Cloud-based Deployment Strategies of Hierarchical Federated Learning Systems for Face Recognition

Stefano Verrilli
Department of Science and Technology
University of Naples Parthenope
Naples, Italy
stefano.verrilli001@
studenti.uniparthenope.it

Massimiliano Giordano Orsini
Department of Science and Technology
University of Naples Parthenope
Naples, Italy
massimiliano.giordanoorsini001@
studenti.uniparthenope.it

Vincenzo Mele
Department of Science and Technology
University of Naples Parthenope
Naples, Italy
vincenzo.mele001@
studenti.uniparthenope.it

Martina Roscica
Department of Science and Technology
University of Naples Parthenope
Naples, Italy
martina.roscica001@
studenti.uniparthenope.it

Renato Esposito
Department of Science and Technology
University of Naples Parthenope
Naples, Italy
renato.esposito001@
studenti.uniparthenope.it

Alfredo Mungari
Department of Science and Technology
University of Naples Parthenope
Naples, Italy
alfredo.mungari001@
studenti.uniparthenope.it

***Abstract*—Critical real-time applications in the IoT context like monitoring, diagnostics, automation, and control have increased demand for processing and analysis while the classical drawbacks that are associated with cloud computing, including latency, bandwidth consumption, and most significantly, privacy issues persist. These challenges are addressed by edge computing due to the fact that it decentralizes data computation and storage closer to the source hence reducing latency, increasing response time while having less utilization of bandwidth and, at the same time, observing the privacy of data. Nonetheless, using deep learning combined with edge computing for elaborate data processing and decision-making at the data generation site raises unique prospects and issues.**

The major aim of this paper is therefore to demonstrate how hierarchical federated learning can improve the operational performance, reliability and security of face recognition systems. It suggests a systematic method that takes advantage of the fact that HFL is a hierarchical structure for computing the amount of distributed computation, for enhancing privacy and scalability. The structure of this paper includes a review of the current literature on hierarchical and federated approaches for face recognition, a discussion of a real-world face recognition scenario, and the presentation of several cloud-based solutions. It also details the experiments conducted and their outcomes, followed by recommendations for future research and a summary of the conclusions and challenges identified.

***Index Terms*—edge-cloud deployment, federated learning, face recognition, privacy**

I. Introduction

Edge computing emerges as a fundamental development in the distributed computing scenario, also reinforced by the increasing demand for real-time data processing and analysis of Internet of Things (IoT) devices [1].

Limitations of traditional cloud computing are mitigated by bringing computations and data storages closer to the source of data. Using edge computing turns out to be very beneficial for latency reduction, bandwidth use, response times, and solving many privacy issues [1].

In addition, there are more advantages to integrating deep learning with edge computing because this leads to more sophisticated data analysis [2] and decision-making [3] directly at the source of the data.

It is possible to reduce the dependency on centralized cloud resources and provide real-time insights using edge computing, which allows for the deployment of these models on edge devices or edge servers [4]. This fusion reduces latency and bandwidth usage, improving privacy and security by processing sensitive data locally [5].

Federated Learning is a technique that implements collaborative model training among multiple edge devices [6], called clients, without sharing raw data, preserving data privacy and allowing collective learning capability [7].

Scalability and improvement of data security are the cornerstones in favor of Federated Learning. Especially in machine learning application settings, security advantages are helpful in exchanging environments, in which model parameters sharing and training cycles carry-out expose the system to vulnerabilities and data leaks.

These drawbacks can be used to alter ML models results or to obtain private user information. Therefore these risks are largely reduced by Federated Learning's capacity to maintain localized data, therefore only model changes are exchanged [8].

Processing sensitive data locally mitigates several privacy risks, such as data interception during transmission to the

cloud, unauthorized accesses, and data breaches that can occur in centralized databases [9].
Furthermore, limited computational and storage capabilities of edge devices can make it challenging to reinforce the security measures, and making them more susceptible to attacks [10]. Hence, the security focus of federated learning makes it tailored for sensitive data. Therefore an open challenge concerns applications where the data is a human face such as in face recognition applications. The objective of the following paper aims to illustrate the role of hierarchical federated learning in a face recognition problem.
The remaining sections of this work are organized as follows. Section II briefly reviews the current literature regarding hierarchical and federated approaches for face recognition, supported by cloud-based solutions when available. Section III discusses a real-world scenario for a face recognition system and proposes several cloud-based solutions. Section IV and V describe experiments carried out and the respective outcomes for the proposed implementation, respectively. Section VI proposes possible ideas and recommendations for further researches and developments. Finally, Section VII summarizes the most relevant conclusions and drawbacks.

II. Related Works

Several approaches to the task of face recognition using a hierarchical federated learning approach have been introduced in the literature, trying to mitigate well-known issues of traditional federated learning (privacy, security, scalability, generalization, performance).

FedGC, proposed in [11] is a revolutionary architecture created especially to improve privacy in face recognition-federated learning. Their method's primary innovation is its correction of gradients from the standpoint of backward propagation. FedGC precisely injects a cross-client gradient term to fix gradients of class embeddings by creating a regularizer based on softmax. Like regular softmax, this approach not only preserves the validity of the loss function but also offers improved privacy.

