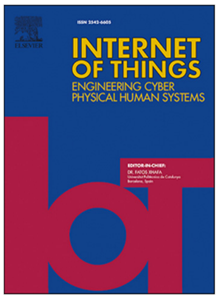
Journal Pre-proof



Attribute recognition for person re-identification using federated learning at all-in-edge

Shini Girija, Thar Baker, Naveed Ahmed, Ahmed M. Khedr, Zaher Al Aghbari, Ashish Jha, Konstantin Sobolev, Salman Ahmadi Asl, Anh-Huy Phan

PII: S2542-6605(23)00116-6
DOI: https://doi.org/10.1016/j.iot.2023.100793
Reference: IOT 100793

To appear in: *Internet of Things*

Received date : 10 February 2023
Revised date : 24 March 2023
Accepted date : 14 April 2023

Please cite this article as: S. Girija, T. Baker, N. Ahmed et al., Attribute recognition for person re-identification using federated learning at all-in-edge, *Internet of Things* (2023), doi: https://doi.org/10.1016/j.iot.2023.100793.

Highlights

Attribute Recognition for Person Re-identification using Federated Learning at All-in-Edge

Shini Girija, Thar Baker, Naveed Ahmed, Ahmed M. Khedr, Zaher Al Aghbari, Ashish Jha, Konstantin Sobolev, Salman Ahmadi Asl, Anh-Huy Phan

- Designing a novel deep learning-based all-in-edge architecture for attribute-based Person Re-identification
- Developing an innovative federated learning framework for Person Re-identification
- Proposing a unique federated aggregation strategy -FedTransferLoss that incorporates transfer learning to overcome data scarcity or erroneous data

Attribute Recognition for Person Re-identification using Federated Learning at All-in-Edge

Shini Girija[a], Thar Baker[b], Naveed Ahmed[a], Ahmed M. Khedr[a], Zaher Al Aghbari[a], Ashish Jha[c], Konstantin Sobolev[c], Salman Ahmadi Asl[c] and Anh-Huy Phan[c]

[a]Department of Computer Science., University of Sharjah, Sharjah, 27272, UAE
[b]School of Architecture, Technology and Engineering, University of Brighton, Brighton BN2 4GJ, UK
[c]1CDISE, Skolkovo Institute of Science and Technology (SKOLTECH), 121205 Moscow, Russia

ARTICLE INFO

Keywords:
Person Re-identification
Edge computing
Federated learning
Transfer learning
Attribute recognition

ABSTRACT

The advancement in person re-identification using attribute recognition is constrained by the increasingly strict data privacy standards since it necessitates the centralization of vast amounts of data containing sensitive personal data in the cloud. Cloud-based person re-identification requires the transfer of original video information to the servers, causing increased communication costs because of the need for significant bandwidth, resulting in unpredictable timing. This work presents an all-in-edge architecture for attribute-based person re-identification, which deploys training data in edge nodes that support distributed inference. Edge nodes independently learn but collaborate with specific neighboring nodes by sharing information to minimize communication and computational costs through the utilization of federated learning and transfer learning methods. Furthermore, this paper proposes a federated aggregation strategy-FedTransferLoss to obtain optimal global accuracy by using transfer learning to re-train the low-quality local models. Extensive experiments on two prominent pedestrian datasets- PETA and RAP show that FedTransferLoss achieves higher accuracy, recall and precision values compared to the traditional FedAvg algorithm.

1. Introduction

Person re-identification (Re-Id) identifies the same individual who appears in multiple cameras. Applications for this technology include company operations, crowd behavior analysis, intelligent monitoring, missing person or criminal tracking, and public safety [1]. With the use of contemporary computer-vision algorithms, attributes may be quantified quite accurately, which opens up the possibility of obtaining information about common image aspects from entirely different sources. A sequence of attributes in an image (such as the dress type, age, gender, and hair color) can be recognized using attribute recognition [2]. To differentiate pedestrian photos with identical versus distinct identities, one must employ discriminative features that can be learned to identify the images. For this, deep learning (DL) techniques have emerged as a viable option, and they are easily adaptable to Re-Id in systems of extensive monitoring [3]. However, the task of learning discriminative features for pedestrian identification becomes challenging in real-world scenarios where individuals may be captured by multiple cameras in different locations, due to factors such as variations in view, pose, illumination, surveillance angle, and the influence of weather, lighting conditions, pose, and resolution differences across cameras. Due to the computational complexity of most DL models, generating output of inference on resource-constrained devices is challenging. As a result, DL services are typically implemented in cloud data centers to handle requests. The transmission of raw video data to the servers is necessary for cloud-based Re-Id, which raises communication costs due to the high bandwidth demands and introduces time uncertainty [4]. Additionally, the centralized structure of these DL techniques presents privacy issues because pedestrian data from many end devices are shared with a central cloud server in order to train a global model [5].

