T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

A FEDERATED LEARNING PLATFORM FOR FACIAL EXPRESSION RECOGNITION

SALIH ARDA KIZIL

SUPERVISOR
DR. YAKUP GENÇ

GEBZE
2024

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

A FEDERATED LEARNING PLATFORM FOR FACIAL EXPRESSION RECOGNITION

SALIH ARDA KIZIL

SUPERVISOR
DR. YAKUP GENÇ

2024
GEBZE

GEBZE TECHNICAL UNIVERSITY	GRADUATION PROJECT JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 25/01/2024 by the following jury.

JURY

Member
(Supervisor) : DR. Yakup Genç

Member : Prof. Dr. Yusuf Sinan Akgül

ABSTRACT

In the landscape of edge computing, this project pioneers a Federated Learning (FL) platform tailored for facial expression recognition. The platform transcends device-specific boundaries, utilizing Raspberry Pi 3B+ boards as illustrative instances of edge devices. It aims to establish a privacy-preserving FL environment, allowing local model training on diverse edge devices while collectively contributing to a global model. Leveraging the Flower framework, the project introduces a privacy-centric workflow, seamlessly integrating with the FL paradigm. Real-time facial expression detection, facilitated by a Convolutional Neural Network (CNN) based on the VGG19 architecture, is complemented by OpenCV for efficient face detection. The federated learning setup empowers users across diverse edge devices to actively participate in model training, contributing local facial expression data without compromising privacy. This work not only advances privacy-preserving machine learning on edge devices but also contributes significantly to the practical implementation and exploration of Federated Learning, demonstrating its versatility beyond specific hardware constraints.

Keywords: Federated Learning (FL), Edge Computing, Facial Expression Recognition, Privacy-preserving, Convolutional Neural Network (CNN), Flower Framework.

ÖZET

Bu proje, uç bilişim ortamında yüz ifadesi tanıma için özel olarak tasarlanmış bir Federe Öğrenme (Federe Öğrenme) platformunu öncülüyor. Platform, spesifik bir cihaza özel sınırları aşmak amacıyla Raspberry Pi 3B+ kartlarını örneklemeler olarak kullanıyor. Platform, çeşitli uç cihazlarında yerel model eğitimine izin vererek aynı zamanda küresel bir modele katkıda bulunarak ölçeklenebilir ve gizliliği koruyan bir Federe Öğrenme ortamı kurmayı amaçlıyor. Proje, Federe Öğrenme paradigmasıyla sorunsuz entegre olan gizliliğe odaklı bir iş akışını Flower yazılım iskeleti ile birleştirerek bu hedefe ulaşmayı amaçlıyor. VGG19 mimarisine dayalı bir Convolutional Neural Network (CNN) tarafından gerçekleştirilen gerçek zamanlı yüz ifadesi tespiti, OpenCV ile etkili yüz tespiti ile destekleniyor. Federe öğrenme kurulumu, çeşitli uç cihazlarında kullanıcıları model eğitimine aktif olarak katılım sağlamaya teşvik ederek, yerel yüz ifadesi verilerini gizliliği ihlal etmeden katkıda bulunmalarına olanak tanıyor. Bu çalışma sadece uç cihazlarda gizliliği koruyan makine öğrenimini ilerletmekle kalmıyor, aynı zamanda Federe Öğrenme'nin pratik uygulanması ve keşfinde önemli katkılarda bulunarak belirli donanım kısıtlamalarının ötesindeki çok yönlülüğünü gösteriyor.

Anahtar Kelimeler: Federe Öğrenme, Uç Bilişim, Yüz İfadesi Tanıma, Gizlilik Koruma, Convolutional Neural Network (CNN), Flower Yazılım İskeleti.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Dr. Yakup Genç for their unwavering support, guidance, and mentorship throughout the development of this project. Their insightful feedback and encouragement were instrumental in shaping the project's direction.

I extend my heartfelt thanks to Selin Duru Derindağ for her invaluable motivational support and unwavering presence throughout my educational journey.

I would also like to extend my thanks to the Flower development team. Their responsiveness and willingness to answer my questions played a crucial role in overcoming challenges and enhancing the overall quality of the project.

Salih Arda Kızıl

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation		Explanation
FER	:	Facial Expression Recognition
FL	:	Federated Learning
CNN	:	Convolutional Neural Network
VGG19	:	Visual Geometry Group 19 (a specific CNN architecture)
OpenCV	:	Open Source Computer Vision
gRPC	:	gRPC Remote Procedure Call