Solomon et al. [3] addressed privacy and data transmission problems of federated learning in supervised and unsupervised face recognition scenarios, both with and without secure aggregators. Generative adversarial networks (GANs) are utilized to produce fake data at the edge in order to introduce different types of data without requiring data transmission.

FedFace [12] introduced a federated learning framework for privacy-aware face recognition model training, where a pre-trained face extractor is already available at the server. Before the first communication round, the server communicates this global model to each of the edge-clients. Moreover, it employs a mean feature initialization method for local identity proxies and uses a spreadout regularizer [13] on the server side to ensure clear separation between these proxies. However, several criticalities of this approach (single indentity for each local device, sending identity proxies to the server) have been raised [14] [15].

To address FL protocols violation, Meng et al. [16] proposed PrivacyFace, a framework largely improves the federated learning face recognition via communicating auxiliary and privacy-agnostic information among clients. It mainly consists of two components: a practical Differentially Private Local Clustering (DPLC) mechanism is proposed to distill sanitized clusters from local class centers, and a consensus-aware recognition loss subsequently encourages global consensuses among clients, which ergo results in more discriminative features.

FedFR [14] introduces a federated learning framework that jointly optimizes generic and personalized face recognition models, by proposing several techniques (hard negative sampling, contrastive regularization, Decoupled Feature Customization) tailored for the face recognition task.

Although all the aforementioned works are more or less valid options for addressing the task of federated face recognition, none of them provide a concrete edge-cloud solution for the deployment.

Koubaa et al. [17] explored the advantages and trade-offs between cloud and edge computing paradigms for deploying real-time face recognition systems. Three inference architectures for the deployment, including cloud-based, edge based, and hybrid were proposed.

This work provides valuable insights about several deployment strategies of a federated face recognition system, together with a deep discussion about its criticalities and a large performance comparison among different cloud-based and edge-based GPU platforms as hardware configurations of the system. However, it is mainly related to issues of real-time face recognition inference, rather than those concerning federated training.

Patel et al. [18] proposed an intelligent smart doorbell based on Federated Deep Learning, which can deploy and manage video analytics applications such as a smart doorbell across Edge and Cloud resources. The Federated Server has been implemented using the Flask framework, containerized using Nginx and Gunicorn, which is deployed on AWS EC2 and AWS Serverless architecture.

An end-to-end federated learning process has been proposed, even though all the discussion is mainly focused on the task of object detection, which is definitely less strict than face recognition, in terms of verification, identification, privacy and security issues.

For this reason, our discussion is fairly centered on several cloud-based deployment strategies to hierarchical federated learning for face recognition. In particular, our contributions could be summarized as follows:

- We discuss a smart-home surveillance scenario, where smart cameras are used for monitoring potentially dangerous activities by unknown individuals (e.g. potentially intruiders, not recognized as part of the extended nuclear family).
- Cloud-enabled and cloud-native solutions are proposed, taking into account critical aspects of both face recognition and federated learning, concerning scalability, privacy and security.

III. Proposed Method

Before diving into the core discussion of this work, which is related to the cloud-based deployment strategies to hierarchical federated face recogntion systems, we provide a formal and technical background to the problem of face recognition.

A. Background on Face Recogniton

Face recognition is the science which involves the understanding of how the faces are recognized by biological systems and how this can be emulated by computer systems. Modern computer systems employ different visual devices to capture and process faces as best indicated by each particular application. These sensors can be video cameras (e.g., a camcorder), infrared cameras, or among others, 3D scans [19].

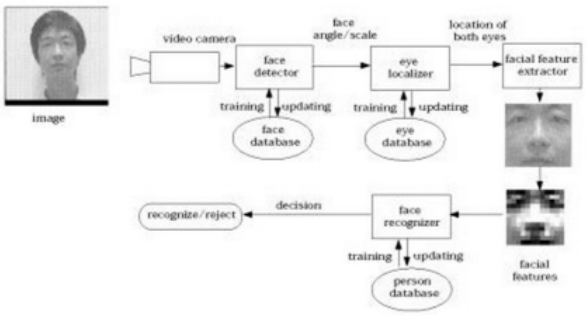

Fig. 1. A framework for face recognition system. [20]

A face recognition inference system involves two main operations:

- Face Detection: the first step is to locate faces within an image or video. Before deep learning, cascaded AdaBoost classifier and Viola-Jones [21] were widely used for face detection. These algorithms use some kind of features, such as Haar-like features, SURF [22] and Multi-Block LBP [23]. With the spread of Deep Learning were introduced some deep face detector such as Faster R-CNN [24], YOLO [25], FaceBoxes [26], SSD [27] and MTCNN [28], these deep face algorithms obtain better results than traditional cascated classifier [29].
- Face identification: The second step is to analyze the detected faces. The goal of this phase is to obtain key facial features and their geometric relationships. After that, the system yields facial signatures, which are transformed into facial embeddings, following recent advancements' trend. Facial embeddings capture the main characteristics of faces as represented by composite numerical information [30]. The last step is the recognition stage, where facial embedding has a fundamental role. During this phase, embeddings retrieved previously are compared by complex algorithms with known embeddings stored in a database [20]. Some recently frameworks have been developped about face recognition, such as DeepFace [31], FaceNet [32] and SphereFace [33].