The development of edge intelligence or intelligent edges is facilitated by the gradual integration of edge computing and artificial intelligence (AI). Edge intelligence is expected to shift as many DL computations as possible from the cloud to the edge, enabling the deployment of various low-latency, dependable, and decentralized intelligent

shinivava@gmail.com (S. Girija); T.Shamsa@brighton.ac.uk (.T. Baker); nahmed@sharjah.ac.ae (.N. Ahmed); akhedr@sharjah.ac.ae (.A.M. Khedr); zaher@sharjah.ac.ae (.Z. Al Aghbari); ashish.jha@skoltech.ru (.A. Jha); (.A. Jha); konstantin.sobolev@skol.tech (.K. Sobolev); s.asl@skoltech.ru (.S.A. Asl); a.phan@skoltech.ru (.A. Phan)
ORCID(s): 0000-0002-5166-4873 (.T. Baker)

applications [6]. Since edge servers are abundant and located close to the source of data generation, edge intelligence is more concentrated on decentralized and distributed architectures. As a result, even though a single central server may not have enough computational capacity to support DL training and inference, a network of edge servers can be used to efficiently train and infer from DL approaches at the edge. Some well-known examples of these DL methods are split learning [7], Federated learning (FL) [5], and distributed machine learning-based technologies [8]. To optimize the performance of a Deep Neural Network (DNN) during both training and inference, edge intelligence should be employed as it optimally utilizes the data and resources across end devices, edge nodes, and cloud data centers [9]. Figure 1 depicts six levels of rating of edge intelligence based on the volume and duration of the data offloading [9]. *Cloud intelligence* involves complete cloud-based training and inference of the DNN model. The most advanced existing Re-Id systems [10] [11] [12] are cloud-based and operate offline over stored videos, delaying the answer to the inquiry. However, the enormous bandwidth leads to communication costs and introduces time uncertainty.

Level 1: *Cloud-Edge Co-inference and Cloud Training* in which DNN model is trained on the cloud, but the DNN model is inferred by edge-cloud cooperation. Cloud-edge co-inference based Re-Id [13] [14] [15] receives input data from edge devices for processing, then sends back the results to the edge devices for co-inference. The input pedestrian photos will be sent from far-end devices to the cloud, increasing communication costs, and wide area network fluctuations may increase delay costs.

Level 2: *In-Edge Co-inference and Cloud Training* involves training of DNN in cloud and complete inference is done in edge. In-Edge co-inference based Re-Id systems [16] [17] can distribute data to various edge servers to enable distributed inference. But a cloud-centric approach like this has significant drawbacks, including huge data transfer costs, rising energy costs, and privacy issues.

Level 3: *On-Device Inference and Cloud Training* involves training the DNN model on the cloud while inference it entirely locally on the device. On-device inference systems [18] [19] can perform a real-time end-to-end Re-Id that analyzes unprocessed security footage in a real open-world environment. However, the input pedestrian photographs will be delivered from far-end devices to the cloud, increasing network overhead.

Level 4: *Cloud-Edge Co-training and Inference* involves the DNN model to be trained and inferred through edge-cloud cooperation. Cloud-Edge co-training and inference systems [20] [21] involve distributed cloud-edge learning methods for detecting salient objects. The majority of them explicitly integrate DL methods into edge networks, making the quality and efficacy of their generalization highly reliant on balanced and adequate training data.

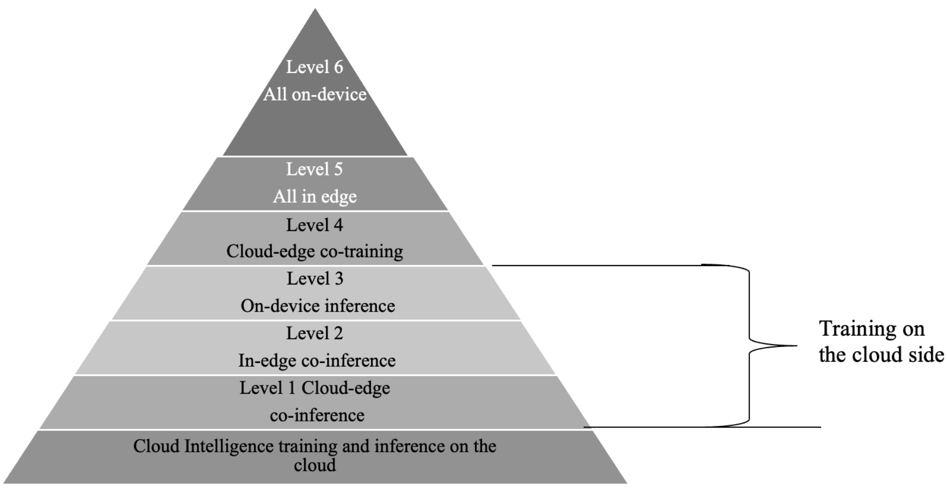


Figure 1: Six levels of ratings for Edge Intelligence[9]