CONTENTS

Abstract iv

Özet v

Acknowledgement vi

List of Symbols and Abbreviations vii

Contents ix

List of Figures x

1 Introduction 1

2 Background 3

2.1 Overview of Facial Expression Recognition (FER) 3

2.2 Existing Paradigms and Limitations 3

2.3 Relevance of Federated Learning in FER 3

3 Methodology and Design 5

3.1 Centralized Approach 5

3.1.1 Model Development 5

3.1.2 Dataset 6

3.2 FL Simulation and FL Platform 7

4 Implementation and Application Development 9

4.1 FL Simulation Implementation 9

4.2 FL Platform Implementation 10

4.2.1 Platform Application 10

4.2.2 FL Strategy 11

4.2.3 Driver Implementation 12

5 Results and Findings 14

5.1 Centralized Results 14

5.2 Simulation Results 16

5.3 FL Platform Results 17

6 Conclusion **18**

Bibliography **19**

LIST OF FIGURES

1.1 Simple illustration of the platform application. 1
1.2 Top-level architecture of the FL setups. [1] 2

2.1 Federated learning formulation for privacy preserving personalization approach for FER [2] 4

3.1 VGG19 Architecture [3] 6
3.2 Sample of the FER2013 dataset [5] 6
3.3 gRPC Overview [7] 7

4.1 Client-Server Architecture of FL [8] 9
4.2 FL Platform in action 10

5.1 Loss-Accuracy Plots for Centralized Model 14
5.2 Confusion Matrix for Centralized Model 15
5.3 FL Simulation Plots 16

1. INTRODUCTION

In the intricate domain of human-computer interaction, the significance of Facial Expression Recognition (FER) reverberates across a spectrum of applications, from fostering emotion-aware computing to enhancing human-robot collaboration. However, the conventional paradigms of centralized machine learning in FER raise poignant concerns regarding user privacy. Against the backdrop of this challenge and the burgeoning ubiquity of edge devices, this project embarks on a mission to introduce an innovative solution— a Federated Learning (FL) platform meticulously crafted for facial expression recognition.

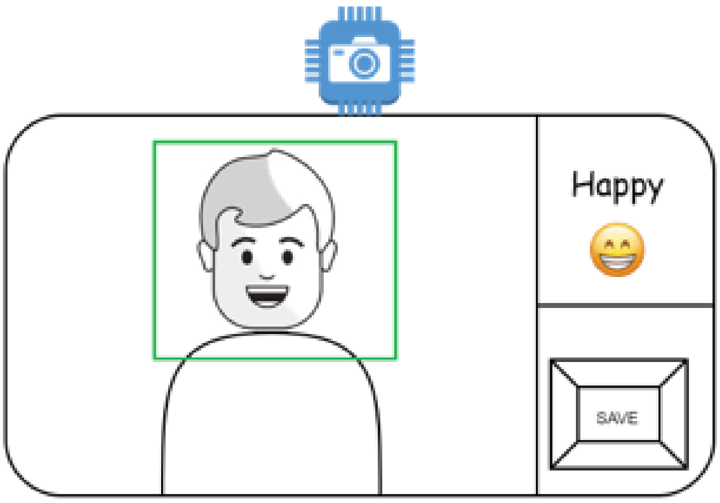


Figure 1.1: Simple illustration of the platform application.

This FL platform, illustraed in the Figure 1.1, showcased through the utilization of Raspberry Pi 3B+ boards as emblematic edge devices (Figure 1.2), marks a departure from traditional methodologies. The core ethos of the project lies in its commitment to privacy-preserving machine learning. The envisioned platform empowers edge devices to collaboratively train a facial expression recognition model without compromising individual user data, thus presenting a novel and ethically sound approach.

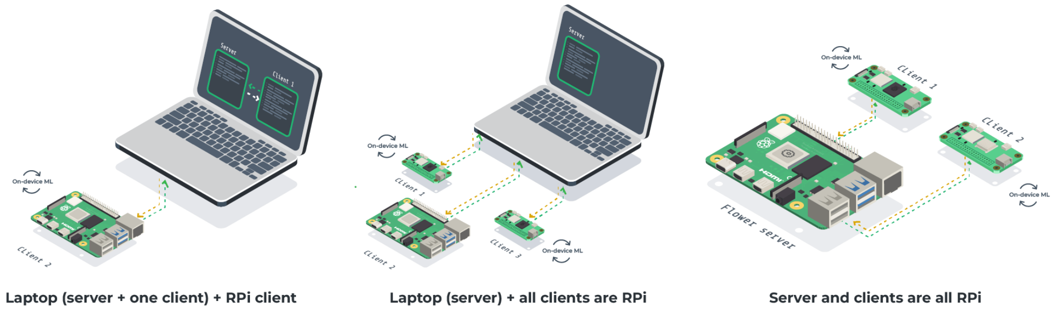


Figure 1.2: Top-level architecture of the FL setups. [1]

2. BACKGROUND

2.1. Overview of Facial Expression Recognition (FER)

Facial Expression Recognition (FER) has emerged as a pivotal domain within human-computer interaction, exerting profound implications across various applications. As humans increasingly interact with computing systems, the ability to comprehend and respond to facial expressions becomes integral for creating emotionally aware and responsive technologies. FER holds significance in applications ranging from human-robot collaboration to virtual reality, where understanding user emotions enhances the overall user experience.