B. Cloud-based Deployment Strategies

Cloud-based architectures could represent a big challenge for federated learning scenarios, if from one hand we have to deal with the privacy and security of the data the cloud could rappresent an huge advantage for scalability and redundancy of user's data. The described architecture leverages the problem of local computation introducing a distributed scenario totally hosted on cloud.

This approach consider local training servers hosted on cloud which will be utilized to sink data coming from local cameras into the household. Each of the local server will then contribute to the global weights computation, following the classical schema of hierarchical federated learning. Since each of the local server will be utilised as endpoint only by the cameras of a given environment, this architecture allows the instantiation of each of them in a separate cloud instance.

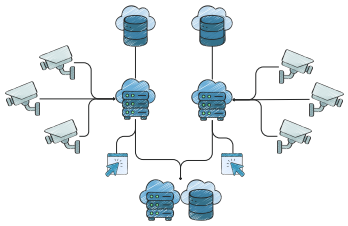

Fig. 2. This approach utilizes local training servers hosted in the cloud to gather data from household cameras. Local servers contribute to the global model by sending computed weights to the central cloud server. High scalability and model flexibility are strong suits of this architecture. Point of failures of this schema are given by possible data leaks and communication delays. This architecture is suitable for environments requiring large-scale, complex model deployment with centralized data processing.

This scenario's primary benefit is the scalability provided by cloud support and the ability of hosting the entire architecture without the specific requirement for local host computers. Conversely, using such a system to store the data itself may lead to a real loss of confidentiality and privacy that were previously provided by the federated learning approach.
Basically, the latter scenario is highly appropriate for potential traffic interception and signal jamming threats that can arise during the transmission of cameras to private cloud servers. Furthermore, the mandatory decision to store facial photos on cloud servers may lead to potential data leaks by cloud providers or, once more, by potential attackers.
From a machine learning perspective, analyzing the proposed architecture reveals key benefits associated with this cloud-based approach. The inherent scalability of the cloud environment enables a more flexible choice of underlying models. Unlike traditional settings constrained by energy consumption limitations, this scenario allows for the deployment of more accurate models. These advanced models often require a

greater number of convolutional layers, leading to improved performance.
Furthermore, the cloud architecture facilitates greater control over data distribution. This can help mitigate the challenge of non-IID (non-independent and identically distributed) data, a common issue in federated learning systems due to the inherent diversity of participant datasets.
For what concerning the possible disadvantages of this kind of approach under the light of machine learning perspective, we have to consider eventually the latency produced by the remote instantiation of inference servers. Since, in the scenario proposed, there is no local entity capable of performing such task, each inference process should be sent to a remote machine, which could originate some delays influenced by the geographical collocation of the datacenter assigned.
Under the hierarchical federated learning perspective, we propose a possible architecture for the face recognition task which implements such canonical structure, with the introduction of a cloud-based weights agglomeration server.

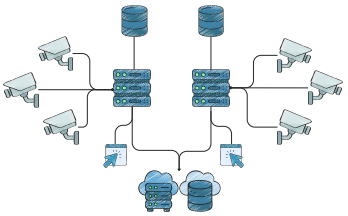

Fig. 3. Local servers handle data storage and processing. Encrypted data is stored locally, reducing transmission risks. Two principal advantages are that data remains on local devices, enhancing security and speed-up of the application due to local processing capabilities. On the other side, this schema has to face challenges such as the problem of scaling for larger environments and continuous local training and cloud communication increase energy use. This architecture is suitable for small to medium scale deployments and environments where data privacy and low latency are critical.

This architecture offers potential benefits for data privacy by storing facial recognition data in local databases. This approach allows for encryption at the local network level, potentially mitigating some security risks associated with data transmission. However, it's crucial to acknowledge that local storage doesn't guarantee absolute security. Breaches of local databases can still occur.
While local storage might enhance perceived security, the local instantiation of the training model introduces significant limitations in terms of scalability and real-time performance. Scaling the service to handle larger environments becomes a challenge, potentially introducing bottlenecks that hinder real-time face recognition.
Furthermore, a 24/7 service model with continuous local model training and communication with the cloud can lead to high energy consumptions. This raises concerns about the long-term sustainability of such a system, especially for the end-user in terms of operational costs and valuable impact.
The proposed architecture needs a careful selection of the facial recognition model based on the computational capabilities of the local server hardware. This approach can significantly improve inference speed at individual server locations. However, it introduces a potential bottleneck in training speed. Local training can have a detrimental impact on the overall time required to aggregate local models into the global one.
The last consideration to make in this possible deployment scenario considers the cloud utilisation made. The latter strategy in fact doesn't really leverage the real power of a cloud architecture, considering such powerful technology only as a communication mean between different local data providers.
Including such sensitive operations such as model training procedure into a cloud-based architecture could arise some concerns and, in some way, it could be considered as a canonical trade-off choice between privacy and security of data and actual computational speed and scalability. Those aspect will be further analized in the hybrid proposed implementation.

