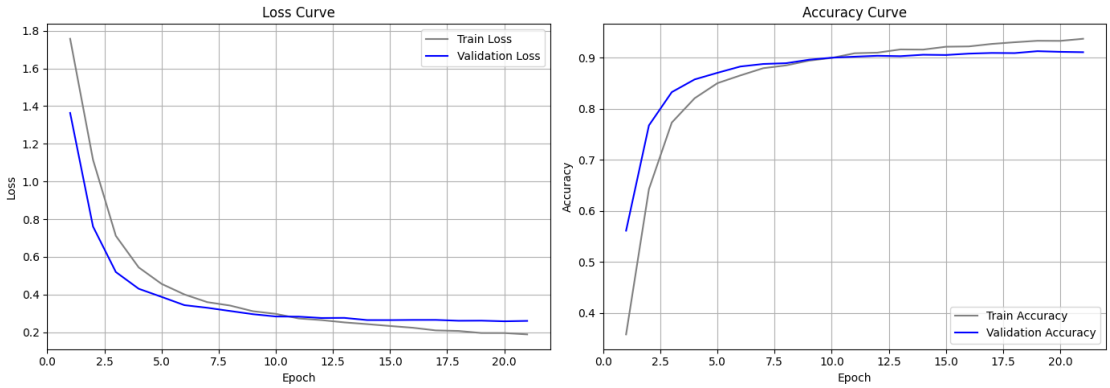

Figure 20 *Loss and Accuracy Curves for ResNet-18 with Learning Rate = 0.00005.*

This figure illustrates the training and validation performance of ResNet-18 across 22 epochs using a learning rate of 0.00005. Both loss and accuracy curves show steady and consistent improvement, with minimal gap between training and validation performance, indicating strong generalization and no signs of strong overfitting throughout the training.

It is important to note that, although EfficientNet achieves one of the highest accuracy scores as shown in Figure below, its loss and accuracy curves reveal a notably plateaued learning behavior.

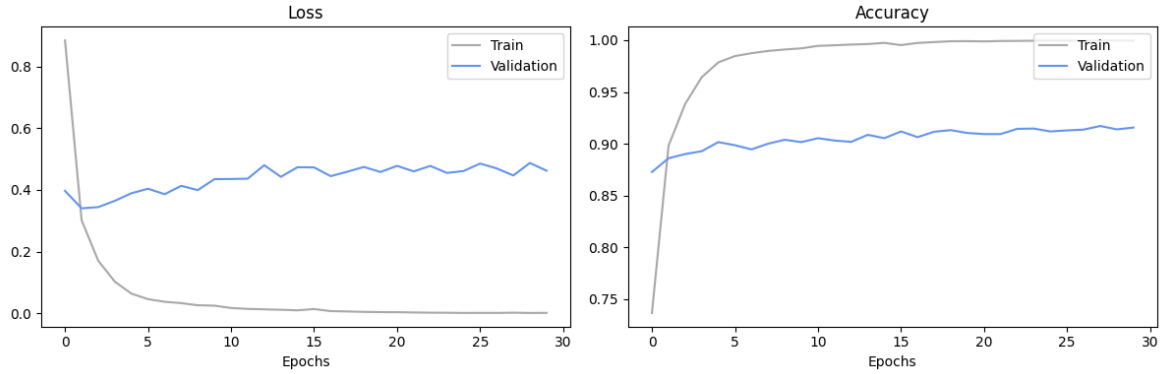

Figure 21 *Training dynamics of an unsuccessful EfficientNet-B0 experiment (LR=0.0001) showing unstable convergence.*

Note erratic validation accuracy fluctuations and diverging loss curves indicating overfitting after epoch 5. This behavior motivated the selection of ResNet-18 with more stable training characteristics.

Due to constraints in time and computational resources, no further experiments were conducted to improve the learning stability of the *EfficientNet* model. However, it is hypothesized that applying a data shuffling strategy during training could enhance the model's ability to generalize by exposing it to a more diverse sequence of samples across epochs.

06.1.5 Evaluating Backbone's Actual Generalization Considering Overfitting

Figure 22 shows the top two accuracy scores per backbone along with their corresponding training curves. Analyzing the learning dynamics of these top-performing configurations reveals consistent trends across architectures, allowing for the following conclusions and recommendations:

06.1.5.1 High Training Accuracy vs. Validation Stability

All models exhibit very high training accuracy (>0.90), with EfficientNet-B0 and ResNet18 reaching the highest levels. However, validation accuracy tends to plateau early and fluctuates significantly, indicating a tendency toward overfitting across all configurations.

06.1.5.2 Early Overfitting

Most models begin overfitting within the initial epochs, suggesting that generalization remains a challenge. This emphasizes the need to refine regularization strategies to ensure more stable convergence during training.

06.1.5.3 Slightly Promising ResNet-18 Configuration

The ResNet-18 configuration using a learning rate of 0.0005, weight decay of 0.005, and no dropout (DR = 0.0) shows weak cues of steady convergence. It may be worth further exploration with dropout and similar regularization parameters to determine whether this model can improve generalization without overfitting.

06.1.5.4 Regularization and Hyperparameter Impact:

Models with higher dropout (dr=0.5) and modest weight decay (e.g., 5e-5) performed better indicating what could be a trend in better regularization. Learning rates of 0.0005 seem promising across models.

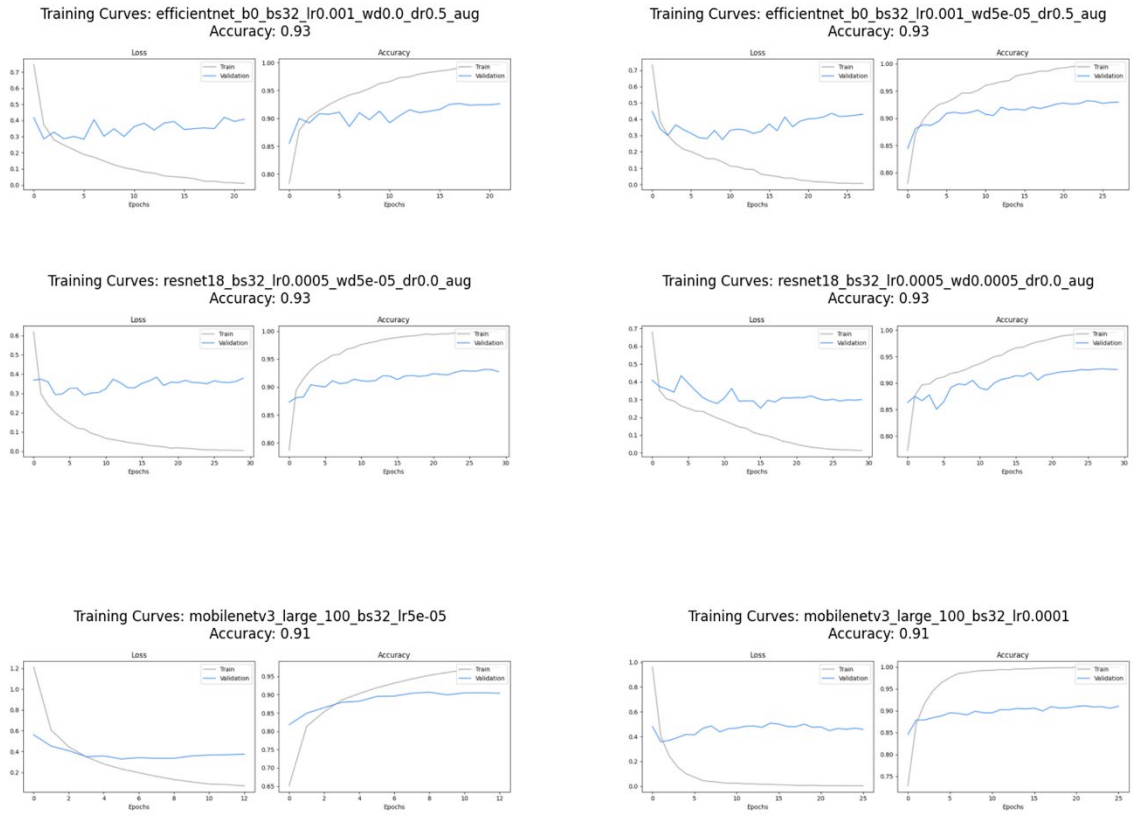

Figure 22 *Top Two Accuracy Scores per Backbone and Corresponding Training Curves during first 75 Runs.*

Training and validation loss and accuracy curves for the two best-performing runs of each backbone architecture: EfficientNet-B0, ResNet-18, and MobileNetV3-Large-100. All models shown achieved at least 0.91 accuracy, although most curves display varying degrees of overfitting, with validation curves plateauing or diverging after several epochs. This highlights the need for further regularization or early stopping to enhance generalization.

06.1.5.5 Next Steps:

06.1.5.5.1 Regularization Tuning:

Further fine-tune dropout and weight decay to narrow the gap between training and validation metrics.

06.1.5.5.2 Early Stopping and Data Augmentation:

Consider stricter early stopping criteria or more diverse augmentation to reduce overfitting and improve generalization.

06.1.6 Refining the ResNet-18 Backbone: Signs of Promising Generalization

Based on the conclusions drawn from the top-performing configurations, 14 additional experiments were conducted to further investigate ResNet-18 and EfficientNet—the two best-performing architectures—using narrower hyperparameter settings. Specifically, configurations with the same learning rate (LR = 5e-5), but with different dropout rates and weight decays, were analyzed to assess their influence on model behavior. These runs introduced a refined data augmentation pipeline as well.

06.1.7 Enhanced Data Augmentation

Initial experiments used a relatively simple augmentation setup, consisting of random rotation, horizontal flipping, coarse dropout, and brightness-contrast adjustments:

```
A.Compose([
    A.Resize(cfg["image_size"], cfg["image_size"]),
    A.Rotate(p=0.6, limit=[-45, 45]),
    A.HorizontalFlip(p=0.6),
    A.CoarseDropout(p=0.3),
    A.RandomBrightnessContrast(),
    ToTensorV2()
])
```

Figure 23 *Data Augmentation Pipeline Used in Initial 75 Training Runs.*

This augmentation setup, applied during the first 75 experiments, includes resizing to the target resolution followed by random rotation (up to ±45°), horizontal flipping, coarse dropout, and brightness/contrast adjustments.

In contrast, the final 14 runs incorporated a more robust augmentation pipeline. The updated augmentations included slight affine transformations, color jittering, Gaussian blur, and reduced dropout aggressiveness:

```
A.Compose([
    A.Resize(cfg["image_size"], cfg["image_size"]),
    A.Rotate(p=0.5, limit=[-20, 20]),
    A.HorizontalFlip(p=0.5),
    A.Affine(
        scale=(0.95, 1.05),
        translate_percent={"x": (-0.05, 0.05), "y": (-0.05, 0.05)},
        rotate=(-10, 10),
        p=0.3
    ),
    A.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1,
p=0.3),
    A.CoarseDropout(p=0.2),
    A.RandomBrightnessContrast(p=0.3),
    A.GaussianBlur(blur_limit=(3, 7), p=0.1),
    ToTensorV2()
])
```

Figure 24 *Enhanced Data Augmentation Pipeline Used in the Final 14 Training Runs.*

This configuration was employed in the last 14 experiments and introduces a more diverse and controlled transformation strategy compared to the initial setup. It includes milder rotations (±20°), affine transformations with scaling and translation, controlled color jitter, Gaussian blur, and adjusted probabilities for dropout and contrast changes. The goal was to simulate more realistic variations while preserving facial structure, thus promoting better generalization.

06.1.8 Focused Hyperparameter Tuning

Alongside augmentation improvements, a grid search was conducted using the following values:

- Weight decay: {4e-4, 5e-4, 6e-4}
- Dropout rate: {0.25, 0.28, 0.35}

06.1.9 Final Optimization Results

This subsection provides a detailed review of the training curves, classification reports, and experimental logs from the final round of optimization experiments tracked. The objective is to analyze the learning dynamics of each configuration to better understand the factors that contributed to generalization improvements.