2.2. Existing Paradigms and Limitations

Conventional approaches to Facial Expression Recognition (FER) have predominantly relied on centralized machine learning models, creating a paradigm where facial data is collected and stored in a centralized repository. While these centralized models have demonstrated competency in recognizing facial expressions, they come with inherent challenges that have spurred the exploration of alternative methodologies. One of the primary concerns associated with centralized models is the potential compromise of user privacy. The centralized storage of sensitive facial data poses significant threats to user confidentiality, raising ethical and privacy-related issues that have become increasingly pertinent in today's digital landscape.

2.3. Relevance of Federated Learning in FER

The advent of Federated Learning (FL) marks a transformative paradigm shift in addressing the limitations inherent in centralized Facial Expression Recognition (FER) models. FL, as a decentralized learning approach, introduces a novel methodology where edge devices collaborate to collectively train a global model without the need to share raw data. This decentralized approach not only aligns with the imperative to prioritize user privacy but also redefines the landscape of FER, presenting a more adaptable and inclusive model that can thrive in diverse contexts and demographics (see Figure 2.1).

In the realm of FER, the importance of user privacy cannot be overstated. Federated Learning addresses this concern by allowing individual edge devices to retain their facial data locally, avoiding the need for centralized storage. This not only safeguards user privacy but also establishes a foundation of trust, crucial for the widespread acceptance and adoption of facial expression recognition technologies.

Beyond its privacy-preserving capabilities, the relevance of Federated Learning extends to its embrace of the inherent diversity in facial expressions across different contexts and demographics. Traditional centralized models may struggle to capture the nuances of facial expressions that vary based on cultural, individual, or situational factors. FL, with its collaborative learning approach, acknowledges and adapts to this diversity, resulting in a more adaptable and inclusive FER model.

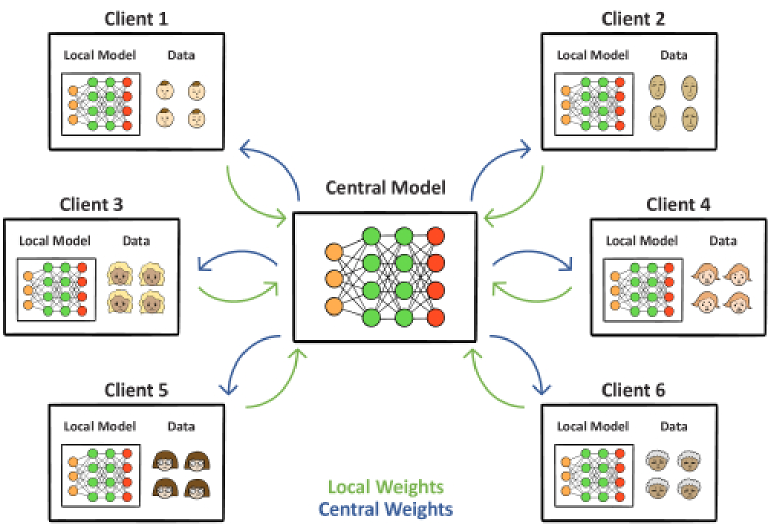


Figure 2.1: Federated learning formulation for privacy preserving personalization approach for FER [2]

3. METHODOLOGY AND DESIGN

3.1. Centralized Approach

As an initial phase of the project, a centralized approach is employed to develop the facial expression recognition (FER) model. The FER model is implemented using the Convolutional Neural Network (CNN) based on the VGG19 architecture and trained on the FER2013 dataset. The decision to adopt a centralized training approach is driven by two key objectives:

1. To verify the functionality and suitability of the FER model for the task of Facial Expression Recognition.
2. To potentially use this pre-trained model as initial parameters in the subsequent Federated Learning (FL) setup.

This strategic choice allows for the thorough validation of the FER model's performance in a centralized context before transitioning to the federated learning environment. By doing so, the project aims to ensure the reliability and effectiveness of the model in recognizing facial expressions and explores the potential benefits of utilizing pre-trained parameters in the FL framework.

3.1.1. Model Development

The development of the Facial Expression Recognition (FER) model involves the meticulous implementation of the VGG19 architecture, a widely recognized deep learning model renowned for its exceptional performance in image classification tasks. The VGG19 architecture is characterized by a deep structure comprising 19 layers, encompassing 16 convolutional layers and 3 fully connected layers.

Implementing the VGG19 [3] architecture involves configuring the network's parameters, including filter sizes, depths, and fully connected layer dimensions. The convolutional layers are designed to capture intricate patterns and features from facial images, while the fully connected layers facilitate the translation of these features into emotion predictions.

The choice of the VGG19 model is deliberate, aiming for a balance between model complexity and performance. This model's proven efficacy in image-related tasks, coupled with its deep architecture, positions it as a suitable candidate for the nuanced task of recognizing facial expressions.