1) Hybrid Deployment strategy: The last architecture proposed takes into consideration the downsides of both the previous approaches to find a possible trade-off between privacy and scalability of the service. In fact, the privacy concerns with possible data leakage of the cloud providers was the main disadvantage of the cloud-based solution, meanwhile for the edge-based solution the scalability was the innate issue to solve.
The hybrid solution tries to solve both those aspects by introducing a "proxy" local server between the cloud-based server and the cameras. This proxy server will be aimed to store the embedded face to compute and query the cloud server for inference tasks.
Indeed, such solution doesn't solve the need of sending sensible data to the cloud architecture but integrates the inference step into the local environment. For what concerning the edge-based issues previously exposed, the hybrid solution allows the same scalability given by the cloud-based scenario. The proposed hybrid solution enhances the local deployed consumer instance, allowing the local partition of the architecture to perform light operations like inference or anomaly detection while keeping all the advantages of the classical federated learning in training and communicating with the global agglomeration server.

TABLE I
CHARACTERIZATION OF THE DEPLOYMENT STRATEGIES

	Security	Privacy	Scalability	Energy
Edge	high	high	low	low
Cloud	medium	low	high	medium
Hybrid	medium	medium	high	high

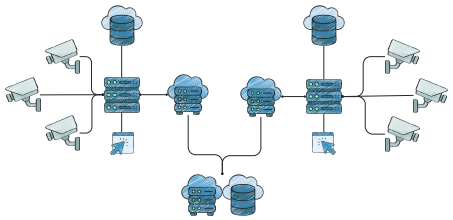

Fig. 4. This solution provides intermediate servers (proxy) between cameras. Cloud infrastructure manages data storage and processing, while cameras capture images/videos and send data to local proxy servers. Each proxy handles light tasks like initial processing and inference, then sent to cloud for computation and training. This architecture combines edge privacy with cloud scalability, offering a middle ground. Local servers manage light operations, reducing cloud dependency for frequent tasks. This architecture is ideal to balance privacy, scalability, and computational efficiency, such as large enterprises or hybrid work environments.

C. Hierarchical Federated Learning for Face Detection

Hierarchical Federated learning is a possible extension of classical federated learning. It is composed of a master aggregator and a hierarchy of multiple aggregators that combine trained local models [34].
This demonstrates to be particularly useful in the context of face recognition for the nature of camera devices as streaming-only devices with absent or low computational capability.
The architecture of the choosen approach is based on the first scenario described in the previous section.
Cameras are the client level where images are captured and sent through the network to Edge Servers hosted on cloud.
Edge Servers' task is to preprocess images containing faces, local face detection is performed and trained on these data.
At the end of the training, obtained parameters are sent to master node hosted on Cloud using network.
The master node, once it has received parameters, performs a global aggregation using Federated Averaging (FedAvg).
FedAvg is built starting from stochastic gradient descent (SGD) and it is applied computing a single batch gradient per round of communication. [7]
The goal of FedAvg is to minimize a global loss function $F(w)$, which is a weighted average of the local loss functions of each Edge Server $F_k(w)$:

$$F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \tag{1}$$

where n_k is the number of data of the edge server k and n is the total number of data from all edge servers.

Each Edge Server k updates model parameters w using data coming from cameras through the Stochastic Gradient Descent. The updating of a general local parameter w_k is given by:

$$w_k^{t+1} = w_k^t - \eta \nabla F_k(w_k^t) \tag{2}$$

Once the local training epochs are terminated, the master server aggregates these updated models parameters:

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1} \tag{3}$$

The two phases, local trainings and aggregation are executed iteratively. [7]

IV. Experiments

A. Model Used

We considered FaceBoxes as face detection model, introduced by Zhang et al. [26]
FaceBoxes provides speed and efficiency thanks to its capability to be suitable for real-time applications. It is optimized for mobile and embedded devices facing a lack of computational resources with high detection speeds on standard CPUs.
FaceBoxes' architecture is based on:

- **Rapidly Digested Convolutional Layers (RDC)**: these layers are devoted to maintaining representations of high-level features while the spatial dimensions of the input images are reduced
- **Multiple Scale Convolutional Layers (MSC)**: The role of these layers is to manage different sizes of faces by processing features at multiple scales
- **Context Anchors**: To capture contextual information around faces, anchors' help is necessary because they allow for better detection accuracy, in particular for small faces.

The choice of a lightweight model allows the possibility of training on low-end devices, which could be potentially the case when considering an edge-computing paradigm.