06.1.9.1 EfficientNet-B0: Persistent Overfitting and Future Adjustments

Three experiments were conducted using EfficientNet-B0. Despite using the same data augmentation pipeline as in the ResNet-18 experiments, all three runs exhibited consistent overfitting. In each case, training accuracy increased steadily while validation accuracy plateaued or declined early in training.

One possible explanation lies in the use of a relatively high learning rate or insufficient regularization. A potential remedy would involve decreasing the learning rate and increasing dropout or weight decay.

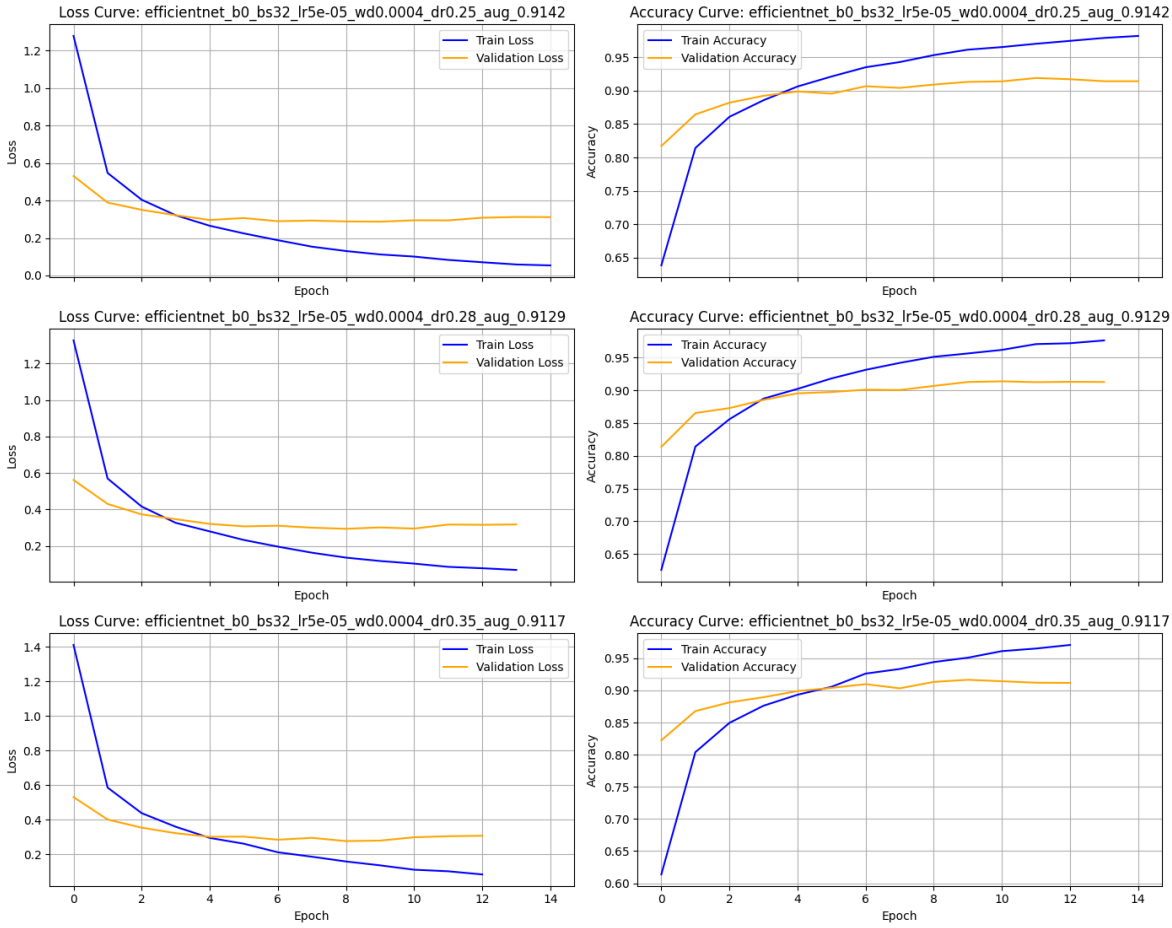

Figure 25 *Training and Validation Curves for EfficientNet-B0 Across Three Dropout Configurations (0.25, 0.28, 0.35), using the same Learning Rate (5e-5), Weight Decay (0.0004), and Batch Size (32).*

Although all runs showed stable training and convergence during the initial epochs, by the final stages all models exhibited signs of overfitting—validation accuracy plateaued while training accuracy continued to increase, and validation loss stopped improving at the same rate as training loss. This divergence indicates a limited capacity to generalize beyond the training data. Among the runs, the model with a dropout rate of 0.25 reached the highest validation accuracy (91.42%).

Additionally, longer warm-up schedules or early stopping criteria could be tested to prevent premature overfitting.

Future experiments may benefit from:

- Reducing the learning rate.
- Increasing dropout beyond 0.35.
- Exploring label smoothing or stronger data augmentation.
- Fine-tuning fewer layers or freezing earlier blocks during initial epochs.

While EfficientNet-B0 has strong performance potential, these findings indicate that further tuning is required to make the most of its capacity without compromising generalization.

06.1.9.2 ResNet-18: Regularization and Data Augmentation Synergy

A total of 10 training runs were conducted using ResNet-18, exploring the combinations of weight decay (WD ∈ {4e-4, 5e-4, 6e-4}) and dropout rate (DR ∈ {0.25, 0.28, 0.35}). Among these, only one experiment was run without any data augmentation. Notably, this run was the only one to exhibit clear signs of overfitting, as evidenced by a rapid divergence between training and validation accuracy after a few epochs. The training loss continued to decrease while the validation performance deteriorated, suggesting poor generalization capacity in the absence of augmentation.

In contrast, all other runs that included data augmentation demonstrated improved convergence and generalization. For example, comparing two runs with identical hyperparameters (WD = 4e-4, DR = 0.25) but differing only in the use of augmentation reveals that the augmented version achieved smoother convergence and better alignment between training and validation curves. While some separation occurred toward the end, the model exhibited a more stable learning trajectory overall.

Within the augmented runs, the best-performing configurations were those using intermediate values of both weight decay and dropout rate. Runs with WD = 5e-4 and DR = 0.28 showed balanced training dynamics, with training and validation curves evolving in tandem and validation accuracy consistently improving. This suggests an effective trade-off between underfitting and overfitting.

Based on these observations, two additional experiments were conducted to further refine the dropout rate, while keeping the weight decay fixed. The selected dropout values (DR ∈ {0.30, 0.32}) were chosen as potential sweet spots for regularization. These configurations were included in the final model comparison; however, neither outperformed the previous best configuration using a dropout rate of 0.28 and a weight decay of 0.0005. As a result, no further tuning was pursued along this parameter dimension.

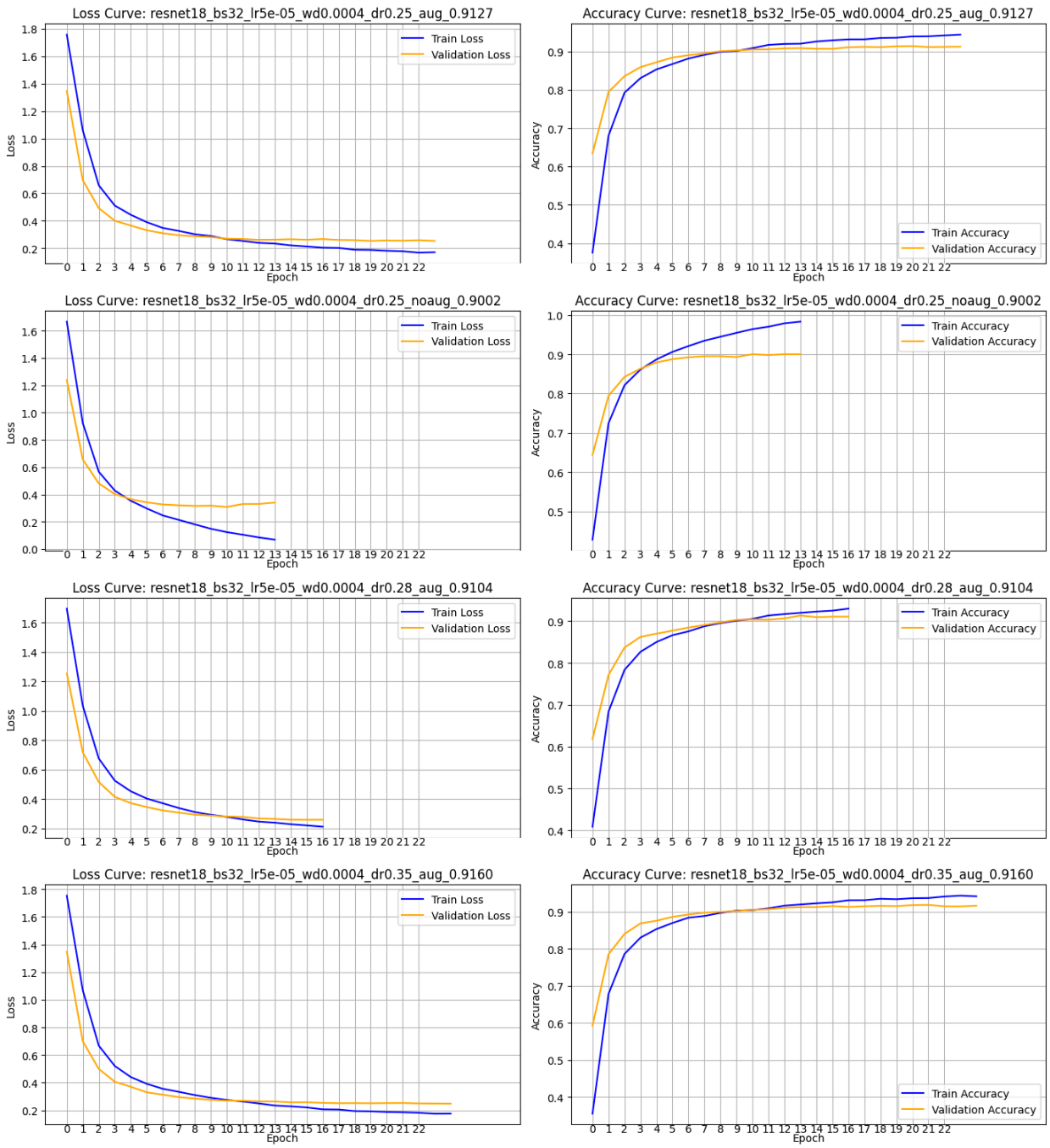

Figure 26 *Training and Validation Curves for Resnet-18 Under Varying Dropout Rates (0.225, 0.28, 0.35) with Fixed Weight Decay (0.004).*

The run without data augmentation exhibits clear signs of early and escalating overfitting, with a widening gap between training and validation accuracy over epochs. In contrast, the augmented runs demonstrate stable learning behavior, with minimal discrepancies between training and validation performance. This suggests that moderate dropout values combined with augmentation contribute to improved generalization.