In this work, we present a DL-based all-in-edge architecture for attribute-based person re-identification. The DL model should be trained and inferred upon in the edge. Tasks are moved closer to the edge, which reduce bandwidth and cloud resource costs while increasing data privacy and reducing transmission delay of offloading data. The proposed work utilizes collaborative learning to reduce the communication and computational cost on edge nodes. The edge nodes learn on their own, but they can collaborate with their neighboring nodes by sharing knowledge as needed. When a model on the target edge encounters difficulties, such as a sharp rise in error rates, knowledge transfer takes place. This study uses two collaborative learning approaches: FL and transfer learning. With FL, edge nodes can aggregate local models using FedAvg, a common aggregation technique, to jointly train a shared global model [5]. Due to insufficient or poisoned data, it offers low-quality models or misclassification in edge nodes. In order to address the issues with the standard FedAvg strategy caused by insufficient data, FedTransferLoss is incorporated in transfer learning. This solution assures improved accuracy and prevents misclassification. Six simulated experiments are used to analyze a variety of aspects related to carrying out Re-Id in the edge environment.

The primary contributions of this study are as follows:

• An innovative DL-based all-in-edge architecture is proposed for attribute-based Re-Id. This all-in-edge architecture makes use of advanced edge computing, FL and transfer learning to decrease processing and transmission latencies [22], maximize accuracy, and achieve the fastest response times [22] possible in attribute-based Re-Id applications compared to traditional centralized machine learning models. This is accomplished by using a decentralized strategy, in which the processing is split across the edge devices, hence minimizing the requirement for data transmission between the edge devices and the central server [23]. As a result, the transmission latency is decreased, and the system's response time is enhanced. Additionally, the proposed system makes use of edge computing strategies, which entail processing data and performing computations closer to the source, hence minimizing the quantity of data that must be transported to the central server [24]. As a result, there is less network traffic and communication overhead, which causes transmission latency to be lower and response time to be quicker.

• A novel FL framework for efficient collaboration among edge nodes is developed that enables the distributed edge clients to constantly learn through collaboration. With fewer communication rounds, it guarantees a communication efficient FL framework in terms of accuracy and robustness to new data. In FL, communication rounds refer to the number of times that the local model updates are sent to the central server for aggregation [25]. Hence, fewer communication rounds increase convergence speed, which will then enable the model to make a more accurate prediction on new, unknown data. Moreover, fewer communication rounds result in models that require fewer computing resources to train or use, which lowers computational costs. The model becomes more resistant to changes and noise in the incoming data by decreasing its reliance on the training data and allowing it to grasp the underlying patterns and relationships [26].

• To overcome the shortcomings of the conventional FedAvg approach caused by inadequate data in some edge nodes, an innovative federated aggregation algorithm FedTransferLoss that incorporates transfer learning is designed. It guarantees greater accuracy when updating global models based on their performance in detection and learning.

• Extensive experiments have been conducted to demonstrate that the proposed FedTransferLoss system provides more accurate results for Re-Id applications with the least amount of training time when compared to centralized learning (where just one edge node is available) and the FedAvg aggregation method.

The rest of the paper is structured as follows : The related works are introduced in Section 2, the system model is presented in Section 3, the proposed federated aggregation algorithm is presented in Section 4, performance evaluation in Section 5, results and discussion is presented in Section 6, and conclusion and future work in Section 7.

2. Related Works

In this section, the authors explore existing works on edge-based person re-identification, FL, and transfer learning by presenting contributions, features, and their limitations.

2.1. Edge-based Person Re-identification

Person Re-identification attempts to locate a person from non-overlapped surveillance footage. The performance of Re-Id in many vision-based applications has been greatly enhanced by the advancements of DL techniques such as CNN and DNN [1] [27] [18] as well as the large-scale Re-Id datasets like PETA and RAP [28]. Since DL models are intricate and resource-intensive, making it challenging to compute the inference results on user devices that have restricted resources. Because of the abundance of GPU resources in the cloud, these DL services are typically implemented there. Large-scale data processing is made possible at cheaper costs by the cloud server's vast storage and strong computation [2]. The majority of the recently evaluated papers [29] [30] [4] [2] are cloud-based, doing offline Re-Id over saved photos and videos. However, due to the significant bandwidth needs, cloud-based DNN model training and inference increases communication costs and introduces time uncertainty. The heavy traffic and slow data transfer speed in the cloud make it difficult to fulfill the immediate response needs of real-time Re-Id applications.

