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

R. Esposito, V. Mele, A. Mungari, M. Giordano Orsini, M. Roscica, S. Verrilli

Cloud Computing Project Presentation

Edge Computing

Edge computing emerges as a fundamental development in the distributed computing scenario.

It offers multiple benefits, as:

- Reduces Latency: Computations are closer to data sources.
- Optimizes Bandwidth: Less data sent to central servers.
- Enhances Privacy: data handled locally, reducing exposure risks.

Federated Learning

Federated Learning is a technique that implements collaborative model training among multiple edge devices (clients).
Most relevant advantages are:

- No raw data sharing, maintains data privacy
- Collective learning capability, improving scalability
- Enhancement of security by keeping data localized, also more privacy for ML application.

FL addresses risks such as data interception, unauthorized access, and centralized database breaches.
However, edge devices' limited computational and storage capabilities can pose security vulnerabilities.

Related Works - 1

Face recognition using a hierarchical FL approach has seen various implementations in the literature, aiming to address the well-known issues of traditional FL. Below are some key technologies:

- **FedGC**: Innovative architecture to improve privacy in federated learning for face recognition .The idea is to manipulate the gradients with a cross-client gradient term based on softmax in order to avoid a reconstruction of the data starting from the gradients.
- **FedFace**: FL framework for training a privacy-aware face recognition model. Rather than using the original faces, it uses an average of facial features to represent identities, keeping representations of people clearly distinct. The main feature is that it makes facial recognition more secure and privacy-friendly.

Related Works - 2

- **PrivacyFace**: Communicate privacy-sensitive information using Differentially Private Local Clustering (DPLC) and consensus-aware recognition loss for better feature discrimination.
- **FedFR**: Jointly optimizes generic and personalized face recognition models with techniques like hard negative sampling and contrastive regularization.

Other works

And other works:

Patel et al. propose a smart doorbell using Federated Deep Learning for video analytics across Edge and Cloud resources, demonstrating the potential of Federated Learning (FL) in distributed video processing applications. **Koubaa et al.** explore cloud versus edge deployment strategies for real-time face recognition, highlighting the benefits of local computation in edge devices for reducing latency, and **Solomon et al.** address privacy and data transmission issues by using Generative Adversarial Networks (GANs) to generate fake data at the edge, enabling Federated Learning without the need for raw data transmission.

Background on Face Detection and Recognition

Face detection and recognition are popular tasks which involve respectively the location and the identification of faces by computer systems within multimedia data(images, videos..) exploiting facial features, geometric relationships.

- Modern systems use various visual devices, such as video cameras (e.g., camcorders), infrared cameras, and 3D scanners
- Traditional methods involve:
  - Face Detection: Cascaded AdaBoost classifier, Viola-Jones algorithm using Haar-like features, Speeded Up Robust Feature (**SURF**), and Multi-Block Local Binary Patterns **(MB-LBP**)
  - Face Identification: Faster R-CNN, YOLO, Single Shot Multibox Detector (**SSD**), Multi-task Cascaded Convolutional Networks (**MTCNN**)

Cloud-Based Deployment Strategy

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.

Advantages:

- High scalability
- Model flexibility

Challenges:

- Privacy risks: possible data leaks
- Latency issues: communication delays

Suitable for environments requiring large-scale, complex model deployment with centralized data processing.