To train FaceBoxes, a large-scale face detection dataset is used, obtaining performances consistent with the state-of-the-art in terms of accuracy while leaving the speed of real-time detection unchanged.
In order to implement a federated training, we relied upon existing federated learning frameworks. Flower [35] is a federated learning framework, built to support heterogeneous edge devices and carry out extensive FL experiments. It is characterized by its easy of use and provides various machine learning libraries and possibility to be extended to accommodate custom requirements.

B. Datasets

WIDER FACE [36] is a well-known face detection dataset, commonly used for benchmarking face detection applications. It consists of 32,203 images split into training, validation and testing sets following a 40-10-50 distribution, respectively. There are only 12,881 samples that have annotations and so, valid for training our model in a supervised manner. This data set is made up of 61 event categories which represent a quite large variation in scale, pose as well as occlusion.

For testing purposes, three different types of datasets are used:

- PASCAL: It is part of the PASCAL Visual Object Classes Challenge and includes a variety of images with annotated objects, making it suitable for evaluating object detection algorithm. [37]
- FDDB: The Face Detection Data Set and Benchmark (FDDB) contains 2,845 images with a total of 5,171 annotated faces, providing a comprehensive set for evaluating face detection performance under various conditions. [37]
- AFW: The Annotated Faces in the Wild (AFW) dataset consists of 205 images with 468 faces, annotated with facial landmarks, head poses, and occlusion labels, offering a challenging dataset for face detection tasks. [38]

At the testing phase, we just make inference using the previously trained model, without considering existing annotations.

C. Model Training

At each round, each client is responsible of training its own local model using just a subset of the annotated images of the WIDER FACE dataset for a given number of local epochs (in our case, one). The training process leverages the FaceBoxes model, which minimizes the Multibox Loss [27], which is described in the following equation:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \qquad (4)$$

where N is the number of matched default boxes. The localization loss is a *Smooth L1 Loss* between the predicted box (l) and the ground truth box (g) parameters [27].

At the end of each round, local model's parameters are then averaged by the server.

The training is organized into three clusters, each with a different number of workers and master nodes, in order to test the validity of the federated approach in the context of federated face detection. We trained the models until convergence, which in our case was obtained after about 130 rounds.

D. Partition Strategy

In the given FL environment, each client participates in the training process by training on a segment of the dataset individually, ensuring that an equal number of samples is assigned to each of them in the study. The dataset is split into multiple subsets with an external script where each split consisting of a collection of image files and their associated annotations files. Here, from the original data set, we form n subsets that are distributed among the clients. We assume that the portion on which a client runs the training of its own model is already placed on the corresponding machine.

E. Client Setup

The setup process begins with each client typing a shell command in response to which an appropriate copy of the assigned partition of the dataset is made from a central server into that client's local directory and then unzipped. This makes it possible for each of the clients to have the right dataset for learning by ensuring that appropriate encryption and transmission security codes are used. In this regard, the above setup is followed, and the client script is run, which allows the client to integrate into the federated learning process with help from Flower. The integration with Flower, which helps avoid significant barriers to communication between clients and the master node that are associated with the coordination of model training updates.

F. Cloud Platform

We deployed and managed our infrastructure using Google Cloud Platform (GCP) in our federated learning experiments. GCP offers a stable and expandable setting ideal for carrying out complex machine learning tasks.[39]

Experimental Configurations

We conducted our tests using three different configurations:

- One master node and eight worker nodes.
- One master node and four worker nodes.
- One master node and two worker nodes.

Master Nodes

- Type: E2 Standard
- Specifications: 2 vCPUs (1 core), 8 GB RAM.
- Storage: 100 GB Standard Persistent Disk.
- Role: Responsible for aggregating and merging the weights from the worker nodes, an essential task in federated learning that requires less computational power compared to the training process.

Worker Nodes

- Type: N2 Standard 2
- Specifications: 2 vCPUs (1 core), 8 GB RAM.
- Storage: 100 GB Standard Persistent Disk.
- Role: Perform the computationally intensive training tasks, necessitating more powerful CPUs compared to the master nodes.

It's interesting to note that we only considered CPU-based processing capability for our investigation and did not use instances with GPUs. This setup demonstrates how well GCP's CPU resources can process workloads including federated learning.

V. Results and Discussion

In this section we will analyze the results produced by the federated model employed; in particular we will shread some light regarding the accuracy and the scalability of the approach. The different results presented are based on different datasets used in phase of testing: Pascal, FDDB, and AFW.

A. Loss Analysis in cloud-based environment

As shown in Fig. 5, the inference results obtained after the training phase can be fully comparable to a classical centralized trained model with a fraction of time employed to produce such results.

Fig. 5. The images shown above illustrate the results of the bounding boxes for face detection and correspondent confidence measure. These images were extracted from detections perfomed by models trained in each experiment on the proposed testing datasets. **From left to right:** AFW, FDDB, PASCAL, respectively. **From top to bottom:** face detections produced by the model using 2, 4 and 8 clients, respectively.