Edge AI is becoming increasingly popular and holds great potential for hosting computing tasks as close to data sources and users as possible [4]. Edge AI is anticipated to move as many DL calculations as feasible from the cloud to the edge, enabling a range of distributed, low bandwidth, and dependable intelligent applications. By performing as many DL computations at the edge as possible, edge intelligence aims to reduce the load on the cloud and provide reliable, low-bandwidth intelligent applications. These characteristics of edge make it the best platform for applications involving Re-Id. Transmission delay can be greatly reduced by offloading the inference tasks of Re-Id and attribute recognition to their adjacent computing edge nodes. Recent research has concentrated on edge-based Re-Id for realistic situations [31] [4] [20]. Xu et al. [16] intoduced a distinctive inference framework consisting of multiple distributed components that address both attribute recognition and person re-identification. They developed a learning algorithm that considers the distributions of these components in a dynamic mobile edge cloud-enabled camera network with uncertainties. However, the cloud's existence brings with it hefty computing expenditures. Jeong et al. [32] developed a strategy to split DNN into multiple parts delivered them to the edge servers. Following these heuristics, Teerapittayanon et al. [33] deployed the multiple DNN partitions to edge nodes and nearby end devices to reduce latency in transmission and achieve high accuracy. Though, the design, the deployment of the fragmented parts among different servers, and the network instabilities are not considered. Lightweight DL models were used at the edge to create a real-time end-to-end video-based re-identification [31]. To determine the similarity between person-image pairs, a deep squared similarity learning-based method was suggested in [30]. Their research suggested an online learning system for an environment using edge computing.

However, none of these studies address the problems with edge node collaboration, such as the privacy of sensitive data, given that training data for Re-Id contain sensitive personal data that could reveal people's identities and whereabouts. This paper introduces a novel attribute-based person re-identification architecture based on an all-in-edge DL architecture. This all-in-edge architecture uses FL, transfer learning, and sophisticated edge computing to reduce processing and transmission latencies, increase accuracy, protect privacy, and provide the quickest reaction times for attribute-based Re-Id applications.

2.2. Federated Learning

Federated Learning (FL) is a new distributed training method [34] that protects data privacy by allowing multiple clients to train models simultaneously, without sharing their data. As a result, numerous applications, including those for consumer products [35] [36] and healthcare apps [37] [38] [39], are implementing FL-based privacy-preserving measures. These distributed clients only send training updates to a central server; they do not send raw data. As the raw data is stored locally, the possibility of privacy leaking is decreased. Despite the benefits of FL, optimizing its performance and applying it to Re-Id are mostly ignored. In [17], the authors discussed this implementation possibility; however, neither a dataset nor evaluation results are provided in that study.

The majority of FL investigations [40] [41] [42] are supervised learning-based. Unsupervised FL has been the subject of several recent investigations [43] [44]. However, because they primarily concentrate on learning generic representations, these techniques are not suitable for individual re-identification.

The most popular FL algorithm for optimization is Federated Averaging (FedAvg) due to its straightforward implementation, statelessness, privacy preservation, and secure aggregation [45]. Clients transmit their locally learnt models to a server, which aggregates them, and then delivers the final global model back to the clients in an iterative training process. However, this aggregation process may lead to low accuracy when some edge nodes send low quality training models due to limited data.

This study presents a novel Re-Id system that utilizes FL, allowing multiple edge clients to collaborate and continually learn. FedTransferLoss is a novel federated aggregation technique that combines transfer learning, and it is proposed as a solution to the problems with the usual FedAvg approach brought on by insufficient data in some edge nodes. It ensures higher accuracy when updating global models based on how well they perform in detection and learning.

2.3. Transfer Learning

Transfer learning is a technique used by researchers in the fields of DL, pattern recognition, Re-Id, and computer vision to resolve issues with incomplete or incorrectly labeled data [46]. Most Re-Id applications require significant labeled datasets, and gathering such data is challenging [47]. The largest person Re-Id dataset is simply big enough. Transfer learning, which involves training the model on a bigger dataset and testing it on the Re-Id datasets, is used to address this drawback of the limited training set. In order to fill the gap left by the lack of a substantial body of data for Re-Id, the model is trained on a substantially bigger dataset, after which it is adjusted and tested. ImageNet [48], one of the notably huge datasets, is widely utilized and effective in transfer learning. Ahmed et al. [49] applied transfer learning to address a crucial issue in Re-Id that has not gotten much attention up to this point: how to swiftly integrate new cameras into an established camera network. By implementing transfer learning, they developed an effective model adaptation approach that leverages information from source models and a limited amount of labeled data, without relying on any source camera data from the current network. In order to overcome the challenge of collecting strong descriptors from low-contrast images with conventional manual features, Cheng et al. [50] adopted a CNN-based knowledge transfer architecture to extract powerful discriminative features from low-quality photos. However, person-identification applications have not explored the use of transfer learning in FL much. In circumstances with constrained data availability, transfer learning can be used to improve the accuracy of global models.