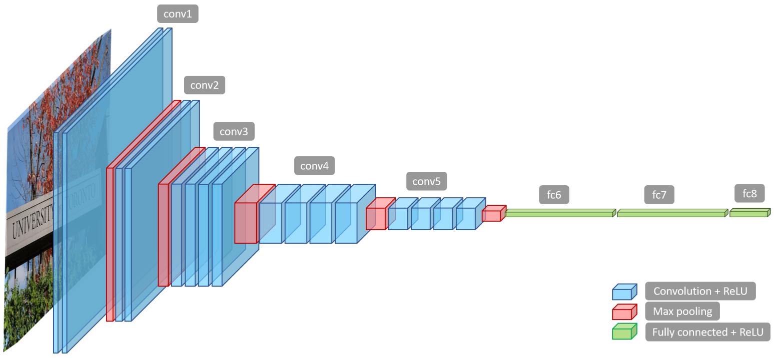


Figure 3.1: VGG19 Architecture [3]

3.1.2. Dataset

The FER2013 [4] (Facial Expression Recognition 2013) dataset is a crucial component for training and evaluating our facial expression recognition model. Comprising 28,709 examples in the training set, along with 3,589 examples each in the public testing and private test sets, it provides a diverse array of grayscale images, each measuring 48×48 pixels. The dataset covers seven distinct emotion categories: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral.

Each image is meticulously labeled, facilitating precise model training and rigorous evaluation. Figure 3.2 provides a sample glimpse of this dataset, showcasing its richness and diversity.

Figure 3.2: Sample of the FER2013 dataset [5]

3.2. FL Simulation and FL Platform

Before implementing the FL platform, as part of the methodology, we initiated the implementation of an FL simulation. This simulation stands as a crucial and strategic step, playing a pivotal role in validating the proposed FL platform's functionality and meticulously assessing its potential within a controlled environment.

Our approach to federated learning leverages the capabilities of the Flower framework developed by Adap. Flower [6], an open-source framework, is meticulously crafted to simplify the creation and management of federated learning systems. With a rich suite of tools and abstractions, it empowers developers with a streamlined workflow, facilitating the seamless design and orchestration of federated learning processes.

The architecture of our FL simulation and platform revolves around the client-server, driver-server, client-driver interactions, facilitated by the gRPC (gRPC Remote Procedure Call) communication protocol. Developed by Google, gRPC is a high-performance and open-source RPC framework. Employing a robust binary serialization format and HTTP/2 for transport, gRPC ensures efficiency and speed in communication between clients and servers. This design choice aligns seamlessly with the decentralized nature inherent in federated learning.

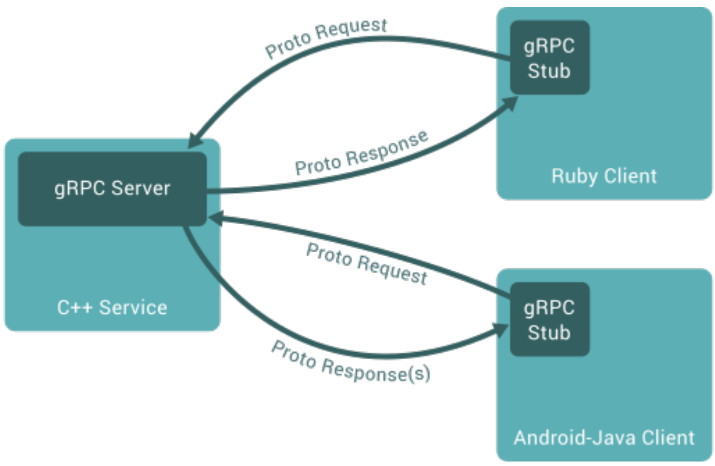


Figure 3.3: gRPC Overview [7]

Additionally, it's important to note that while the connection between the driver-server and client-server is established using gRPC, the triggering mechanism within our system involves a connection between the client and driver through a normal socket.

This socket-based connection serves as the mechanism for triggering federated learning rounds, providing a flexible and effective means for coordination.

Within our simulation and platform setups, Flower assumes a central role as the core framework for both clients and the server. The server operates continuously, ready to receive orchestration signals from a dedicated driver script. This driver script plays a pivotal role in orchestrating and coordinating the federated learning process, managing intricate interactions between the server and actively participating clients. The orchestration, in essence, forms the backbone of our FL simulation, ensuring a cohesive and collaborative environment for distributed model training across diverse edge devices.

4. IMPLEMENTATION AND APPLICATION DEVELOPMENT

4.1. FL Simulation Implementation

Before embarking on the implementation of the FL platform, a comprehensive federated learning simulation setup was meticulously crafted. In this simulation, clients undergo both individual and collaborative learning experiences using the FER2013 dataset, facilitated by the robust Flower framework. The simulation unfolds through iterative rounds of model training and evaluation, capturing the essence of decentralized learning dynamics.