Over the visual analysis of the detections produced, some statistical considerations can be made: the agglomeration procedure of the federated learning approach positively impacted the measurements, allowing a faster convergence of the averaged global model.
Such results can be explained from a high-level point of view by the heterogeneously of different client which allowed the specialization on different recording envariaments prompting the global model to correct himself in case of misclassification from non-specialized units. From a data-driven point of view, as presented in Fig. 6, the measurement of the loss presents a monothonic behavior towards the convergence point. Such results were taken during the phase of training by considering a 8-workers node configuration which was, in line of theory, the one more prone to a greater loss.
In particular, after a total amount of 130 rounds of federated training the model was capable of reaching a loss of $\simeq 6.07$ which can be considered a good result, especially considering the low-end configuration proposed. Even if not explicitly shown, this behavior is presented also when considering the other master-workers configurations.

For what concerning the results produced regarding the loss measurements on a lower configuration of client, such behavior persists with a slower descend towards the results

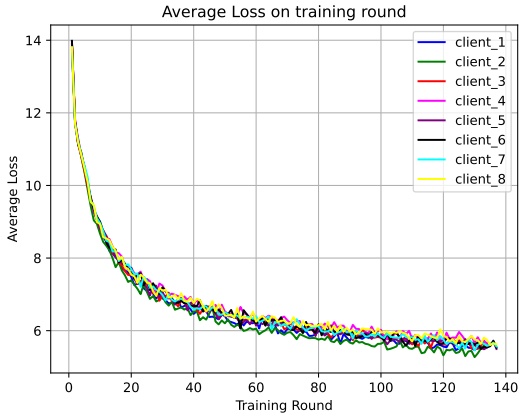

Fig. 6. Average loss computed per training round according to the 1-Master 8-Workers configuration. Clearly, each client converges towards the same solution of the optimization problem, even though they are trained in a federated manner and considering an IID distribution of the dataset among clients.

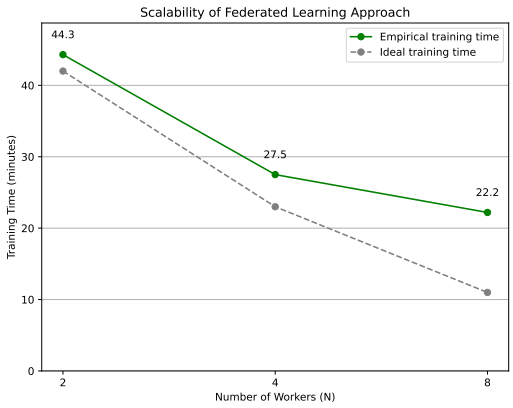

Fig. 7. In order to test the scalability of the federated learning approach, a plot of the average training time per round vs. number of workers is provided. The dashed gray line might resemble an "ideal" curve if the training was performed on an homogeneous cluster, where the communication overhead could be considered negligible. The plotted green line corresponds to the empirical training time per round measured. As we could note, increasing the number of workers, the training time per round decreases accordingly. However, at the same time, also the communication overhead among server and client increases. This bounds the potential gain of considering a large number of workers in a federated configuration. Moreover, network delays and workers slower than others heavily affect the behavior of the green curve.

obtained by the 8 clients based model.

B. Training time and Scalability

In the following subsection we will analyze the behavior of the model and, in particular, the training time parameter in relation to the horizontal scaling of the federated learning network. Such results are proposed in this paper to allow a possible formulatition for a better configuation to employ in real case scenarios.

The employed setting considers a 2, 4 and 8 clients case scenarios each one with the same characteristics described in the forth section. After the end of the training phase for each of those configurations, the average training time per round was extracted and compared with the others cases to produce the chart shown below (Fig. 7).

The highly scalable architecture proposed would auspicate for a possible curve similar ideal shown above (gray curve) but the actual results produced suggest an actual stationary behavior with an increasing number of clients.

Such results show a fairly small decreasing in training time from a 4 based client configuration to the one based on 8 clients, also considering the major difference between training set on which they are delegated to train. Those results obtained could also be described to the client's dataset split dimension which can in turn represent a possible thread in real case scenarios.

On the other hand, a possibile solution to such problem could be implemented by leveraging the actual cloud configuration. In fact, the actual hosting of client nodes in a cloud-based architecture allows the creation of different sub-clusters, increasing the amount of data fed to each of them but, at the same time, exploiting the scalability of an arbitrary dimension of the cluster.

VI. Future Works

Building upon the foundational work presented in this paper, several directions for future research are proposed to enhance the federated learning framework and its applications in face recognition.

Extension to Face Recognition: extension to Face Recognition: Face recognition applications may take advantage of the described federated learning solution. With this addition, sensitive biometric data handling will be handled more effectively and privately by incorporating algorithms for face recognition into the current federated learning architecture.

Complex Federated Learning Approaches: more in-depth federated learning techniques or sophisticated aggregation mechanisms may be the subject of future study.

As for federeted learning techniques, these might include meta-learning based federated learning [16], federated multi-task learning [40] and federated asynchronous [41]. About aggregation methods these could be Secure Aggregation [42], Federated Learning with Differential Privacy [43] or Adaptive Federated Learning [44].