The contributions, features, and gaps of the examined state-of-the-art research are summarized in Table 1. The table makes it abundantly evident that very few publications concentrate on the key difficulties that arise during the implementation of person re-identification in edge platforms, such as privacy, poor local models, and the manner in which edge nodes collaborate. The remaining study focuses on edge node heterogeneity of data and cloud-based recognition. The examined literature generally takes into account compute costs, real-time operations, and data heterogeneity, however, this research aims for a high quality global model with the best accuracy for Re-Id in edge platforms. Hence transfer learning is leveraged in the proposed algorithm to handle inadequate data situations and increase the prediction accuracy of the trained global model.

Table 1: Summary of the common reviewed state-of-the-art research

Authors	Contribution	Features	Research gap
Lin et al. [2]	Attribute-person recognition system for pedestrian identification taking into account the relationships and correlations between a person's attribute	cloud-based	not suited for real time applications
Zhong et al. [29]	Utilized distributed cloud computing to complete the process of Re-Id	cloud- based	Recognition accuracy is less
Zheng et al. [30]	pedestrian alignment network that concurrently learns pedestrian descriptions and aligns pedestrians within images	cloud-based	excessive transmission latency in the cloud.
Xu et al. [16]	models for pedestrian attribute recognition with re-identification in a mobile edge cloud-enabled camera surveillance system	Mobile edge cloud-enabled	hefty computing expenditures not suitable in real time.
Teerapittayanon et al. [33]	split training across the edge nodes for Re-Id	all-in-edge	privacy problems and uncertainty in edge nodes collaboration.

Bipin et al. [18]	complete multi-person, multi-camera tracking surveillance system that was put into use on an edge in real time	all-in-edge	Uncertainty in edge nodes collaboration
Zhuang et al. [40]	proposed FedPav strategy to research the statistical heterogeneity issue when using FL to Re-Id	all-in-edge	did not consider system heterogeneity in real time.
Zhuang et al. [41]	introduced FedUReID, a federated unsupervised Re-Id method that learns models without labels while maintaining privacy.	Edge-cloud based	did not consider system heterogeneity in real time.