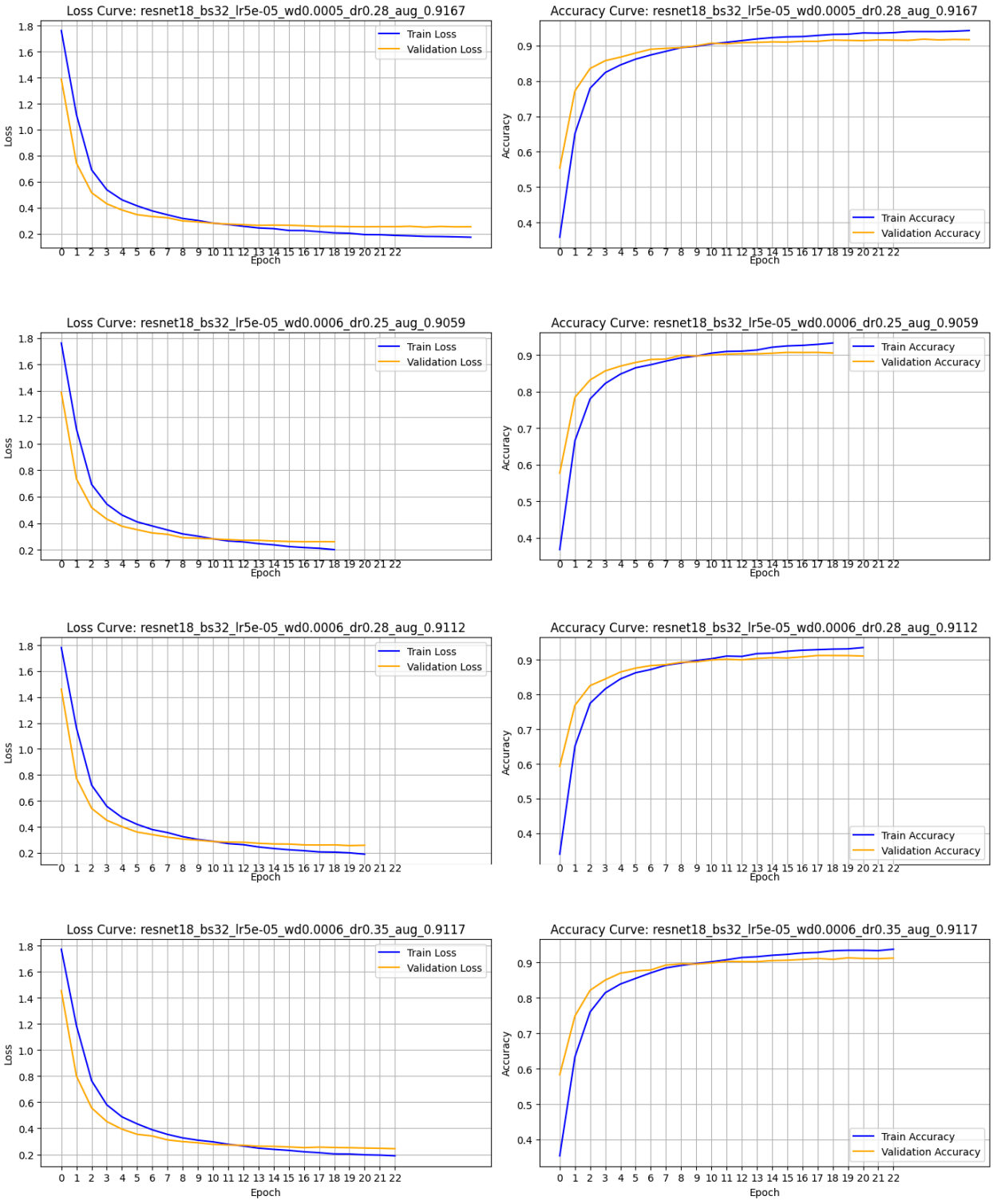

Figure 27 *Training and validation curves for ResNet-18 under different dropout rates (0.25, 0.28, 0.35) and weight decay values (0.0005, 0.0006), with fixed batch size (32) and learning rate (5e-5).*

All runs show stable learning trajectories with minor fluctuations, indicating consistent convergence behavior. Among them, the configuration with dropout rate of 0.28 and weight decay of 0.0005 achieved the best overall performance, reaching a validation accuracy of 91.67%, thus suggesting an optimal regularization balance for this setting.

06.2 Best Performing Model

After evaluating dozens of configurations, the best-performing model was found to be based on the ResNet-18 architecture, fine-tuned with the aforementioned data augmentation and dropout regularization.

This configuration provided a good balance between accuracy and model efficiency, demonstrating strong performance across all seven universal emotion classes.

06.2.1 Selected Configuration

***Table** 7 Best-Performance Model Specific Hyperparameters.*

This table outlines the exact hyperparameter configuration used in the highest-performing model of the study. The best result was achieved using ResNet-18 as the backbone with a batch size of 32, a learning rate of 5e-5, weight decay of 5e-4, and dropout rate of 0.28. Training employed early stopping with a patience of 3 epochs and data augmentation enabled, using the Adam optimizer.

Hyperparameter	Value
Backbone	ResNet-18
Batch Size	32
Learning Rate	5e-5
Weight Decay	5e-4
Dropout Rate	0.28
Epochs	30 (early stopping w/ patience = 3)
Data Augmentation	Enabled
Optimizer	Adam

06.2.2 Accuracy on Test Dataset

This model achieved a test accuracy of 92%, indicating a strong ability to generalize to unseen facial expression data. The high accuracy underscores the effectiveness of the data curation process, the quality of the augmentation strategies, and the suitability of the ResNet-18 architecture for the FER task when applied to a carefully curated and diverse dataset.

It is important to emphasize that this level of performance was attained under resource constraints using a relatively lightweight architecture, which further supports the model's applicability to real-world environments where computational resources may be limited.

06.2.3 Confusion Matrix

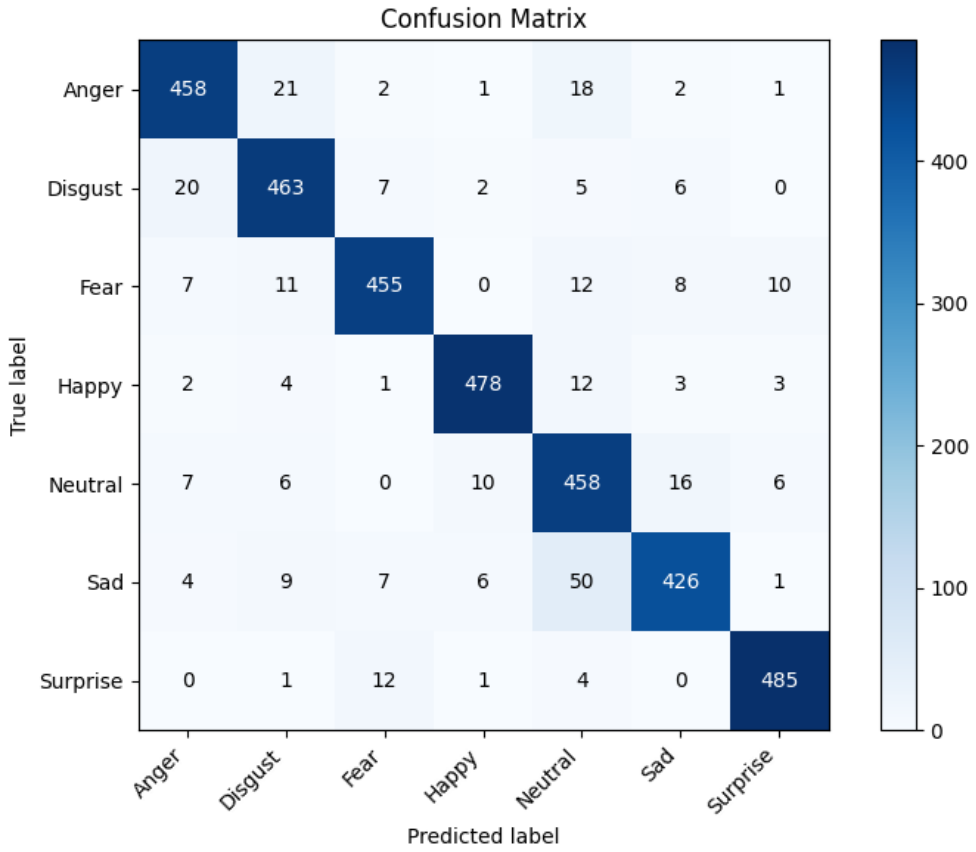

Figure 28 *Best-Performance Confusion Matrix on Test Dataset.*

Confusion matrix illustrating the classification performance of the best-performing model on the test dataset. The model demonstrated high precision and recall across most emotion categories, with the majority of misclassifications occurring between semantically or visually similar expressions, such as sadness and neutral, or fear and surprise.

The confusion matrix for the best model, shown above, provides a granular analysis of the model's classification behavior across the seven target emotion classes. The model demonstrates strong per-class performance with especially high diagonal values, indicating that most predictions matched the true labels.

Key insights include:

- Happy and Surprise emotions were recognized with high reliability, with minimal confusion with other categories.
- Neutral was occasionally misclassified as Sad or Fear, a common challenge in FER tasks due to the subtle facial cues shared between low-arousal emotional states.
- Disgust and Anger also showed slight overlap, but with high overall precision and recall.

These patterns are consistent with findings in existing FER literature, where neutral and ambiguous emotions often exhibit more inter-class confusion [16].

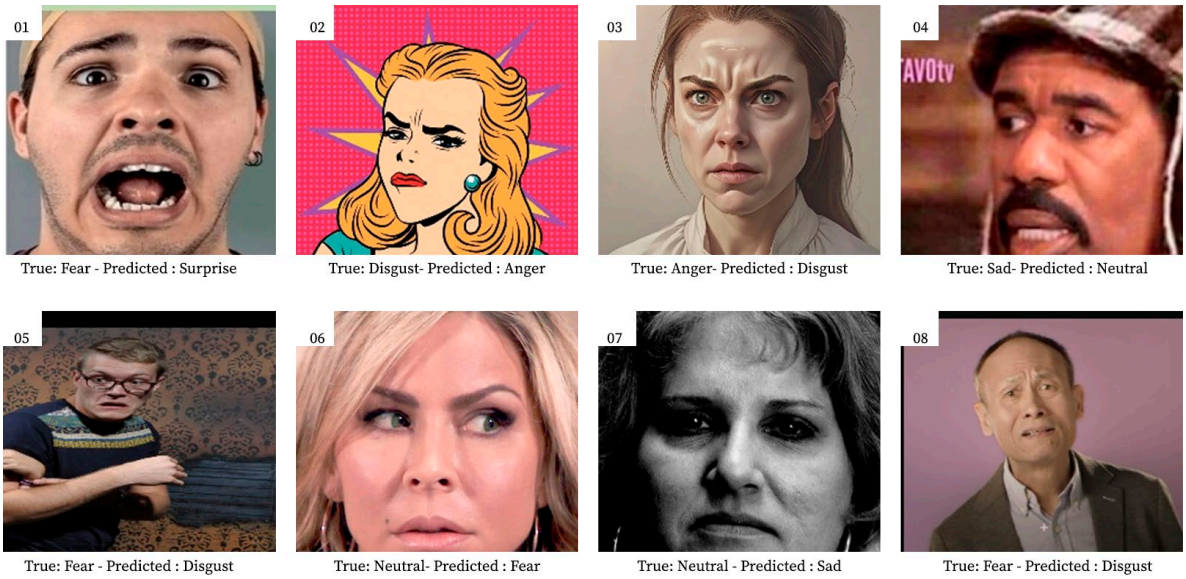

Figure 29 *Sample Images of Incorrect Predictions within Test Dataset.*

Representative examples of misclassified facial expressions by the model. These cases highlight typical confusion patterns such as fear being mistaken for surprise or disgust, and neutral expressions being predicted as fear or sadness. The image set includes diverse demographics and expression intensities, reflecting challenges in generalization and underscoring the need for enhanced representation and discrimination across subtle emotional cues.

Upon examining the sample images of incorrect predictions, it becomes evident that some cases—such as Image 07 and Image 08—would likely be challenging even for a human to classify accurately. Additionally, there is a noticeable pattern of confusion between the emotions of disgust and anger, with genuine disgust expressions misclassified as anger and vice versa, as seen in Images 02 and 03. Lastly, certain instances, such as Image 04, suggest that the label assigned as the ground truth may be questionable, and the model's prediction could in fact be more accurate.