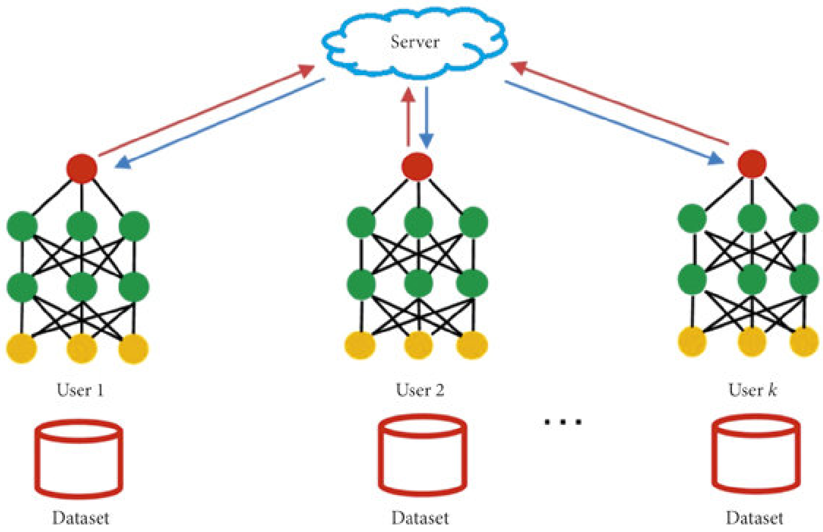


Figure 4.1: Client-Server Architecture of FL [8]

The server-side architecture comprises the Flower server and a driver, orchestrating the intricate dance of the FL simulation. The server persistently runs, poised and ready to be orchestrated into action by the driver. The driver, functioning as the master conductor, assumes the critical role of managing orchestration, communication with clients, aggregation of model updates, and the overarching evaluation of the entire federated learning process.

4.2. FL Platform Implementation

4.2.1. Platform Application

Within the realm of our federated learning platform, an application has been meticulously implemented using the Qt framework. Qt, renowned for its cross-platform capabilities, empowers the creation of graphical user interfaces and applications that seamlessly transcend various software and hardware platforms. This includes but is not limited to Linux, Windows, macOS, Android, or embedded systems. The strength of Qt lies in its ability to maintain a consistent codebase while delivering native applications with inherent capabilities and speed.

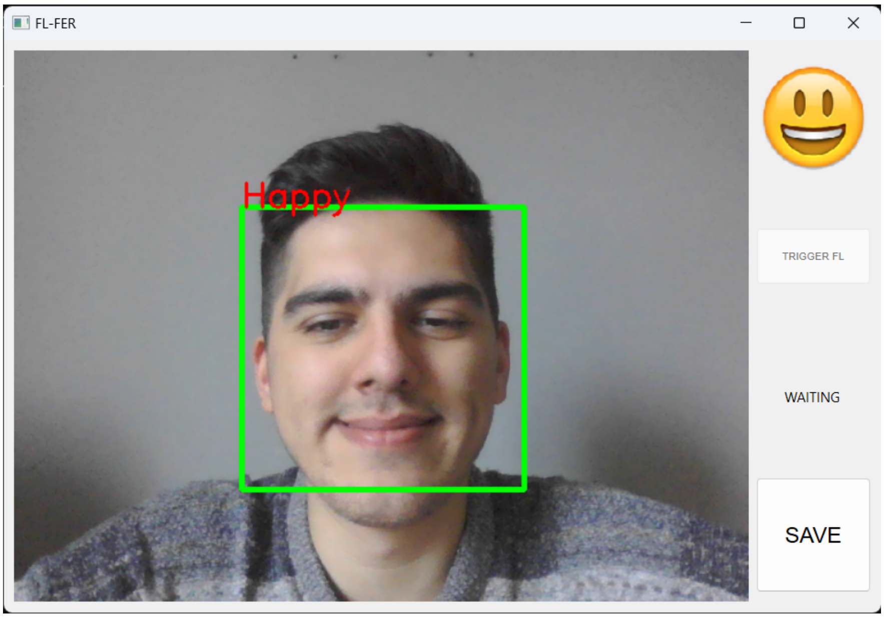


Figure 4.2: FL Platform in action

The application, albeit basic in functionality, is designed with two primary layouts: Video Processor and Actions.

In the Video Processor layout, the application taps into the camera feed, displaying real-time video. When a face is detected, the application showcases the predicted emotion associated with the detected face. The face detection leverages OpenCV, and the emotion prediction is facilitated by the FER model.