Comparative Analysis of Models: Finding the model that best fits the data in terms of accuracy, performance, inference time, training and complexity, and other factors requires a comparative examination of different models. Since they are designed for resource-limited contexts and can provide a better trade-off between performance and resource consumption, small and mobile models should be considered in future research.

Edge-Based vs. Cloud-Based Solutions: a thorough comparison of edge-based and cloud-based solutions is necessary to

understand the performance implications of each approach. This comparison should include metrics such as latency, bandwidth usage, scalability, and privacy. The insights gained from this analysis will help in determining the most suitable deployment strategy for various real-world scenarios.

GPU experiments: in order to speed up the training process, GPUs should be used in future studies. The performance of federated learning models can be improved and training time can be greatly decreased by using GPU devices. Federated learning framework optimization will benefit greatly from a comparison of GPU-based training to CPU-based training in terms of both performance and cost-efficiency.

Implementation of a True Hierarchical Structure: our current implementation uses a simplified cloud-native architecture where both local and global servers are hosted in the cloud, and data generators (e.g., cameras) are integrated with local servers (the FL workers). Actually, data should be generated by external entities (e.g., edge devices like cameras) and sent to local cloud servers (the FL workers). Future work should aim to implement a real-world scenario where data-generating devices exist outside the cloud, transmitting data to local cloud servers. This approach would better reflect a true hierarchical federated learning structure and enhance the realism and applicability of the model.

By addressing these areas, future work can significantly contribute to the advancement of federated learning, particularly in the domain of face recognition, ensuring enhanced privacy, scalability, and performance.

VII. Conclusion

Our study explored the application of federated learning in the context of face recognition. A significant focus was placed on cloud-based deployment strategies, which provide substantial benefits in terms of scalability and redundancy of federated systems. By hosting local training servers on the cloud, we can efficiently process data from household cameras and facilitate the contribution of local servers to the global model weight computation. This architecture not only supports high scalability and model flexibility but also introduces potential challenges such as data leaks and communication delays.

To mitigate these issues, a hybrid deployment strategy was proposed, which balances privacy and scalability. This involves introducing a local proxy server between the cloud-based server and the cameras. The proxy server allows for local storage and processing of data, reducing dependency on cloud services for frequent tasks and addressing some of the privacy concerns associated with a purely cloud-based solution.

In terms of results, our investigation demonstrated that federated learning can achieve performance comparable to centralized training while significantly reducing training time. The federated learning setup proved efficient, with each client converging towards a similar solution. The model reached a satisfactory loss value within 130 training rounds, showcasing the approach's viability even with a low-end configuration.

The scalability of the federated learning architecture was also notable. Although the ideal scenario predicted by theoretical models was not fully achieved due to communication overhead and network delays, empirical results showed that training time per round decreased as the number of workers increased. This indicates the potential for further optimization in real-world deployments.

In conclusion, the federated learning approach presented in this study offers a scalable, efficient, and privacy-preserving solution for face recognition. With further research and optimization, this framework can be extended to more complex tasks and real-world applications, paving the way for broader adoption of federated learning in various domains.

Code availability

In order to reproduce the experiments discussed so far in the current submission, software implementation is available at *https://github.com/gomax22/Federated-FaceBoxes*.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7488250

[2] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-Time Video Analytics: The Killer App for Edge Computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8057318

[3] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey," *IEEE Access*, vol. 8, pp. 58 322–58 336, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9044329

[4] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, "Bringing the cloud to the edge," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 346–351. [Online]. Available: https://ieeexplore.ieee.org/document/6849256

[5] J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8763885

[6] P. M. Mammen, "Federated Learning: Opportunities and Challenges," Jan. 2021. [Online]. Available: http://arxiv.org/abs/2101.05428

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," Jan. 2023. [Online]. Available: http://arxiv.org/abs/1602.05629

[8] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, Feb. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X20329848

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020. [Online]. Available: http://arxiv.org/abs/1908.07873

[10] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A Survey on Security and Privacy Issues in Edge-Computing-Assisted Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4004–4022, Mar. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9163078

[11] Y. Niu and W. Deng, "Federated Learning for Face Recognition with Gradient Correction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, pp. 1999–2007, Jun. 2022. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20095

[12] D. Aggarwal, J. Zhou, and A. K. Jain, "FedFace: Collaborative Learning of Face Recognition Model," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*. Shenzhen, China: IEEE, Aug. 2021, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/9484386/

[13] F. X. Yu, A. S. Rawat, A. K. Menon, and S. Kumar, "Federated Learning with Only Positive Labels," Apr. 2020. [Online]. Available: http://arxiv.org/abs/2004.10342

[14] C.-T. Liu, C.-Y. Wang, S.-Y. Chien, and S.-H. Lai, "Fedfr: Joint optimization federated framework for generic and personalized face recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 1656–1664, issue: 2. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20057

[15] C. N. Duong, T.-D. Truong, K. G. Quach, H. Bui, K. Roy, and K. Luu, "Vec2Face: Unveil Human Faces from their Blackbox Features in Face Recognition," Mar. 2020. [Online]. Available: http://arxiv.org/abs/2003.06958