Cloud-Based Deployment - Figure

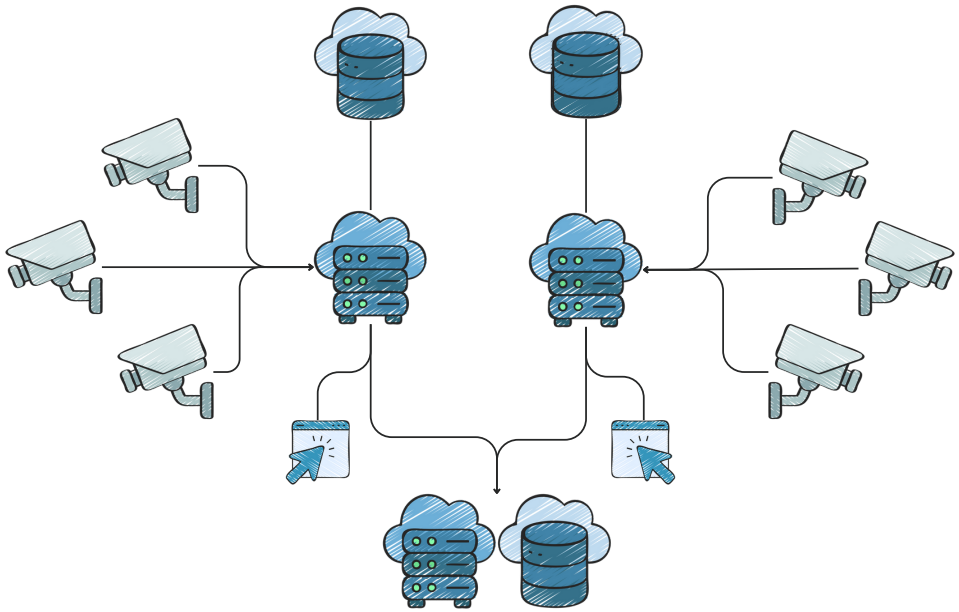


Figure 1: Sketch of the 1st scenario.

Edge-Based Deployment Strategy

Local servers handle data storage and processing. Encrypted data is stored locally, reducing transmission risks.

Advantages:

- Data Privacy: Data remains on local devices, enhancing security.
- Inference Speed: Faster due to local processing capabilities.

Challenges:

- Scalability: Difficult to scale for larger environments.
- Energy Consumption: Continuous local training and cloud communication increase energy use.

Suitable for small to medium scale deployments: Suitable for environments where data privacy and low latency are critical.

Edge-Based Deployment - Figure

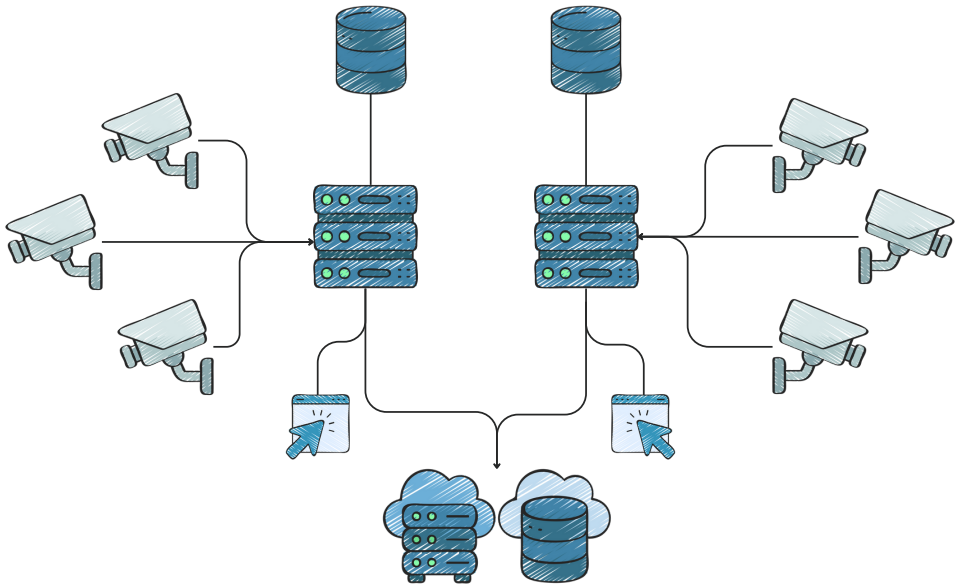


Figure 2: Sketch of the 2nd scenario.

Hybrid Deployment Strategy

Has intermediate servers (proxy) between cameras and the cloud manage data storage and processing.
Cameras capture images/videos and send data to local proxy servers. Each proxy handle light tasks like initial processing and inference, then sent to cloud for computation and training.

Advantages:

- Combines edge privacy with cloud scalability, offering a middle ground.
- Local servers manage light operations, reducing cloud dependency for frequent tasks.

Ideal to balance privacy, scalability, and computational efficiency, such as large enterprises or hybrid work environments.