06.2.4 Classification Report

Table 8 *Classification Report from Best-Performing Model on Test Dataset.*

Performance metrics including precision, recall, and F1-score for each emotion category based on the final evaluation of the best-performing model. The model demonstrates particularly strong performance on emotions such as happy, surprise, and anger, while slightly lower scores on neutral and sad highlight areas for improvement. Both macro and weighted averages indicate an overall balanced and robust classification capability across classes.

Emotion	Precision	Recall	F1-Score	Support
Anger	0.92	0.91	0.92	503
Disgust	0.90	0.92	0.91	503
Fear	0.94	0.90	0.92	503
Happy	0.96	0.95	0.96	503

Neutral	0.82	0.91	0.86	503
Sad	0.92	0.85	0.88	503
Surprise	0.96	0.96	0.96	503
Macro Average	**0.92**	**0.92**	**0.92**	**3521**
Weighted Average	**0.92**	**0.92**	**0.92**	**3521**

The fine-tuned ResNet-18 model achieved consistently strong performance across all seven basic emotions, with an overall accuracy of 92%. Emotions such as Happy and Surprise reached the highest F1-scores (0.96), indicating the model's excellent ability to detect high-arousal facial expressions. Meanwhile, more subtle emotions like Neutral and Sad showed slightly lower F1-scores, which aligns with known challenges in differentiating low-intensity expressions. Nevertheless, the macro-averaged F1-score of 0.92 confirms the model's balanced and robust classification performance, making it a promising candidate for real-world emotion recognition applications.

06.2.5 Accuracy and Loss Curves

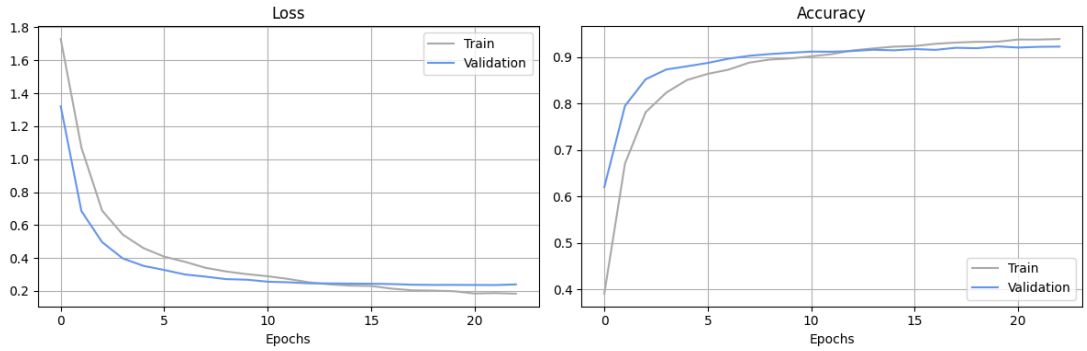

Figure 30 *Best-Performance Model Loss and Accuracy Curves.*

Training and validation loss and accuracy curves for the best-performing model. The model exhibits a fast convergence during the first epochs, followed by stable and parallel curves with minimal gap between training and validation performance. This indicates strong generalization and minimal overfitting.

The training and validation curves for accuracy and loss across epochs are shown in Figure above. These plots demonstrate a smooth and stable training process:

- Loss Curves: Both training and validation loss decreased steadily throughout the training process, eventually plateauing near the end, which indicates convergence and minimal overfitting.
- Accuracy Curves: A consistent improvement in both training and validation accuracy is observed across epochs, with the validation accuracy closely tracking the training curve, suggesting that the model retained generalization capacity during training.

Together, these curves validate the selection of hyperparameters (e.g., cosine annealing learning rate scheduler, batch size, weight decay) and confirm that early stopping was not prematurely triggered, as the model continued to improve up to the final epochs.

06.2.6 Statistical Validation

To assess consistency, the final ResNet-18 model was trained under three different random seeds. Results were as follows:

Table 9 *Statistical Validation Results Across Random Seeds.*

Accuracy scores obtained from three independent training runs of the final ResNet-18 model using different random seeds (seed101112, seed123, and seed42). The reported mean and standard deviation (91.74 ± 0.41%) indicate consistent performance across varying initialization conditions, supporting the model's robustness.

Run	Accuracy (%)
seed101112	92.27
seed123	91.29
seed42	91.67
Mean ± Std	**91.74 ± 0.41**

Additionally, a 5-fold cross-validation was conducted, yielding:

- **Mean Accuracy:** 91.81 ± 0.36%
- **Mean F1-Score:** 91.82 ± 0.37%

These results confirm that the proposed system demonstrates consistent performance across different training seeds and under cross-validation, with low variance in both accuracy and F1-score. Such statistical stability suggests a reliable learning process and supports the overall robustness of the final ResNet-18 configuration. However, while these metrics provide evidence of technical soundness, a deeper evaluation is needed to understand whether this consistency extends across diverse demographic groups. The following subsection explores the fairness of the model by analyzing its behavior across gender, skin tone, and age categories.

06.2.7 Fairness Evaluation

To assess potential demographic bias in the model's performance, we evaluated the best-performing ResNet-18 model on a manually labeled subset of the test dataset. This subset comprises 100 images per emotion class (a total of 700 images), each annotated with gender, skin type (Fitzpatrick [68] adjusted scale types F1–F4), and age group (Adult, Senior, or Young).

06.2.7.1 Accuracy by Gender

Table 10 *Accuracy by Gender on Labeled Subset.*

Accuracy scores of the best-performing ResNet-18 model evaluated on a subset of 700 manually annotated test images. Gender-specific performance is reported, showing similar accuracy for female (91.67%) and male (92.07%) subjects, indicating minimal disparity across gender groups.

Gender	**Accuracy**
Female (F)	91.67%

Male (M)	92.07%

The model performed similarly across genders, with a slight advantage in accuracy for male subjects. The difference (0.4%) is minimal and does not suggest a significant disparity in performance by gender.

06.2.7.2 Accuracy by Skin Type

Table 11 *Accuracy by Skin Type on Labeled Subset.*

Performance of the best-performing ResNet-18 model across different skin types based on adjusted Fitzpatrick scale (F1–F4), using a subset of 700 labeled test images. Results indicate slightly lower accuracy for subjects with light skin (F1: 87.56%), while medium and fair skin tones (F2–F3) achieved the highest accuracy rates.

Skin Type	Accuracy
F1 (Light)	87.56%
F2 (Fair)	93.36%
F3 (Medium)	94.41%
F4 (Dark)	93.26%

Performance across skin tones shows a wider range. The lowest accuracy (87.56%) was observed for skin type F1 (lightest skin tone), while the highest was for F3 (medium tone) at 94.41%. Although no skin group was severely misrepresented, the 6.85% difference between F1 and F3 indicates a potential bias that may warrant further investigation in larger, more balanced datasets.

06.2.7.3 Accuracy by Age Group

Age group analysis revealed a performance drop for senior subjects (82.35%), compared to adults (91.30%) and younger individuals (93.12%). This discrepancy suggests that the model may struggle to generalize across older faces, possibly due to limited representation in the training data.

06.2.7.4 Interpretation and Next Steps

These results highlight areas where performance disparities exist, particularly across skin tones and age groups. While the gender gap appears minimal, reduced accuracy in light-skinned individuals (F1) and seniors (S) may signal representation bias in the training distribution or facial morphology challenges that the model fails to capture.

As future work, it is recommended:

- Expanding the demographically annotated test set for stronger statistical significance.
- Exploring domain adaptation or data augmentation techniques tailored to underperforming groups.

These steps aim to enhance both the equity and robustness of FER systems in real-world deployments.

06.2.8 Comparison with State-of-the-Art Models

To contextualize the performance of the system developed in this work, a comparative analysis was conducted against state-of-the-art (SOTA) models evaluated on prominent FER datasets, including RAF-DB [64], AffectNet [11], and OULU-CASIA [8].

It is important to emphasize that this comparison is not intended to establish direct performance equivalence. Several differences in experimental conditions prevent one-to-one benchmarking. Most notably, the resolution of the images in the curated dataset used for this project. As previously mentioned, this choice was motivated by the intended application scenario, which involves webcam and mobile camera inputs where higher image resolution is typically available. Moreover, most top-performing models in the literature rely on computationally intensive architectures such as Vision Transformers (ViTs) or graph-based models, while the present study focused on lightweight convolutional architectures like ResNet-18 to favor feasibility in real-time deployment environments.

Table 12 *State-of-the-Art Accuracy Comparison Across Datasets.*

Comparison of emotion recognition accuracy between selected state-of-the-art models and the best-performing ResNet-18 model from this work. The ResNet-18 model used in this work (≈11.2 M parameters, 2.4 GFLOPs) achieved 92.00% accuracy on the test set. By comparison, Vision Transformers (e.g., ViT-B/16) typically have ~86 M parameters and ~17 GFLOPs , making them significantly larger and more computationally demanding [73].

Dataset	Model	Accuracy (%)	Source
RAF-DB	FER-former	**91.30**	[16]
AffectNet	Emotion-GCN	**66.46**	[16]
OULU-CASIA	Compact CNN	**91.67**	[16]
This Work	ResNet-18	**92.00** (test set)	—

The ResNet-18 (11. Million parameters and 2.4 G FLOPs) model trained in this project achieved an accuracy of 92.00% on a held-out test set of never-before-seen images, composed of a balanced combination of samples from AffectNet, and OULU-CASIA. The average inference time was measured at 0.0055 seconds per image, further demonstrating the model's suitability for real-time scenarios.

Although some of the SOTA models achieve high accuracy on standard datasets, they typically benefit from significantly larger training datasets, advanced augmentation pipelines, and highly expressive architectures. In contrast, the results obtained in this work illustrate the viability of using streamlined CNNs like ResNet-18 to deliver competitive performance under practical deployment constraints.

Future work should explore the incorporation of domain adaptation techniques to further improve generalization, as well as the evaluation of newer architectures such as Swin Transformers [74] and MobileViT [75], which may offer improved performance without sacrificing efficiency.

06.2.9 Reproducibility

Although the complete codebase and trained model weights are not publicly released at this stage, the following details are provided to facilitate the replication of key results.