[16] Q. Meng, F. Zhou, H. Ren, T. Feng, G. Liu, and Y. Lin, "Improving Federated Learning Face Recognition via Privacy-Agnostic Clusters," Jan. 2022. [Online]. Available: http://arxiv.org/abs/2201.12467

[17] A. Koubaa, A. Ammar, A. Kanhouch, and Y. AlHabashi, "Cloud versus edge deployment strategies of real-time face recognition inference," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 143–160, 2021, publisher: IEEE. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9350171/

[18] V. Patel, S. Kanani, T. Pathak, P. Patel, M. I. Ali, and J. Breslin, "An Intelligent Doorbell Design Using Federated Deep Learning," in *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*. Bangalore India: ACM, Jan. 2021, pp. 380–384. [Online]. Available: https://dl.acm.org/doi/10.1145/3430984.3430988

[19] M. O. Oloyede, G. P. Hancke, and H. C. Myburgh, "A review on face recognition systems: recent approaches and challenges," *Multimedia Tools and Applications*, vol. 79, no. 37-38, pp. 27 891–27 922, Oct. 2020. [Online]. Available: https://link.springer.com/10.1007/s11042-020-09261-2

[20] S.-H. Lin, "An introduction to face recognition technology," *Informing Sci. Int. J. an Emerg. Transdiscipl.*, vol. 3, pp. 1–7, 2000. [Online]. Available: http://inform.nu/Articles/Vol3/v3n1p01-07.pdf

[21] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, Dec. 2001, pp. I–I. [Online]. Available: https://ieeexplore.ieee.org/document/990517

[22] J. Li and Y. Zhang, "Learning SURF Cascade for Fast and Accurate Object Detection," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2013, pp. 3468–3475. [Online]. Available: https://ieeexplore.ieee.org/document/6619289

[23] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face Detection Based on Multi-Block LBP Representation," in *Advances in Biometrics*, S.-W. Lee and S. Z. Li, Eds. Berlin, Heidelberg: Springer, 2007, pp. 11–18.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jan. 2016. [Online]. Available: http://arxiv.org/abs/1506.01497

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," May 2016. [Online]. Available: http://arxiv.org/abs/1506.02640

[26] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "FaceBoxes: A CPU Real-time Face Detector with High Accuracy," Dec. 2018. [Online]. Available: http://arxiv.org/abs/1708.05234

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," 2016, vol. 9905, pp. 21–37, arXiv:1512.02325 [cs]. [Online]. Available: http://arxiv.org/abs/1512.02325

[28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016. [Online]. Available: http://arxiv.org/abs/1604.02878

[29] Y. Feng, S. Yu, H. Peng, Y.-R. Li, and J. Zhang, "Detect Faces Efficiently: A Survey and Evaluations," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 1, pp. 1–18, Jan. 2022. [Online]. Available: http://arxiv.org/abs/2112.01787

[30] Y. Sun, X. Wang, and X. Tang, "Deep Learning Face Representation by Joint Identification-Verification," Jun. 2014. [Online]. Available: http://arxiv.org/abs/1406.4773

[31] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 1701–1708, iSSN: 1063-6919. [Online]. Available: https://ieeexplore.ieee.org/document/6909616

[32] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823, arXiv:1503.03832 [cs]. [Online]. Available: http://arxiv.org/abs/1503.03832

[33] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep Hypersphere Embedding for Face Recognition," Jan. 2018. [Online]. Available: http://arxiv.org/abs/1704.08063

[34] O. Rana, T. Spyridopoulos, N. Hudson, M. Baughman, K. Chard, I. Foster, and A. Khan, "Hierarchical and Decentralised Federated Learning," Apr. 2023. [Online]. Available: http://arxiv.org/abs/2304.14982

[35] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, and N. D. Lane, "Flower: A Friendly Federated Learning Research Framework," Mar. 2022. [Online]. Available: http://arxiv.org/abs/2007.14390

[36] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A Face Detection Benchmark," Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.06523

[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4

[38] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 2879–2886, iSSN: 1063-6919. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6248014

[39] P. Garraghan, P. Townend, and J. Xu, "An Analysis of the Server Characteristics and Resource Utilization in Google Cloud," in *2013 IEEE International Conference on Cloud Engineering (IC2E)*, Mar. 2013, pp. 124–131. [Online]. Available: https://ieeexplore.ieee.org/document/6529276

[40] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated Multi-Task Learning," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1705.10467

[41] Y. Xie, L. Ding, A. Zhou, and G. Chen, "An Optimized Face Recognition for Edge Computing," in *2019 IEEE 13th International Conference on ASIC (ASICON)*, Oct. 2019, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/8983596

[42] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. Dallas Texas USA: ACM, Oct. 2017, pp. 1175–1191. [Online]. Available: https://dl.acm.org/doi/10.1145/3133956.3133982

[43] R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective," Mar. 2018, arXiv:1712.07557 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1712.07557

[44] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 7611–7623. [Online]. Available: https://doi.org/10.48550/arXiv.2007.07481