3. System Model

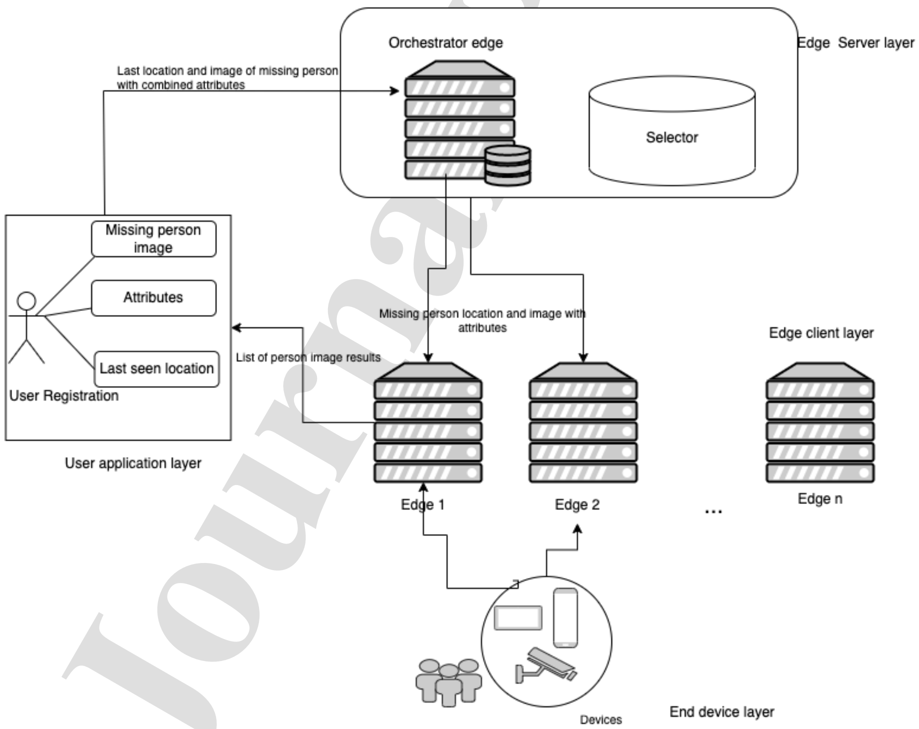


Figure 2: Proposed architecture of edge-based Re-Id system

This section outlines the design of the proposed edge-based Re-Id system, depicted in Figure 2. We incorporated the distributed architecture [3] into our application for DL training at the edge server. One of the edge servers performs the role of an orchestrator in a distributed architecture. The other collaborating edge servers help in training the DNN model. The edge clients receive the DNN model from the orchestrator, and they use the local data to train it. After the DNN model has finished local training, the edge clients communicate the finished model back to the orchestrator. The orchestrator then integrates every DNN model that comes in to create a global model. The orchestrator then distributes the generated global DNN model to the remaining edge clients for additional training cycles. In the suggested architecture, collaborative learning based on training data contributions from numerous edge devices is utilized. Edge nodes acquire information on their own but collaborate with logical neighbors only when necessary. Edge node communication and processing overhead are decreased through collaborative learning. There are many collaborative learning methods that may be utilized for training, including split learning [7], gradient compression [51], and FL [41] and [40]. The two collaborative learning ideas used in the proposed study are FL and transfer learning.

3.1. Proposed architecture of edge-based person re-identification system

Figure 2 illustrates the utilization of a four-layer, edge-based architecture that facilitates the analysis of distribution of data and collaboration in model training through FL.

Edge server layer: This layer consists of an orchestrator edge server and a selector. One of the edge servers in a distributed architecture serves as the orchestrator. The DNN model is trained with assistance from the other participating edge servers. The orchestration node manages resources and decides whether to carry out tasks using the node's resources or those of a neighboring node. The selector modules determine the selection of edge clients based on GPS coordinates of edge nodes and the missing person's location.

Edge client layer: This layer is made up of computing nodes acting as edge clients like base stations. Training and inference of the pedestrian data is conducted in this layer and results of missing person is sent to user application.

User application layer: The application in which missing person's query can be registered by providing images, last seen location and pedestrian attributes.

End device layer: The end device layer is made up of different kinds of end devices, including GPS, accelerometers, cameras, mobile phones, CCTV, and sensors.

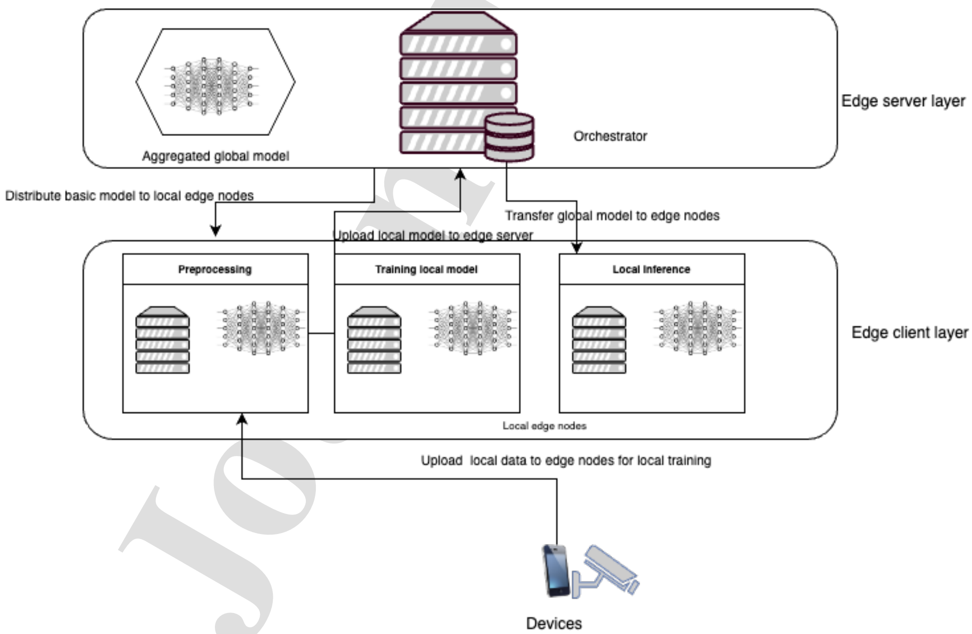


Figure 3: Proposed federated learning framework

3.2. Proposed federated learning framework for collaboration among edge nodes

Edge devices may be enabled by FL [34] [38] to train DL models using their own gathered data, and then only upload the refined model. Figure 3 depicts the FL process in the proposed architecture. The following are the primary stages in the FL process:-

(1) Local edges are coordinated by the orchestrator edge server so that models can be trained using locally captured images.

(2) Download the global DL model from an orchestrator edge node and deliver it to every edge node nearby the missing person's last known position.

(3) Using the downloaded global model and their own locally acquired images, train their local models.