06.2.9.1 Training Pseudocode

```
# Prepare dataset
FOR each emotion_class IN ["Anger", "Disgust", ..., "Surprise"]:
    FOR each image_file IN directory(emotion_class):
        IF file extension is in [".jpg", ".jpeg", ".png"]:
            ADD (image_path, label) TO data_list

dataset ← build DataFrame(data_list)

# Split data into training and validation sets
train_set, valid_set ← train_test_split(dataset, test_size=0.2, seed=42)
```

Figure 31 *Dataset Preparation and Splitting Pseudocode.*

Pseudocode representing the process of building the dataset by iterating through labeled image folders and splitting it into training and validation subsets using an 80/20 ratio.

```
# Define transformations
A.Compose([
    A.Resize(cfg["image_size"], cfg["image_size"]),
    A.Rotate(p=0.5, limit=[-20, 20]),
    A.HorizontalFlip(p=0.5),
    A.Affine(
        scale=(0.95, 1.05),
        translate_percent={"x": (-0.05, 0.05), "y": (-0.05, 0.05)},
        rotate=(-10, 10),
        p=0.3
    ),
    A.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1,
p=0.3),
    A.CoarseDropout(p=0.2),
    A.RandomBrightnessContrast(p=0.3),
    A.GaussianBlur(blur_limit=(3, 7), p=0.1),
    ToTensorV2()
])


# Initialize datasets and loaders
train_dataset ← CustomDataset(train_set, transform=train_transform)
valid_dataset ← CustomDataset(valid_set, transform=valid_transform)


train_loader ← DataLoader(train_dataset, batch_size, shuffle=True)
valid_loader ← DataLoader(valid_dataset, batch_size, shuffle=False)
```

Figure 32 *Data Augmentation and DataLoader Initialization.*

Pseudocode showing the definition of data augmentation techniques applied during training, the minimal preprocessing for validation, and the initialization of custom datasets and corresponding data loaders for both phases.

```
# Load pretrained model and set up training components
model ← load_pretrained("resnet18", num_classes=7, dropout=0.28)
loss_fn ← CrossEntropyLoss()
optimizer ← Adam(model.parameters, lr=5e-5, weight_decay=5e-4)
scheduler ← CosineAnnealingLR(optimizer, total_steps, min_lr=1e-6)

# Define training for one epoch
FUNCTION train_one_epoch(dataloader, model, optimizer, scheduler):
    model.train()
    INIT empty lists for predictions, labels, and losses
    FOR each batch IN dataloader:
        X, y ← batch.to(device)
        optimizer.zero_grad()
        logits ← model(X)
        loss ← loss_fn(logits, y)
        Backpropagate loss
        optimizer.step()
        scheduler.step()
        Collect predictions, labels, and loss
    RETURN accuracy, average_loss
```

Figure 33 *Model Initialization and Single Epoch Training Loop.*

Pseudocode illustrating the initialization of the pretrained ResNet-18 model along with key training components (loss function, optimizer, and learning rate scheduler), followed by the structure of the training loop for a single epoch.

```
# Define validation for one epoch
FUNCTION validate_one_epoch(dataloader, model):
    model.eval()
    INIT empty lists for predictions, labels, and losses
    FOR each batch IN dataloader:
        X, y ← batch.to(device)
        logits ← model(X)
        loss ← loss_fn(logits, y)
        Collect predictions, labels, and loss
    RETURN accuracy, average_loss
```

Figure 34 *Validation Loop for One Epoch.*

Pseudocode describing the validation process for one epoch, including model evaluation mode, batch-wise inference, loss computation, and aggregation of predictions and metrics for final accuracy and loss reporting.

```
# Training loop with early stopping
FUNCTION fit(model, train_loader, valid_loader, epochs, patience=3):
    INIT best_val_acc ← -∞
    INIT patience_counter ← 0
    FOR epoch IN range(epochs):
        SET seed for reproducibility
        train_acc, train_loss ← train_one_epoch(...)
        val_acc, val_loss ← validate_one_epoch(...)
        IF val_acc > best_val_acc:
            best_val_acc ← val_acc
            patience_counter ← 0
            Save model as best_model.pth
        ELSE:
            patience_counter += 1
            IF patience_counter ≥ patience:
                BREAK
        Save model checkpoint for current epoch
    RETURN training history and final model
```

Figure 35 *Training Loop with Early Stopping.*

Pseudocode illustrating the full training loop with early stopping. The procedure includes training and validation at each epoch, monitoring validation accuracy for improvement, saving the best-performing model, and halting the process if no improvement is observed for a defined number of epochs.

06.2.9.2 Best Configuration: Hyperparameter Settings

The best-performing configuration—achieved using ResNet-18—was obtained under the following settings:

Table 13 *Best-Performing Configuration Summary.*

The configuration is based on a ResNet-18 backbone with data augmentation enabled and early stopping, optimized using the Adam optimizer with a learning rate of 5e-5 and a dropout rate of 0.28.

Hyperparameter	Value
Backbone Architecture	ResNet-18
Batch Size	32
Learning Rate	5e-5
Dropout Rate	0.28
Weight Decay	5e-4
Data Augmentation	Enabled (full pipeline)
Image Resolution	256 × 256
Early Stopping Patience	3 epochs
Optimizer	Adam
Loss Function	CrossEntropyLoss

06.2.9.3 Hardware and Training Time

All experiments were conducted on a consumer-grade laptop with the following hardware specifications:

Table 14 *Hardware Specifications Used for Training.*

Details of the hardware environment where all experiments were conducted. The system consisted of a consumer-grade laptop equipped with an NVIDIA GTX 1650 Ti GPU, Intel Core i7-10750H CPU, and 16 GB of RAM, running Windows 11.

Component	Specification
GPU	NVIDIA GTX 1650 Ti (4GB VRAM)
CPU	Intel Core i7-10750H
RAM	16 GB DDR4

Component	Specification
Storage	512 GB SSD
Operating System	Windows 11

The average training time per model was approximately 70–80 minutes for 20–30 epochs.

06.2.9.4 Library Versions and Dependencies

The following versions of key libraries and frameworks were used:

Table 15 *Software Libraries and Versions Used.*

List of the main libraries and frameworks employed during the model development and experimentation.

Library	Version
Python	3.10
PyTorch	2.7.0
CUDA	11.8
cuDNN	90100
timm	1.0.15
albumentations	2.0.8
scikit-learn	1.6.1
MLflow	2.22.1
OpenCV	4.11.0

Additional artifacts such as training logs, confusion matrices, and MLflow dashboards have been retained for archival purposes and can be made available upon request.

In summary, the experimental evaluation demonstrated that the fine-tuned ResNet-18 model achieved a strong balance between accuracy, efficiency, and generalization under constrained resources. Through methodical tuning of hyperparameters, data augmentation, and cross-validation, the model exhibited statistically consistent results and competitive performance compared to more complex SOTA systems. However, demographic analysis revealed areas requiring further attention, particularly in age and skin tone representation. These findings underscore both the strengths and current limitations of the proposed approach.

To move beyond static evaluations and explore its viability in real-world settings, the next section presents a real-time implementation of the FER system. This module integrates the trained model into a functional application capable of detecting and visualizing facial emotions using standard devices like webcams or smartphones, offering insight into the system's performance under dynamic, unconstrained conditions.

07. REAL-TIME EMOTION DETECTION AND VISUALIZATION

Brief: *In addition to offline evaluation, this project includes a real-time FER module designed for emotion detection using standard consumer-grade devices. The objective of this component is to bridge the gap between model development and real-world application by demonstrating how the trained model performs in dynamic, unconstrained environments such as webcam or smartphone inputs.*

07.1 Real-Time Detection System

The system captures frames using either a laptop camera or a cellphone camera. Each frame is processed to detect faces, extract features, and classify emotions using the fine-tuned ResNet-18 model. The process includes:

- Face detection via OpenCV's Haar cascades
- Image preprocessing consistent with training pipelines (resizing, normalization)
- Inference through the pretrained model with softmax output
- Visualization of prediction and confidence directly overlaid on the video stream

07.2 Window-Based Temporal Smoothing

To reduce noise from frame-level prediction variability, emotion labels are aggregated in 1-second windows. The most frequent emotion within each window is logged alongside a confidence score derived from its frequency.

07.3 Output and Visualization

After each session, the system generates three types of outputs:

- Emotion Detection Video – A video file showing the real-time classification per frame, with emotion labels displayed directly over the individual's face.
- Temporal Emotion Plot – A time series graph showing emotion evolution throughout the session, rendered per frame using detected timestamps.
- Emotion Distribution Pie Chart – A summary visualization showing the proportion of time spent in each emotional state.

These outputs are saved locally and can be used for clinical, research, or personal monitoring purposes.

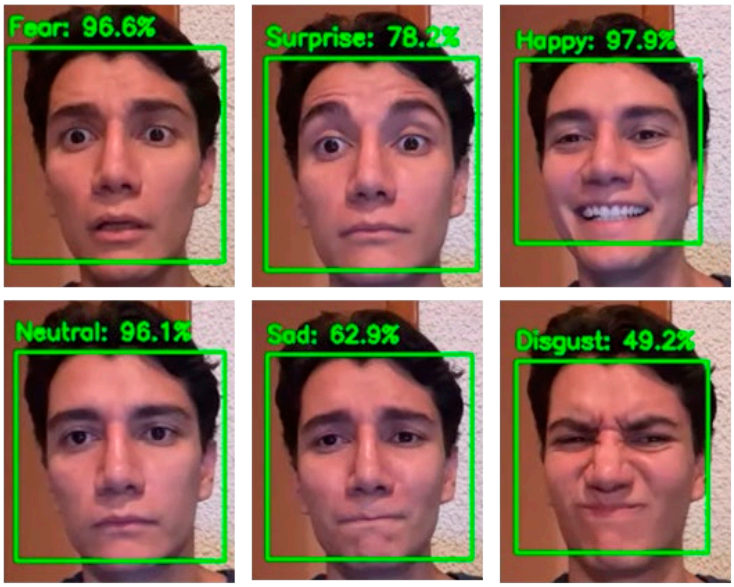

Figure 36 *Real-Time Emotion Detection Sample Output.*

Example frame showing the model's predictions during a real-time inference session using webcam input. Each detected emotion is labeled with its corresponding confidence score.

07.4 Sample Interface and Results

The following figures illustrate sample outputs generated by the system:

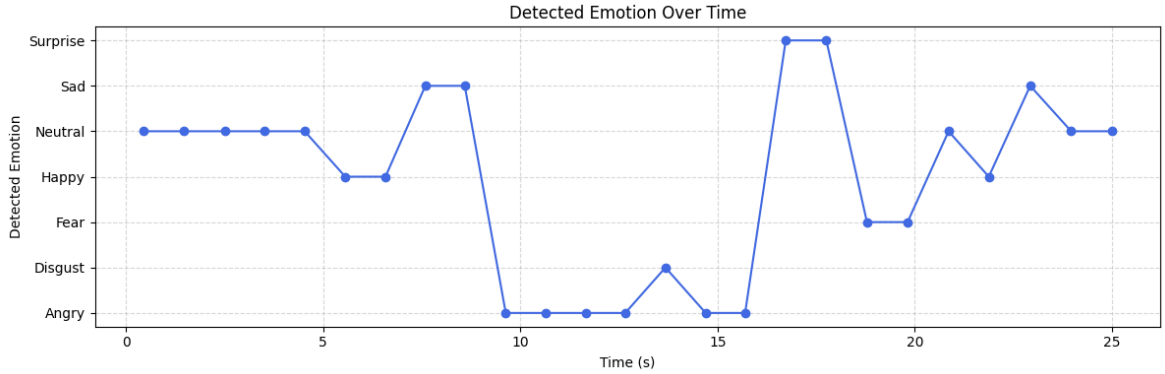

Figure 37 *Time Series of Detected Emotions Across the Session.*

Line plot showing the sequence of detected emotions over a 25-second session. The x-axis represents elapsed time (in seconds), while the y-axis corresponds to the emotion predicted at each timestamp.

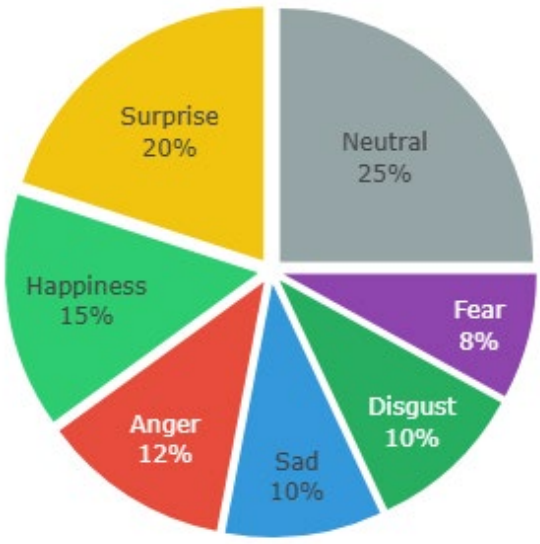

Figure 38: *Final Emotion Distribution During Session.*

Pie chart illustrating the proportional distribution of detected emotions at the end of a real-time session. Each slice represents the percentage occurrence of an emotion class throughout the entire session.