Hybrid Deployment - Figure

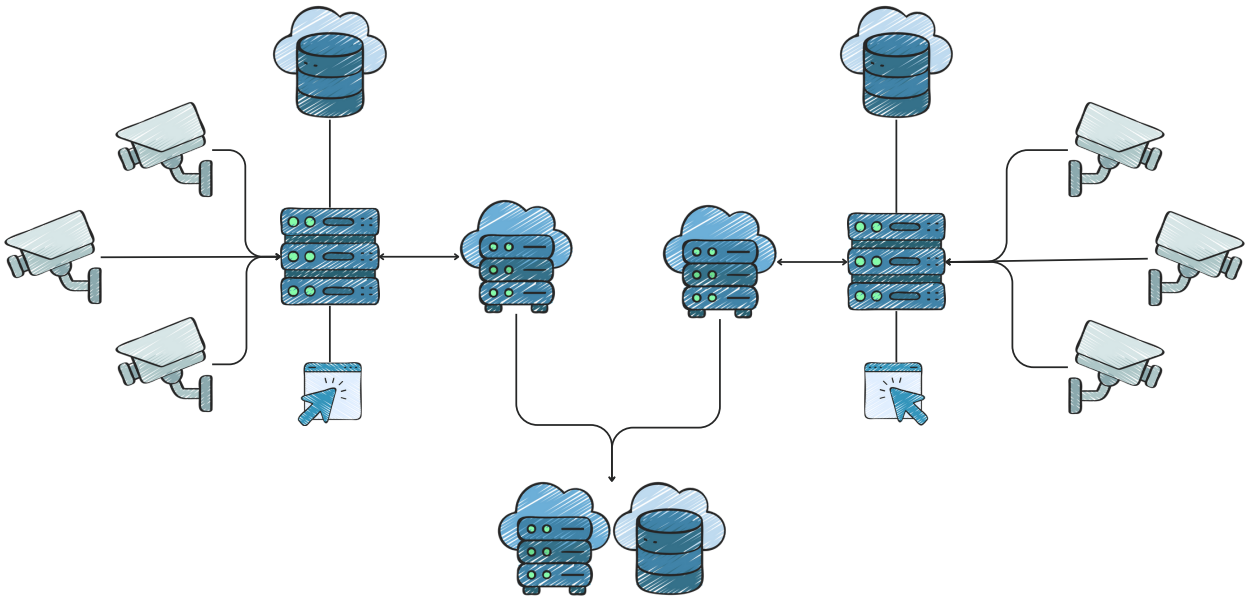


Figure 3: Sketch of the 3rd scenario.

Cloud-Based Federated Deployment (1st Scenario)

Recalling the first scenario:

- Cameras are the client level
  - images are captured and sent through the network to Edge Servers hosted on cloud.
- Edge Servers' task
  - preprocess images containing faces, local face detection model is trained on those data for a given number of local epochs.
- At the end of the training, local parameters are sent to the master node hosted on cloud
- The master node performs a global aggregation using **Federated Averaging** (FedAvg).

At the next round, all clients will start from the same global model, provided by the master node.

Experimental setup

The followings are the details of our implementation:

- **Model**: FaceBoxes, proposed by *Zhang et al.*
- **Datasets**: WIDER FACE (12800 annotated images) for training; PASCAL, FDDB, AFW for testing.
- **Learning Strategy**: Supervised training, Federated Learning using Flower, FedAvg, SSD MultiBox Loss.
- **Cloud Platform**: Google Cloud Platform.
- **Evaluation Metrics**: Execution time, loss curves, visual inspection of detected faces.

In order to simulate a real-world scenario, each local model is trained on an subset (iid) of WIDER FACE

- subsets are distributed before starting the federated training.

FedAvg

FedAvg is built starting from stochastic gradient descent (SGD) and it is applied computing a single batch gradient per round of communication.

- minimize a global loss function $F(w)$, which is a weighted average of the local loss functions of each Edge Server $F_k(w)$:

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

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

FedAvg

- Each Edge Server k updates model parameters w using data coming from cameras through the Stochastic Gradient Descent.

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

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

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

- local training and aggregation are executed iteratively.

Single Shot Detector MultiBox Loss

Given an annotated image (label and ground truth bounding boxes), **SSD MultiBox Loss** is made up of two terms:

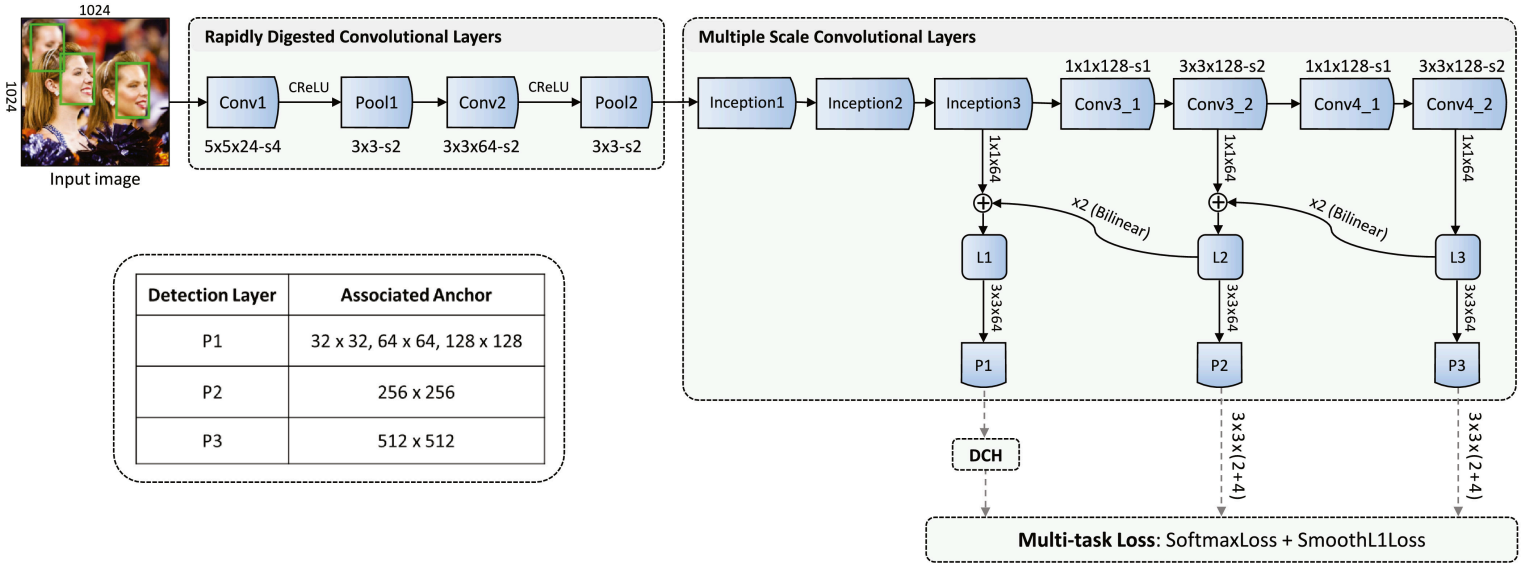
- **Confidence loss**: categorical cross-entropy loss for classifying the detected objects.
- **Location loss**: regression loss (Smooth L1 Loss) on the parameters of the detected bounding box.

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

- N is the number of matched default boxes.
- predicted box (l) and the ground truth box (g) parameters.
- α is a regularization constant.

FaceBoxes

Lightweight model, well-optimized for real-time applications and suitable for mobile and embedded devices.



Detection Layer	Associated Anchor
P1	32 x 32, 64 x 64, 128 x 128
P2	256 x 256
P3	512 x 512

Figure 4: Architecture of FaceBoxes: Rapidly Digested Convolution Layers (RDCL), Multiple Scale Convolution Layers (MSCL) and Divide and Conquer Head (DCH).

Experimental Configurations and GCP

Following details explain the configuration of the test experiments:

- Cloud Platform: Google Cloud Platform (GCP)
- 1 Master Node:
  - Type: E2 Standard
  - Specs: 2 vCPUs, 8 GB RAM, 100 GB storage[1]
  - Role: Aggregates and merges weights.
- N Worker Nodes (tested for N=2, 4, 8):
  - Type: N2 Standard 2
  - Specs: 2 vCPUs, 8 GB RAM, 100 GB storage[1]
  - Role: Handles computationally intensive training tasks.
- Code Availability: `https://github.com/gomax22/Federated-FaceBoxes`

[1] Only CPU-based processing used to demonstrate GCP's capability for federated learning workloads.

Visual Results

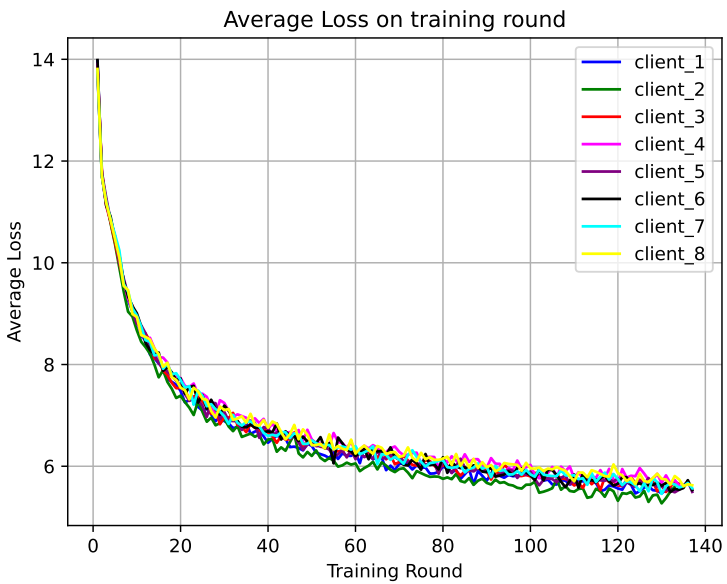
In following images is shown bounding boxes around detected faces and the corresponding confidence measure.