(4) For model averaging, only upload the updated local model to the orchestrator.

This architecture significantly reduces the risk of privacy breaches because the data remains at the edges. Local training datasets are created by combining photos taken from multiple camera perspectives, similar to how benchmark datasets are collected. Meanwhile, edge servers collaborate with their local datasets to perform FL under the oversight of an orchestrator edge.

FL offers a crucial method for enabling the DL model training and inference at the all-in-edge architecture. The development of the DL model at the edge can be aided by the integration of numerous low resource edge servers using FL. Based on the resources accessible at the edge and the design's communication cost, a distributed or decentralized approach can be adopted.

3.3. Communication protocols of edge based person re-identification system

The proposed system makes use of a collection of protocols and algorithms created to streamline communication between the system's various components.

(1) User sends missing person request to orchestrator edge node by providing image, pedestrian attributes and last seen location.

(2) We assume that orchestrator edge node has knowledge of the location of edge nodes registered in the application. Orchestrator forwards the request to the nearest edge nodes along with received person details and global information regarding optimal edge node locations.

(3)Edge node broadcasts the request to selected end devices (CCTV camera, mobile phones etc.). Selection of end devices is based on missing person last seen location and global route status from orchestrator node.

(4) End devices send person images matching with attributes and current location to edge node.

(5)Edge generates a set of optimal matching person images from received images using an inference module. The requests are fed to inference module and an optimal matching set of images are generated with accuracy acc. If the obtained accuracy acc is less than threshold accuracy, the request is sent to logical edge neighbors and repeat the steps 2 to 4 again.

(6) The user examines the resultant images and informs orchestrator edge node that the missing person is tracked.

4. The proposed federated aggregation algorithm

This section illustrates the limitations of centralized training, and traditional federated aggregation algorithm and presents the proposed algorithm -FedTransferLoss.

4.1. Limitations of Centralized training

Traditionally, Machine Learning (ML) has been performed by training a general model, which is then sent to all connected devices by uploading all data from each device to a central server. Therefore, this training has several limitations. First of all, users are growing more concerned about the privacy of the data that is made available for centralized learning. These records could be extremely private such as personal sensitive information, medical records, and payment card details. It is quite possible that users' privacy will be violated by eavesdropping attacks when this data is shared with the central server [52]. Furthermore, bandwidth is frequently extremely constrained, and the sheer volume of data may be too much to send to the central server. Finally, latency caused by the round trip to the central server is frequently a problem for real-time applications such as Re-Id [53].

4.2. Limitation of FedAvg algorithm

The edge devices in FL train a DNN locally on their data. As a result, the edge clients communicate the parameters or gradient changes of their particular client model with an orchestrator edge server rather than sharing the raw data. After collecting the parameters from the participating clients, the orchestrator shares the averaged global model with the clients. The desired number of client and server communication rounds are completed by repeating this cycle. The most common algorithm used for the aggregation of local models to a global model is Federated Averaging (FedAvg) algorithm [45] [54] [36] [55]. The FedAvg algorithm is explained in detail in the following.

The goal of FedAvg is to minimize the global model's objective, w, which is just the weighted average of the local device loss added together for each round.

$$min_w f(w) = \sum_{k=1}^{N} p_k F_k(w) \tag{1}$$

where $F_k(w)$ represents loss on edge k

1. A random sample is taken from a subset of clients.
2. Each client receives a broadcast from the server's global model.
3. The client performs Stochastic Gradient Descent (SGD) on their own loss function in parallel and sends the finished model to the server for aggregation.
4. After that, the server changes its global model based on the average of these local models.
5. The procedure is carried out for n communication rounds.

As a result, the FedAvg aggregation approach aggregates the weight averages across all neural network models. However, low accuracy could result if some nodes send the server low-quality training models due of insufficient data, damaged models, or even malicious settings[17]. By using the proposed model, this limitation can be rectified which offers a high accuracy compared to FedAvg.

Algorithm 1 FedTransferLoss learning process- Server side execution

Input: Total data size n, K edges are indexed by k, client k's data volume n_k, E is number of local epochs, B is the local minibatch size, θ is the threshold loss value, T is number of rounds, C is number of clients that perform computations in each round and η is the learning rate.