07.5 Exploratory Real-Time Evaluation Across Diverse Subjects

To obtain preliminary insights into the real-world applicability of the proposed real-time FER system, a functional test was conducted involving 14 participants (9 women, 5 men) aged between 23 and 65, with varying skin tones and facial features. Each participant was instructed to simulate all seven basic facial expressions (Happy, Sad, Angry, Disgusted, Fearful, Surprised, and Neutral) in two distinct intensities: a subtle expression (as if the emotion had just begun) and an exaggerated or peak-level expression.

The goal of this test was not to conduct a rigorous statistical validation, but rather to observe how well the model generalized when exposed to unseen users performing realistic emotional expressions under everyday conditions. Participants recorded themselves using standard webcams or mobile cameras, resulting in varied lighting, background, and camera quality.

Observations were annotated manually and classified into three categories based on recognition quality:

- Good (correct and confident classification in both subtle and intense forms),
- Moderate (detected only under clearer expression or with fluctuating confidence),
- Poor/None (emotion not detected or consistently misclassified).

The following table summarizes performance across all samples:

Table 16 *Manual Evaluation of Real-Time Emotion Recognition Across Expression Quality Levels.*

Summary of model performance on 14 manually annotated real-time samples per emotion. Each instance was categorized into Good, Moderate, or Poor/None based on detection quality, considering both subtle and intense expressions. Accuracy represents the percentage of correct predictions within each class.

Emotion	Total Samples	Good	Moderate	Poor/None	Accuracy (%)
Happiness	14	13	1	0	92.9%
Sadness	14	10	3	1	71.4%
Neutral	14	12	2	0	85.7%
Surprise	14	10	3	1	71.4%
Fear	14	6	4	4	42.9%
Disgust	14	4	3	7	28.6%
Anger	14	1	1	12	7.1%

As shown, the system exhibited high reliability for happiness and neutral with over 85% accuracy. Sadness and surprise showed promising performance, with over 70% of expressions correctly recognized. Notably, disgust and anger yielded the weakest results, with anger recognition being particularly poor across all participants—even in clearly acted expressions. Several subjects produced expressions that matched the Action Units (AUs) defined by Ekman's FACS guidelines [66], but the system failed to detect them accurately, especially for anger and disgust.

Some recurrent failure patterns emerged during the review:

- Anger and disgust were frequently misclassified as sadness or neutral.
- Participants with naturally downturned lips or distinctive eyebrow curvature were more likely to be misclassified as sad.
- In subjects with facial modifications (e.g., botox, reconstructive surgery), face detection and emotional inference were significantly hindered.
- Even when expressions were accurately performed, subtle displays were more prone to being misclassified or missed entirely.

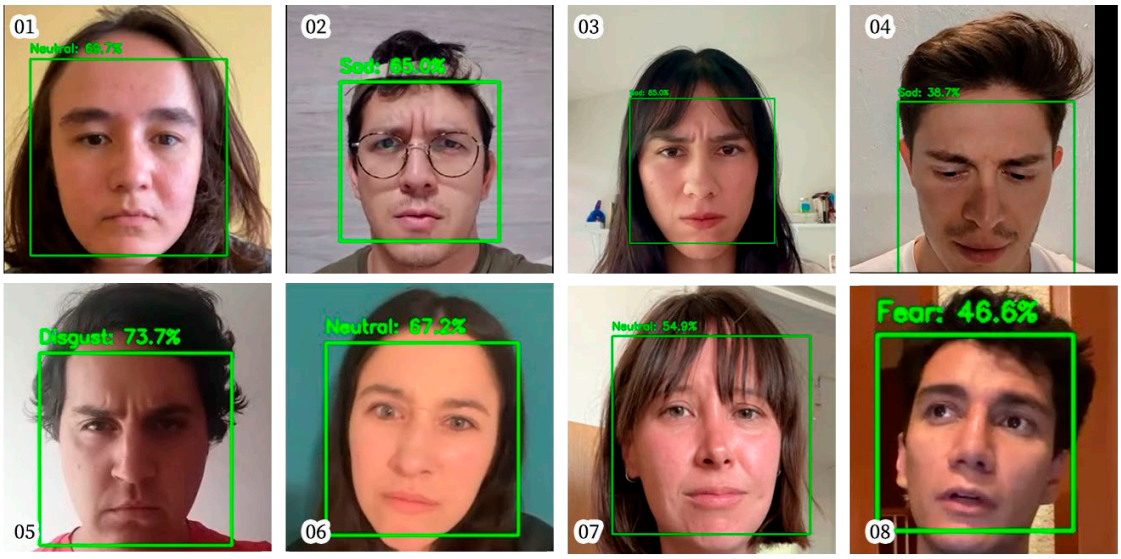

Figure 39 *Examples of Misclassifications Observed During Real-Time Testing Across Diverse Subjects.*

Sample 01 illustrates how the subject's natural downward mouth curvature led the system to consistently predict sadness, highlighting the potential benefit of user-specific calibration in future clinical implementations. Samples 02 and 03 also show sadness misclassifications, likely triggered by pronounced inner brow raises — despite the intended expression being anger. Sample 04 depicts a subtle micro expression preceding an anger display, which could be considered an understandable misclassification. In Sample 05, sadness was mistakenly labeled as disgust. In Samples 06 and 07, the subjects' attempts to express anger were interpreted as neutral. Finally, Sample 08 shows a fleeting pre-speech expression, likely neutral, that was classified as fear.

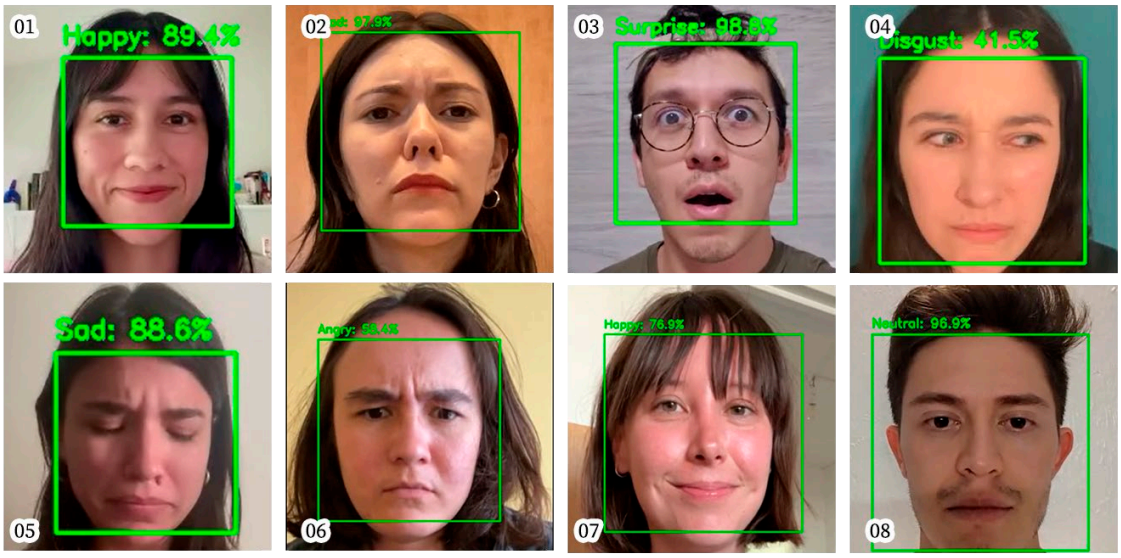

Figure 40 Correctly Classified Expressions During Real-Time Testing.

Samples 01, 02, 03, 04, and 08 demonstrate the system's strong performance on emotions such as surprise, happiness, sadness, and neutral, which were among the best-performing categories according to Table 16. Notably, Samples 04 and 06 reveal that, despite correct classification, the model exhibited relatively low confidence scores, suggesting the need for more robust training data to reinforce model certainty under such conditions.

While this evaluation does not meet the standards of a formal user study, it demonstrates that the real-time FER module shows promising capabilities for certain expressions in real-world use. However, it also highlights the need for improvements in distinguishing emotions with shared or ambiguous facial cues and the importance of considering demographic and facial diversity in training and evaluation stages.

These findings reinforce the value of continued refinement and testing before the system can be applied in therapeutic or high-stakes settings. Nonetheless, the exploratory test confirms that the system performs reliably for clear expressions of happiness, neutral state, and surprise—making it potentially useful for educational, research, or self-monitoring applications.

07.6 Discussion and Use Cases

Rather than presenting this module as a finished product, it is more accurate to regard it as a functional prototype—an initial proof-of-concept that validates the core feasibility of the system and lays the groundwork for more rigorous future developments. With further validation, especially under clinical or emotionally complex conditions, this tool has the potential to evolve into a reliable component within broader human-centered AI systems.

Having established the viability of real-time deployment, the following section turns to a broader reflection on the project's key contributions and the future directions that emerge from its limitations and accomplishments.

To further explore the real-world applicability of the system, particularly within the context of psychological practice, a brief exploratory consultation was conducted with four licensed mental health professionals. This initiative was not intended to generate conclusive or statistically representative data, but rather to obtain preliminary qualitative insights regarding the conceptual relevance, perceived utility, and possible risks of using FER technologies in therapeutic settings.

08. PRELIMINARY PSYCHOLOGICAL FEEDBACK ON THE SYSTEM AND ITS APPLICATIONS

Brief: *A brief exploratory consultation was conducted with four mental health professionals experienced in therapeutic practice. The objective was not to obtain statistically significant or generalizable data, but rather to gather qualitative insights regarding the potential relevance and feasibility of integrating FER tools within psychological contexts. The responses aimed to identify conceptual alignment, perceived utility, and ethical concerns from a practitioner's perspective.*

08.1 Table with Questions and Answers

Table 17 *Expert Feedback from Psychologists on the Use of FER Systems in Therapy.*

Summary of responses from clinical psychologists regarding the potential use, opportunities, risks, and improvement recommendations for facial emotion recognition tools in therapeutic contexts. The insights emphasize the need for contextual interpretation and ethical safeguards, while highlighting the system's value as a complementary input.

Question	Response
Do you consider a FER tool potentially useful in therapy?	Yes, it could help identify moments of emotional disconnection or suppressed affect. It may assist in reflecting emotional states not verbally expressed. Useful as a complementary, but not standalone, input.
What opportunities do you see in the use of this type of tool?	Detect emotional flattening, anxiety, or frustration in real time. Support emotional self-regulation work. Enable long-term emotion tracking to assess therapeutic progress.
What risks or limitations do you identify?	Misclassification of emotions due to overreliance can lead to a wrong diagnosis. Lack of narrative/context for interpretation. Overreliance on facial data, ignoring other emotional cues. Ethical concerns around privacy and data sensitivity.
What would you recommend to improve the system?	Define thresholds between adaptive and maladaptive emotions. Consider incorporating physiological indicators (e.g., cortisol). Allow customization based on patient history. Clarify how longitudinal tracking supports therapeutic processes.

08.2 Perceived Relevance and Limitations

All four professionals considered that a system capable of detecting facial expressions could, at the very least, offer auxiliary value in therapeutic and research contexts. Two of them expressed confidence in the emotional relevance of such systems, while the others emphasized the importance of combining this tool with other sources of data and psychological judgment, mentioning a visual analysis is an advancement but not a standalone tool.

08.3 Potential Contributions

Respondents identified specific areas of added value:

- Detecting emotional incongruence between verbal expression and facial affect.
- Supporting long-term emotional monitoring for self-reflection or therapeutic follow-up.
- Enhancing patient self-awareness during sessions or at-home tracking.

One professional highlighted the relevance of this approach in contexts where emotional awareness or expression is limited due to strong traumas, although all agreed that interpretative caution is essential.

08.4 Recommendations for Future Development

Key suggestions for improvement included:

- Establishing emotion recognition thresholds to distinguish functional from maladaptive emotional patterns.
- Exploring multimodal integration, such as coupling FER with physiological or behavioral data.
- Enabling flexible calibration based on individual patient profiles.