Moving to the Actions layout, positioned at the top right corner, the detected emotion is visually represented as an emoji. Below this representation, there are two buttons and a label. The first button, labeled ”TRIGGER FL,” serves as a signal trigger, prompting the driver to initiate federated learning. The second button, labeled ”SAVE,” enables the user to store the current frame as part of the local dataset. The label provides real-time feedback on the federated learning (FL) process's status, indicating one of the following states:

- **READY:** Sufficient number of `min_available_clients` are connected.
- **WAITING:** Awaiting connection from other clients to satisfy the min available clients.
- **FL RUNNING:** FL is triggered and actively running.
- **FL ERROR:** An error occurred during the FL process.
- **IDLE:** The application is currently in an idle state.

This application serves as the user interface for interacting with the federated learning platform, providing essential controls and real-time feedback on the FL process.

4.2.2. FL Strategy

In the realm of Federated Learning, the term ”strategy” encompasses the methodology used to aggregate model parameters. Various strategies, such as FedAvg, FedAdam, and FedMedian, dictate the approach to parameter aggregation. In our Federated Learning setup, a custom implementation of the FedAvg strategy, termed FedSGD, has been employed as the baseline for our federated learning process.

FedAvg, short for Federated Averaging, is a foundational strategy in federated learning. The key principle involves distributing the model training process across decentralized clients while aggregating their model updates to create a global model. The FedAvg strategy operates in rounds, where each round consists of the following steps:

1. Clients receive the current global model from the server.
2. Clients locally train their models using their respective datasets.
3. Clients send their model updates (gradients) to the server.
4. The server aggregates the model updates using a weighted average.

5. The updated global model is sent back to the clients for the next round.

This iterative process allows the global model to improve over time without exposing raw client data to the central server. The weighted average ensures that clients with more data or higher accuracy contribute proportionally to the global model. FedAvg strikes a balance between collaboration and privacy, making it a widely adopted strategy in federated learning scenarios.

The rationale behind the implementation of a custom FedAvg lies in the asynchronous nature of our workflow. The asynchronous version of FedAvg aligns seamlessly with the dynamics of our federated learning setup, enhancing the efficiency and adaptability of the overall strategy.

4.2.3. Driver Implementation

In the context of our federated learning platform, the driver plays a crucial role in orchestrating the federated learning process. The driver is responsible for managing communication with clients, triggering federated learning rounds, and overseeing the overall coordination of the decentralized training.

The driver implementation involves the use of socket programming, establishing a server that listens for incoming connections from clients. Clients connect to the server and communicate their status, such as readiness or the initiation of federated learning. The driver maintains a dynamic list of connected clients.

The orchestration process includes the following key functionalities:

1. **Waiting for Clients:** The driver continuously waits for clients to connect. Upon connection, it acknowledges the client's presence and updates the overall readiness status.

2. **Client Handling:** Each connected client is handled by a separate thread. The driver monitors client activity, such as receiving signals to trigger federated learning. It broadcasts relevant messages to all connected clients.

3. **Federated Learning Trigger:** When a client signals the initiation of federated learning ("TRIGGER_FL" message), the driver triggers the federated learning process. It starts the custom implementation of the FedAvg strategy, coordinating the aggregation of model updates from participating clients.

4. **Broadcasting Messages:** The driver can broadcast messages to all connected clients, informing them of the status of the federated learning process (e.g., "FL_STARTED," "FL_ENDED," "FL_ERROR").

5. **Graceful Disconnection:** The driver handles disconnections gracefully, updating the list of connected clients and adjusting the readiness status accordingly.

The implementation utilizes threading to manage concurrent client connections, ensuring that the driver can interact with multiple clients simultaneously. The driver's role is pivotal in maintaining the synchronization and coordination required for an effective federated learning process within the proposed platform.

5. RESULTS AND FINDINGS

5.1. Centralized Results

The centralized training process resulted in a model with an accuracy of 65%. While this accuracy is noteworthy, it is essential to consider the context of facial expression recognition, where state-of-the-art performance on the FER2013 dataset hovers around 70%. The achieved accuracy serves as a solid starting point for evaluating the effectiveness of federated learning.

Figure 5.2 presents the confusion matrix, while Figure 5.1 displays the loss-accuracy plots. These visualizations provide a comprehensive overview of the centralized model's performance and behavior during training.