From left to right: AFW, FDDB, PASCAL.
From top to bottom: face detection produced with 2, 4 and 8 workers.

- Results are comparable to a centralized model but with a significant speedup.
- Agglomeration approach of FL has positive impact on measurements, allowing **faster convergence** of the averaged global model.

Average Loss

In the figure shown below, the average loss per training round for the 8-workers configuration.

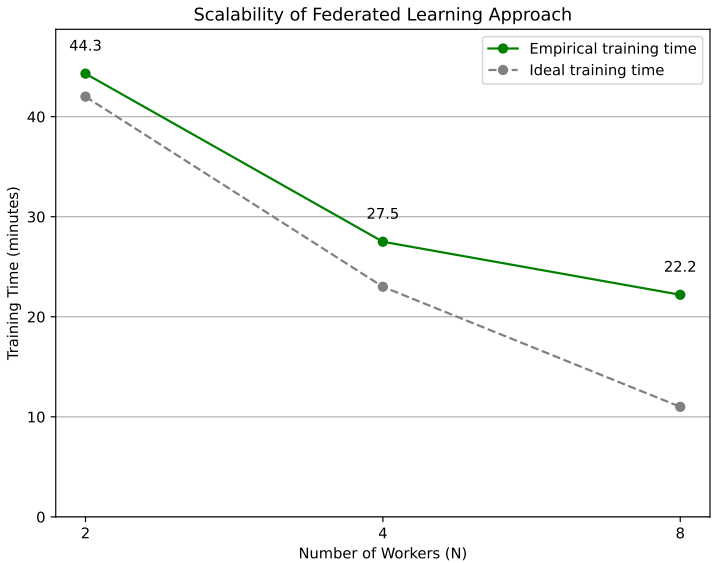


- Measurement of the loss shows a monotonic behaviour toward the convergence point.
- The other configurations' loss show the same behaviour with a slower descent.

After 130 rounds, the average loss is ~ 6.07, good enough under the implementation premises.

Training Time

Time per round is shown in the figure below, for each configuration to analyze model scalability.



- Empirical t.t. decreases for larger values of N, as expected, but also communication overhead should be considered.
- Above the "*ideal*" t.t. line, it reaches stationary behaviour for 8 workers.

It reveals practical limits in reducing training time with increasing N workers due to network and dataset distribution challenges.
A cloud-based with sub-clusters architecture, exploiting both scalability and data splitting can enhance this aspect.

Future Research

The proposed method offers several opportunities for enhancement, as:

Incorporating **face recognition** into federated learning can improve the privacy and handling of bio-metric data by using advanced algorithms.

Research can explore more **complex FL** techniques like meta-learning for improved personalization, multi-task learning for handling multiple tasks, and asynchronous learning for flexible client participation.

Also it should investigate secure methods for **data aggregation**, like differential privacy and adaptive learning, to enhance security and address inconsistencies.

Comparison

Nonetheless, comparing models for accuracy, performance, inference time, training, and complexity is essential.

Future research should focus on small, **mobile models** that are optimized for resource-limited environments, balancing performance and resource usage.

Use **GPUs** in future studies to speed up training, improve model performance, and decrease training times. Compare GPU-based training to CPU-based training for performance and cost-efficiency.

Implementation of **true hierarchical structure** in which data is generated by external edge devices (e.g., cameras) and sent to local cloud servers, enhancing the realism and applicability of the FL model.

Conclusion

Federated learning provides a **scalable** and **efficient solution**, enabling the efficient processing of data and global model updates by utilizing cloud-hosted local training servers.

Our research shows that federated learning models achieve performance levels comparable to centralized training while significantly reducing training time.

Despite challenges such as communication overhead and network delays, the system exhibits promising scalability, as evidenced by empirical results.

With further research and optimization, this federated learning framework could extend to complex tasks and diverse application

Thank you for the attention!

Renato Esposito
renato.esposito001@studenti.uniparthenope.it

Vincenzo Mele
vincenzo.mele001@studenti.uniparthenope.it

Alfredo Mungari
alfredo.mungari001@studenti.uniparthenope.it

Massimiliano Giordano Orsini
massimiliano.giordanoorsini001@studenti.uniparthenope.it

Martina Roscica
martina.roscica001@studenti.uniparthenope.it

Stefano Verrilli
stefano.verrilli001@studenti.uniparthenope.it