08.5 Ethical and Practical Risks

Participants raised several concerns about potential ethical and clinical misuses, including:

- The risk of patient inhibition if aware of being continuously analyzed.
- Overinterpretation of automatically generated reports by therapists.
- The necessity of maintaining clinical judgment as the primary interpretative lens.

These insights reinforce the importance of using FER systems as support tools rather than diagnostic instruments, and of ensuring that any deployment in mental health contexts is preceded by rigorous validation, fairness assessment, and ethical review.

In summary, preliminary feedback from mental health professionals underscores the promise and limitations of integrating FER tools into therapeutic contexts. While the system is not viewed as a replacement for human clinical judgment, it is regarded as a potentially valuable complement when used with interpretive care and ethical safeguards. The reflections shared by practitioners suggest a pathway toward responsible, patient-centered innovation.

Building on this foundation, the following section explores an equally critical dimension of real-world deployment: the technical and economic feasibility of deploying FER systems in resource-constrained settings. This perspective is especially relevant for developing regions, where low-cost, lightweight technologies could help bridge gaps in emotional health monitoring and access to care.

09. AN ARGUMENT FOR FER IN LOW-RESOURCE SETTINGS

Brief: *While research on FER often emphasizes state-of-the-art models [16] deployed on high-performance computing infrastructure, there are arguments that lightweight FER systems can be realistically deployed in developing regions.*

09.1 Technical Feasibility: Lightweight Hardware Requirements

The proposed FER system is built upon compact models such as ResNet-18 and EfficientNet-B0, both of which are recognized for their favorable trade-off between accuracy and computational cost [6][12]. These architectures can be deployed on edge devices such as the NVIDIA Jetson Nano [76] or Raspberry Pi [77]. According to [77], lightweight convolutional networks have been successfully optimized for inference on devices with limited processing power, offering real-time emotion classification while avoiding cloud latency.

09.2 Precedent in Mobile and Edge AI Health Deployments

Projects like UMPHCS and AICOM illustrate the viability of deploying AI in mobile and resource-constrained contexts. For example, [1] reports the successful use of deep learning on mobile devices for dermatological screenings, while [6] highlights the integration of AI into portable systems applied in the healthcare together with the domains where it is being applied, ranging from haematology to digital pathology and rapid infectious disease diagnostics [2]. These examples underscore the potential of applying FER for mental health support through similar infrastructure.

09.3 Deployment Platforms and Performance

Table 18 *Inference Performance and Cost Comparison Across Edge Devices.*

*Summary of benchmark results for different edge computing platforms tested with various deep learning models. Metrics include frames per second (FPS), mean inference time per frame (ms), and estimated retail price. The table highlights trade-offs between speed, model complexity, and hardware cost for real-time FER deployment scenarios. Data adapted from [76] While some of these devices have been discontinued, similar or more accessible alternatives are expected to emerge as edge AI continues to evolve. **

Platform	Model Name	FPS	Mean Time (ms)	Approx. Price (USD)	Notes
Coral USB	MobileNetV2	365.82	2.73	$75 [78]	Outstanding speed, compact form factor, ideal for edge applications

Jetson Nano (FP16)	MobileNetV2	82.75	12.08	-	Balanced performance, widely used in low-budget AI systems
Coral USB	EfficientNetV2B0	75.91	13.17	$75 [78]	Lightweight and energy-efficient, suitable for constrained hardware
Jetson Nano (FP16)	EfficientNetV2B0	50.92	19.64	-	Real-time capable with optimization
Jetson Nano (FP16)	ResNet50	35.13	28.47	-	Borderline real-time, heavier model but still viable
Neural Stick	MobileNetV2	40.31	24.81	$69 [79]	Plug-and-play USB accelerator, ideal for lightweight models
Neural Stick	ResNet50	26.03	38.42	$69 [79]	Below real-time threshold, not ideal for high-speed scenarios

While low-cost edge devices such as the Google Coral USB, Jetson Nano, and Neural Stick offer promising real-time performance, their deployment is not trivial. These platforms require technical knowledge to configure, convert, and optimize models, often involving frameworks like TensorFlow Lite, TensorRT, or OpenVINO. Additionally, the advertised prices do not reflect the total cost of implementation, which may include peripherals, setup time, and maintenance.

For non-technical users or clinical settings, these requirements may be a significant barrier. As an alternative, using a standard laptop for inference remains a feasible option. Inference tasks are far less demanding than training, meaning mid-range consumer laptops could support lightweight real-time emotion recognition without the need for high-end hardware. Future work could explore streamlined deployment strategies tailored to clinicians or therapists with limited technical expertise.

09.4 Local Applicability: The Case of Mexico

Despite economic limitations, there are targeted initiatives that aim to close this gap. *Mexico's National Digital Health Strategy* emphasizes the importance of expanding digital services in remote communities, including investment in telemedicine and low-cost health technologies and holds a place in the country's annual operations budget [1]. Moreover, rural clinics in Mexico typically receive an annual operational budget through public health programs and local government support, allowing for modest but strategic technology acquisitions [80].

Programs such as UNESCO's digital-inclusion initiatives and local innovation support the adoption of computing tools in low-resource settings, further increasing the viability of FER deployment in community clinics and mental health outreach programs [3].

09.5 Viability

Deploying the proposed FER system in low- and middle-income regions is not only technically feasible but can become economically and logistically viable. The use of efficient models like ResNet-18 and EfficientNet-B0 allows for operation on modest hardware, while previous success stories in edge-AI health applications reinforce the practicality of this approach. Further studies are recommended to assess pilot programs, cost-effectiveness, and integration with local healthcare frameworks.

10. CONCLUSIONS

The present work successfully fulfilled its general objective by developing and evaluating a facial emotion recognition system based on a pretrained ResNet-18 architecture, contributing to the creation of scalable and accessible tools for mental health monitoring. The model was fine-tuned using a carefully curated hybrid dataset that combined public benchmark datasets with hand-selected images, thereby achieving the first specific objective.

To improve the model's capacity to generalize across diverse conditions, an efficient image preprocessing pipeline was implemented, addressing the second objective. Standard performance metrics—such as accuracy, confusion matrices, and classification reports—were used to rigorously assess model performance, thereby meeting the third objective. The final model achieved an average accuracy of 91.74% ± 0.40% (n=3), with a maximum test accuracy of 92.27%, highlighting both its robustness and consistency.

A modular experimentation framework was designed to support systematic tuning of hyperparameters and comparison between architectures, fulfilling the fourth objective. Furthermore, real-time performance was validated through tests using both laptop and smartphone cameras in varied expression scenarios, thus accomplishing the fifth objective. Finally, an analysis of the societal and economic implications of FER technology was conducted, highlighting its potential impact in the context of digital mental health support, thereby addressing the sixth and final objective.

Through these contributions and validated results, this work demonstrates the viability and relevance of FER systems in the broader landscape of affective computing and psychological well-being.

11. FUTURE WORK

While this work achieved a solid performance using compact CNN architectures trained on a curated dataset, several limitations and open questions were identified throughout the process. These present fertile ground for future improvements and research. This section outlines potential directions for addressing current constraints and expanding the system's applicability, robustness, and technical contributions.

11.1 Architectural Enhancements

The current system leverages a standard ResNet-18 model without structural modifications. Although this architecture proved robust and efficient, particularly when trained with higher-resolution images, it does not introduce architectural innovations specific to FER. Future work could explore:

- Incorporating attention mechanisms to enhance the model's ability to focus on emotionally relevant facial regions.
- Designing or adapting FER-specific architectures, potentially integrating spatial-temporal modeling for video-based emotion detection.
- Experimenting with multi-branch networks or ensemble methods that combine compact backbones with more advanced feature extractors [75].

11.2 Custom Loss Functions and Augmentation Strategies

The current training pipeline uses standard loss functions and augmentation techniques. Improvements could include:

- Exploring loss functions tailored to emotion classification, such as focal loss [81] to handle class imbalance.
- Proposing novel data augmentation techniques that simulate more realistic occlusions, lighting changes, or micro-expressions to improve model robustness.
- Evaluating the effectiveness of emotion-preserving augmentations, ensuring that synthetic variations do not distort the expression semantics.

11.3 Handling and Leveraging Synthetic Data

A significant portion of the dataset used in this work includes high-quality synthetic facial expressions. While effective, this raises concerns about domain shift and generalizability. Future work may focus on:

- Applying domain adaptation techniques to bridge the gap between synthetic and real-world data distributions.
- Using domain randomization or generative models to increase the diversity of synthetic samples while preserving label fidelity.

11.4 Bias Mitigation and Fairness Evaluation

Although this study incorporated an initial fairness analysis across gender, age group, and skin type, future research should extend this with:

- A more systematic demographic audit, using larger annotated samples and formal fairness metrics.
- Implementing bias mitigation strategies, such as reweighting or data balancing.
- Evaluating model behavior on intersectional subgroups (e.g., older females with darker skin tones), which often receive less accurate predictions.

11.5 Deployment-Oriented Validation

While the system was designed with real-time applications in mind, particularly webcam or smartphone usage, extensive real-world validation remains pending. Future steps should include:

- Conducting controlled user studies to quantify system accuracy under varied lighting, pose, and expression subtleties.
- Measuring latency and resource usage across multiple deployment environments (e.g., Android, browser-based, embedded systems).

11.6 Benchmark Generalization

The current system was not tested on standard FER benchmarks under their native conditions (e.g., FER2013 [65]at 48×48 resolution). Future work should:

- Evaluate the model on external datasets to assess generalization across distributions.
- Investigate multi-resolution training schemes to improve performance across diverse input sources.
- Compare performance with larger transformer-based models using matched input resolutions, to isolate architectural advantages.

11.7 Advanced Exploration of Compact Architectures

Although ResNet-18 was selected as the primary model for this work, alternative lightweight architectures such as EfficientNet-B0 and MobileNetV3 demonstrated strong potential in preliminary experiments. Further tuning of these models—considering aspects such as dropout, weight decay, and transfer learning strategies—may lead to even higher accuracy with reduced computational costs, making them ideal candidates for real-time applications on constrained hardware.

11.8 Expanded Collaboration with Mental Health Professionals

Future iterations of this project would benefit from deeper collaboration with psychologists and psychiatrists. Structured interviews and formal usability studies could guide improvements in the system's integration with therapeutic practices, ensuring the tool aligns with established clinical standards and offers meaningful insights into emotional well-being.

11.9 Theoretical Analysis of ResNet-18 for FER

Given the model's strong empirical results despite its simplicity, it would be valuable to understand:

- Why ResNet-18 exhibits stable learning behavior and generalization in the FER context.
- Whether low-level convolutional features suffice for most emotion-related cues or whether newer models are overfitting to dataset-specific patterns (e.g., by performing a layer activation analysis to better understand which visual features are actually being learned).

11.10 Extension Beyond Basic Emotions

While this work focused on Ekman's six universal emotions and a neutral category, there is an opportunity to include more subtle and context-specific emotional states such as contempt, confusion, or interest. These additional classes could expand the system's relevance in more nuanced psychological analyses.

By addressing these directions, future work can further the system's impact in both technological and mental health domains, moving closer to scalable, human-centered applications of affective computing.

12. REFERENCES

[1] R. Yotsu, Z. Ding, J. Hamm, and R. Blanton, “Deep learning for AI-based diagnosis of skin-related neglected tropical diseases: a pilot study,” Mar. 15, 2023. doi: 10.1101/2023.03.14.23287243.

[2] I. Hernández-Neuta *et al.*, “Smartphone-based clinical diagnostics: towards democratization of evidence-based health care,” Jan. 01, 2019, *Blackwell Publishing Ltd*. doi: 10.1111/joim.12820.

[3] Organización Mundial de la Salud, “Estrategia mundial sobre salud digital 2020–2025,” 2021.