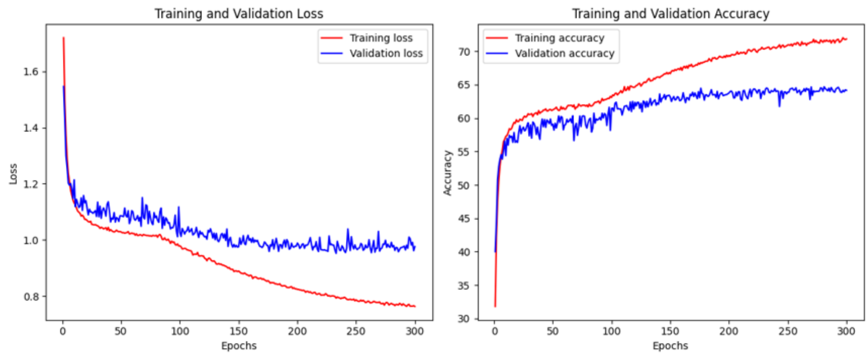


Figure 5.1: Loss-Accuracy Plots for Centralized Model

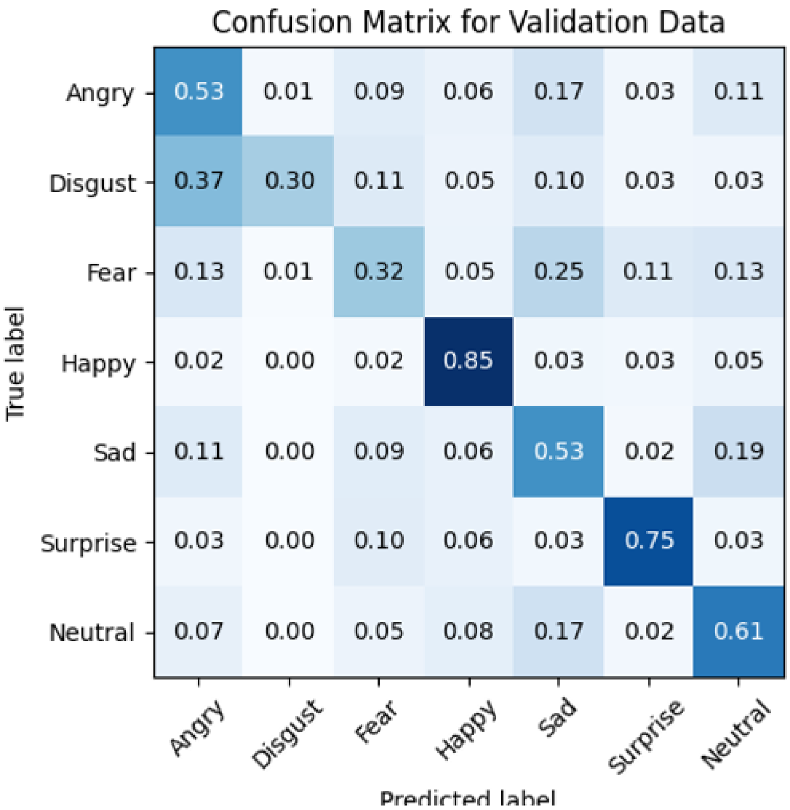
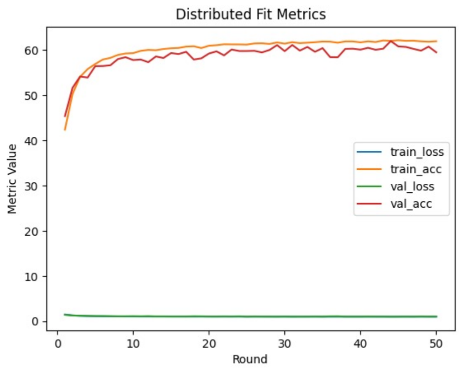


Figure 5.2: Confusion Matrix for Centralized Model

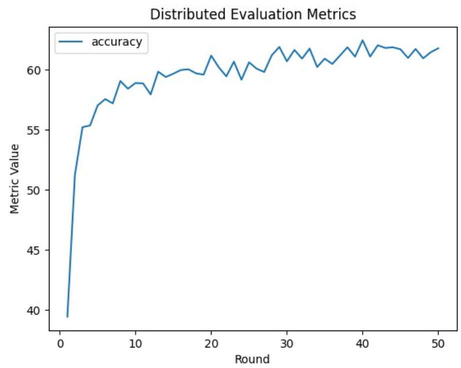
5.2. Simulation Results

The federated learning simulation was conducted over 50 rounds with 2 clients, each comprising 2 epochs on every participating client. The results provide insights into the model's collaborative learning performance and the impact of decentralized training.

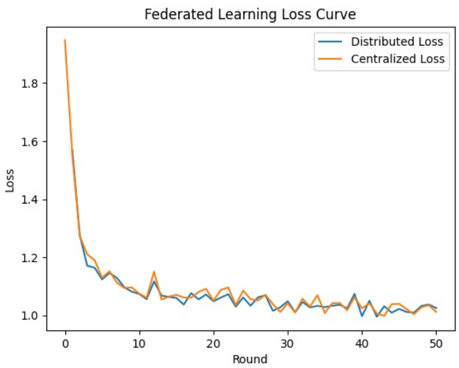
The metrics obtained from federated learning, including accuracy, loss, and other relevant statistics, were recorded and analyzed. Additionally, the loss curve over the simulation rounds was plotted to visualize the convergence and performance of the federated learning model.