[4] F. Ramzan *et al.*, “A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer’s Disease Stages Using Resting-State fMRI and Residual Neural Networks,” *J Med Syst*, vol. 44, no. 2, p. 37, Feb. 2020, doi: 10.1007/s10916-019-1475-2.

[5] H. Hadjar, B. Vu, and M. Hemmje, “TheraSense: Deep Learning for Facial Emotion Analysis in Mental Health Teleconsultation,” *Electronics (Switzerland)*, vol. 14, no. 3, Feb. 2025, doi: 10.3390/electronics14030422.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[7] M. Pantić, M. Valstar, R. Rademaker, and L. Maat, “Web-based database for facial expression analysis,” in *IEEE Int. Conf. Multimedia and Expo (ICME)*, Amsterdam, Jul. 2005.

[8] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietikäinen, “Facial expression recognition from near-infrared videos,” *Image Vis Comput*, vol. 29, no. 9, pp. 607–619, Aug. 2011, doi: 10.1016/j.imavis.2011.07.002.

[9] C. Ronquillo, “Ecuadorian Facial Expressions (EFE),” Kaggle.

[10] K. Roman, “Facial Emotion Recognition Dataset,” Kaggle.

[11] A. Mollahosseini, B. Hasani, and M. H. Mahoor, “AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild,” *IEEE Trans Affect Comput*, vol. 10, no. 1, pp. 18–31, Jan. 2019, doi: 10.1109/TAFFC.2017.2740923.

[12] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946

[13] A. Howard *et al.*, “Searching for MobileNetV3,” May 2019, [Online]. Available: http://arxiv.org/abs/1905.02244

[14] P. Ekman, “An argument for basic emotions,” *Cogn Emot*, vol. 6, no. 3–4, pp. 169–200, May 1992, doi: 10.1080/02699939208411068.

[15] R. Y. da Silva Franco, R. S. do Amor Divino Lima, R. do Monte Paixão, C. G. R. dos Santos, and B. S. Meiguins, “UXmood-A sentiment analysis and information visualization tool to support the evaluation of usability and user experience,” *Information (Switzerland)*, vol. 10, no. 12, Dec. 2019, doi: 10.3390/info10120366.

[16] T. Kopalidis, V. Solachidis, N. Vretos, and P. Daras, “Advances in Facial Expression Recognition: A Survey of Methods, Benchmarks, Models, and Datasets,” *Information (Switzerland)*, vol. 15, no. 3, Mar. 2024, doi: 10.3390/info15030135.

[17] V. R. Boppana, “Machine Learning and AI Learning: Understanding the Revolution,” 2022.

[18] R. Guo, H. Guo, L. Wang, M. Chen, D. Yang, and B. Li, “Development and application of emotion recognition technology — a systematic literature review,” Dec. 01, 2024, *BioMed Central Ltd*. doi: 10.1186/s40359-024-01581-4.

[19] R. Singh and C. Prabha, “A Review on Facial Expression Recognition Models, Datasets, and Future Direction of Research,” in *5th International Conference on Sustainable Communication Networks and Application, ICSCNA 2024 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 661–666. doi: 10.1109/ICSCNA63714.2024.10864269.

[20] Allied Market Research, “Emotion Detection and Recognition Market Size, Share, Competitive Landscape and Trend Analysis Report, by Software Tool, by Application, by Technology, by End User: Global Opportunity Analysis and Industry Forecast, 2021–2031,” Feb. 2023.

[21] The Insight Partners, “Pharmaceutical Market Size is projected to reach US$ 2.84 billion by 2031 at 7.1% CAGR - The Insight Partners,” 2025.

[22] S. Zoting, “Artificial Intelligence (AI) Market Size, Share, and Trends 2025 to 2034.”

[23] World Health Organization, “World mental health report,” 2022.

[24] T. Lee *et al.*, “A Deep Learning Driven Simulation Analysis of the Emotional Profiles of Depression Based on Facial Expression Dynamics,” *Clinical Psychopharmacology and Neuroscience*, vol. 22, no. 1, pp. 87–94, Feb. 2024, doi: 10.9758/cpn.23.1059.

[25] D. Sharma, J. Singh, S. S. Sehra, and S. K. Sehra, “Demystifying Mental Health by Decoding Facial Action Unit Sequences,” *Big Data and Cognitive Computing*, vol. 8, no. 7, p. 78, Jul. 2024, doi: 10.3390/bdcc8070078.

[26] Y. Zhou, F. Ren, S. Nishide, and X. Kang, “Facial sentiment classification based on resnet-18 model,” in *Proceedings - 2019 International Conference on Electronic Engineering and Informatics, EEI 2019*, Institute of Electrical and Electronics Engineers Inc., Nov. 2019, pp. 463–466. doi: 10.1109/EEI48997.2019.00106.

[27] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” Apr. 2022.

[28] A. Vaswani *et al.*, “Attention Is All You Need.”

[29] Y. Li, M. Wang, M. Gong, Y. Lu, and L. Liu, “FER-former: Multi-modal Transformer for Facial Expression Recognition.”

[30] G. Tecuci, “Artificial intelligence,” *WIREs Computational Statistics*, vol. 4, no. 2, pp. 168–180, Mar. 2012, doi: 10.1002/wics.200.

[31] V. H. Martínez, “Deep Learning Course, Sessions 1–12 (Spring 2025),” Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO), 2025.

[32] R. Nadarajan and N. Sulaiman, “Comparative Analysis in Execution of Machine Learning in Breast Cancer Identification: A Review,” *J Phys Conf Ser*, vol. 1874, no. 1, p. 012032, May 2021, doi: 10.1088/1742-6596/1874/1/012032.

[33] G. Kaur, “Power of Artificial Neural Networks: A Comprehensive Guide,” 2023.

[34] Shaundra B. Daily *et al.*, "Affective Computing: Historical Foundations, Current Applications, and Future Trends," Emotions and Affect in Human Factors and Human-Computer Interaction. Accessed: Nov. 21, 2023. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/affective-computing

[35] Ö. Ezerceli and M. T. Eskil, "Convolutional Neural Network (CNN) Algorithm Based Facial Emotion Recognition (FER) System for FER-2013 Dataset," in *International Conference on Electrical, Computer, Communications and Mechatronics Engineering, ICECCME 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICECCME55909.2022.9988371.

[36] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Aug. 01, 2018, *Springer Verlag*. doi: 10.1007/s13244-018-0639-9.

[37] G. Kumar, P. Kumar, and D. Kumar, "Brain Tumor Detection Using Convolutional Neural Network," in *2021 IEEE International Conference on Mobile Networks and Wireless Communications, ICMNWC 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICMNWC52512.2021.9688460.

[38] R. Liu, "Empirical Evaluation of Residual CNN in Emotion Recognition."

[39] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," 2020.

[40] L. Torrey and J. Shavlik, "Transfer Learning."

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[42] Z. C. L. M. L. and A. J. S. Aston Zhang, "14.2. Fine-Tuning," Dive into Deep Learning.

[43] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. Feris, "SpotTune: Transfer Learning through Adaptive Fine-tuning."

[44] Torch Contributors., "resnet18," 2017.

[45] B. Bischl *et al.*, "Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges," Nov. 2021.

[46] E. Elgeldawi, A. Sayed, A. R. Galal, and A. M. Zaki, "Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis," *Informatics*, vol. 8, no. 4, Dec. 2021, doi: 10.3390/informatics8040079.

[47] R. Wang, I. Nabney, and M. Golbabaee, "Efficient Hyperparameter Importance Assessment for CNNs," Oct. 2024.

[48] A. Hernández-García and P. König, "Further advantages of data augmentation on convolutional neural networks," Jun. 2019, doi: 10.1007/978-3-030-01418-6_10.

[49] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," 2014.

[50] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," Dec. 2012, [Online]. Available: http://arxiv.org/abs/1212.5701

[51] S. Tayabali, "A simple guide to building a confusion matrix," Oracle AI & Data Science Blog.

[52] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2018, doi: 10.1016/j.aci.2018.08.003.

[53] T. Viering and M. Loog, “The Shape of Learning Curves: a Review,” Nov. 2022, [Online]. Available: http://arxiv.org/abs/2103.10948

[54] M. Ibrahim, “A Deep Dive Into Learning Curves in Machine Learning,” W&B Fully Connected.

[55] scikit-learn developers, “classification_report.”

[56] NumPy, “NumPy: the absolute basics for beginners.”

[57] W3Schools, “NumPy Introduction.”

[58] Pandas, “User Guide.”

[59] scikit-learn developers, “User Guide,” 2019.

[60] PyTorch Contributors, “PyTorch documentation.”

[61] fastai. Inc, “Pytorch Image Models (timm),” 2022.

[62] M. Zaharia *et al.*, “Accelerating the Machine Learning Lifecycle with MLflow,” 2018.

[63] MLflow Project, “MLflow Documentation.”

[64] S. Li, W. Deng, and J. Du, “Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jul. 2017, pp. 2584–2593. doi: 10.1109/CVPR.2017.277.

[65] I. J. Goodfellow *et al.*, “Challenges in Representation Learning: A report on three machine learning contests,” Jul. 2013.

[66] P. Ekman and W. V. Friesen, “Facial Action Coding System,” Jan. 14, 2019. doi: 10.1037/t27734-000.

[67] B. Farnsworth, “Facial Action Coding System (FACS) – A Visual Guidebook,” *ResearchGate*, Jun. 2025.

[68] T. B. Fitzpatrick, “The Validity and Practicality of Sun-Reactive Skin Types I Through VI,” *Arch Dermatol*, vol. 124, no. 6, p. 869, Jun. 1988, doi: 10.1001/archderm.1988.01670060015008.

[69] E. Engel, L. Li, C. Hudy, and R. Schleusner, “Multi-modal Transfer Learning for Dynamic Facial Emotion Recognition in the Wild,” Apr. 2025, [Online]. Available: http://arxiv.org/abs/2504.21248

[70] H. Sheng and M. Lau, “Optimising Facial Expression Recognition: Comparing ResNet Architectures for Enhanced Performance,” in *International Conference of Control, Dynamic Systems, and Robotics*, Avestia Publishing, 2024. doi: 10.11159/cdsr24.123.

[71] Tutuianu Gianmarco, Liu Yan, Alamäki Ari, and Kauttonen Janne, “Benchmarking Deep Facial Expression Recognition: An Extensive Protocol with Balanced Dataset in the Wild,” 2023. [Online]. Available: https://github.com/hardikvasa/google-

[72] B. M. Hussein and S. M. Shareef, “An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning,” *ITM Web of Conferences*, vol. 64, p. 01003, 2024, doi: 10.1051/itmconf/20246401003.

[73] The PyTorch Foundation, “vit_b_16.”

[74] Z. Liu *et al.*, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” Aug. 2021.

[75] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Jun. 2021.

[76] R. Tobiasz, G. Wilczyński, P. Graszka, N. Czechowski, and S. Łuczak, “Edge Devices Inference Performance Comparison,” Jun. 2023, doi: 10.5626/JCSE.2023.17.2.51.

[77] Z. Jiang, T. Chen, and M. Li, “Efficient Deep Learning Inference on Edge Devices,” California: SysML 2018, Apr. 2018.

[78] Google LLC., “USB Accelerator,” 2025.

[79] Inc. Amazon.com, “Intel NCSM2450.DK1 Movidius Neural Compute Stick Click to see full view Intel NCSM2450.DK1 Movidius Neural Compute Stick,” 2025.

[80] CÁMARA DE DIPUTADOS DEL H. CONGRESO DE LA UNIÓN, “PRESUPUESTO DE EGRESOS DE LA FEDERACIÓN PARA EL EJERCICIO FISCAL 2025,” 2024.

[81] S. Tripathi, A. Kumar, A. Ramesh, C. Singh, and P. Yenigalla, “Focal Loss based Residual Convolutional Neural Network for Speech Emotion Recognition,” Jun. 2019.