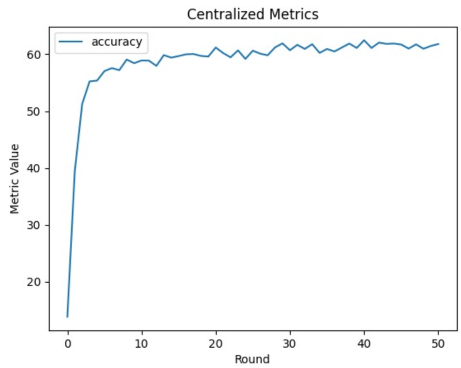
(a) FL Distributed Fit Metrics



(b) FL Distributed Accuracy



(c) FL Loss



(d) FL Centralized Accuracy

Figure 5.3: FL Simulation Plots

5.3. FL Platform Results

The FL platform successfully orchestrates federated learning rounds, triggering collaborative model updates across participating clients. The asynchronous workflow, although challenging, provides a foundation for scalable and collaborative learning scenarios.

To assess the practical implications of the FL platform, a specific scenario was devised to emulate real-world conditions:

1. **Client Initialization:** Clients start with a not-pretrained model, initializing their parameters randomly. This represents a common scenario where devices have no prior knowledge of the facial expression recognition task.

2. **Local Frame Collection:** Client 1 actively engages with the FL platform, saving local frames and tagging them using the application. This action contributes to local dataset enrichment, simulating real-world user interactions.

3. **Federated Learning Trigger:** Client 1 initiates the federated learning workflow by triggering the FL process. This action sets the collaborative learning mechanism in motion.

4. **Participation of Clients:** Clients participate in the federated learning workflow. Notably, Client 2, lacking any saved frames, skips the model training phase but actively contributes to the federated learning process. This mimics scenarios where devices may have varying degrees of local data.

5. **Post-FL Evaluation:** After the federated learning process concludes, an evaluation is conducted. Client 2, initially without any saved frames, demonstrates the ability to accurately detect emotions, showcasing the collaborative learning outcomes.

This scenario provides a tangible demonstration of the FL platform's capabilities in real-world conditions, illustrating its potential for collaborative facial expression recognition even when clients start with randomly initialized models. The platform successfully runs this scenario as a result.

6. CONCLUSION

This project aimed to explore the intersection of federated learning (FL) and facial expression recognition, with a primary focus on the development and evaluation of a FL platform. The journey unfolded through distinct phases, commencing with the training of a centralized model on the FER2013 dataset.

The centralized approach provided valuable insights into the capabilities and limitations of facial expression recognition, achieving a recognition accuracy of 65%. However, the true essence of the project lies in the subsequent implementation of the FL platform, a crucial component designed to enable collaborative learning in a decentralized fashion.

The FL platform, built upon the Flower framework, showcased its ability to orchestrate federated learning workflows seamlessly. The asynchronous nature of the workflow, although posing challenges, underlines the potential for collaborative learning across multiple devices. The custom implementation of the FL platform introduced a user-friendly application that integrates facial expression recognition, face detection, and federated learning orchestration.

As we reflect on the findings, it becomes evident that the FL platform serves as the cornerstone of this project, opening avenues for future research and real-world applications. The asynchronous FL workflow, coupled with a carefully designed Qt application, represents a scalable solution for collaborative learning in facial expression recognition.

While the centralized results provided a baseline, the true innovation and promise lie in the decentralized, collaborative nature of federated learning. The FL platform paves the way for exploring more robust strategies, enhancing scalability, and addressing the challenges associated with asynchronous FL workflows. This project sets the stage for further advancements at the intersection of FL and facial expression recognition, driving the exploration of real-world applications and the refinement of federated learning strategies.

BIBLIOGRAPHY

[1] "Flower framework diagram." Accessed: Month Day, Year. (), [Online]. Available: `https://github.com/adap/flower/blob/main/examples/embedded-devices/_static/diagram.png`.

[2] A. Salman and C. Busso, "Privacy preserving personalization for video facial expression recognition using federated learning," Nov. 2022, pp. 495–503. DOI: `10.1145/3536221.3556614`.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, Sep. 2014.

[4] I. J. Goodfellow, D. Erhan, P.-L. Carrier, *et al.*, *Fer2013: Facial expression recognition challenge*, `https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data`, 2013.

[5] G. Verma and H. Verma, "Hybrid-deep learning model for emotion recognition using facial expressions," *The Review of Socionetwork Strategies*, vol. 14, Aug. 2020. DOI: `10.1007/s12626-020-00061-6`.

[6] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," *CoRR*, vol. abs/2007.14390, 2020. arXiv: `2007.14390`. [Online]. Available: `https://arxiv.org/abs/2007.14390`.

[7] "Grpc: A high-performance, open-source universal rpc framework." (), [Online]. Available: `https://grpc.io/`.

[8] A. Alferaidi, K. Yadav, Y. Alharbi, W. Viriyasitavat, S. Kautish, and G. Dhiman, "Federated learning algorithms to optimize the client and cost selections," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–9, Apr. 2022. DOI: `10.1155/2022/8514562`.