Output: Aggregated global model w_T, optimal client k's local model w_T^k.

```
Server executes:
    initialize $w_0$;
    for each round $t$=0 to $T-1$ do
        $m \leftarrow max(C.K, 1)$;
        $S_t \leftarrow$ (random set of $k$ edges selected from $K$ edges);
    for each client $k \epsilon S_t$ in parallel do
            $w_{t+1}^k \leftarrow ClientUpdation(k, w_t)$
        If $w_{t+1}^k \geq \theta then$
            $Server\ retrain\ client\ k\ with\ local\ model$ $(w_t^k \leq \theta)$
            repeat ClientUpdation ($k$,$w$)
        $w_T \leftarrow \sum_{n=1}^{k} \frac{n_k}{n} w_{t+1}^k$
return aggregated global model $w_T$, optimal client $k$'s local model $w_T^k$
```

4.3. FedTransferLoss

This work incorporates the transfer learning (TL) concept in FedAvg algorithm to overcome the limitations of low-quality training models. TL is a machine learning technique that transfers knowledge from a source domain (or model) into a target domain. The ideas of TL are built on the idea that a model's attributes can be applied to other jobs that are similar. When there is insufficient data to properly train models, especially in the case of FL where data

Algorithm 2 Client Updation

Input: Total data size n, K edges are indexed by k, client k's data volume n_k, E is a number of local epochs, B is the local minibatch size, θ is the threshold loss value, P is a number of client k's data points and η is the learning rate.

Output: Weight parameters of local client model of each edge w_T^k.

```
ClientUpdation (k,w):
    B← (divide P_k into batches of size B)
    for each local epoch i from 1 to E do
        for batch b∈B do
            w ← w − η∇(w; b)
return w to Server
```

access is constrained, TL becomes essential. The proposed algorithm is **FedTransferLoss** in which if the minimum loss function of a local edge node is greater than a threshold value, we can apply an optimal available local model to train that edge node again to achieve better accuracy. When an edge device implementing a model encounters an issue, such as an unanticipated increase in error rates, knowledge transfer takes place.

The term "minimal loss function threshold" refers to a predetermined threshold for the loss value that causes some action or decision to be made, such as stopping the training process, changing the learning rate, or retraining the model with a different set of hyperparameters. The pre-determined threshold value at the beginning of each epoch t is computed [56]as following

$$\text{Threshold loss value}, \theta_t = \mu + k * \sigma \tag{2}$$

Where μ is the mean of the loss values over all participating customers, and is the standard deviation of the loss values. The threshold value's distance from the mean is determined by the scaling parameter k. A higher k number indicates a more cautious threshold value, whereas a lower k value indicates an aggressive threshold value. This method for obtaining a predetermined threshold value is based on the notion that a threshold value can be generated by exploiting statistical features of the loss function to regulate the convergence of the global model in FL. The typical statistical features of mean and standard deviation can give a good approximation of the model's present performance as well as the distribution of loss values among the participating clients. Using the aforementioned formula, a threshold value of 0.302 based on the average loss values among the clients is calculated for this FL system with 8 participating clients and k=2.

In FedTransferLoss, the server may decide to retrain the local client edge node with an optimal local model received from another node in order to increase the accuracy of the global model if a local edge node's weighted average loss, which is the average of the losses over all samples weighted by their respective importance, exceeds a predetermined loss threshold value. The model that produces the smallest loss on the local dataset is referred to as the optimal local model. If the loss for all local models is larger than the threshold, the threshold value may need to be recomputed to enhance the model's performance during training. A moving average of the loss values can be used for recalculating the threshold value [57]. The loss function can be smoothed out with the moving average, which also offers a more reliable estimate of the model's present performance. The moving average is calculated using the following formula:

$$\text{New threshold value} = \alpha * \text{loss value} + (1-\alpha) * \text{Previous threshold value} \tag{3}$$

The smoothing factor α regulates how much weight is given to the most recent loss value in relation to the previous threshold value. A smaller number gives the prior threshold value more weight, whereas a greater value gives the most recent loss value more weight. Using the aforementioned formula, a new threshold value based on the moving average of the loss values across the clients is computed as 0.282. This new threshold value is based on the existing loss value (0.2) and a smoothing factor of 0.2.

The detailed FedTransferLoss learning process in which server side execution is given in Algorithm 1 and client updation is given in Algorithm 2. Table 2 presents the list of notations used in this paper and its definition.

Table 2: Notations and Definitions

Notations	Definitions
n	Data size
K	Number of edges
n_k	Data volume of client edge k
E	Number of local epochs
B	Local minibatch size
θ	Threshold loss
T	Number of rounds
C	Number of clients that perform computations in each round
η	Learning rate
w	Weighted average of the local device loss added together for each round
w_T	Aggregated global model
w_T^k	Local client model
P_k	Number of client k's data points
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
w_t	local model's gradient weight on round t
$f(w)$	Loss with model parameter w
g_k	Average gradient on the global model

All client gradient weights are initially set to w_0 in FedTransferLoss. A random subset of clients participating in the training is selected. Each selected client will locally train the model using its own local data and then modify the model, indicated by $w \leftarrow w - \eta\nabla(w; b)$. w_t provides the local model's gradients on round t. These local models are transmitted to the server for global aggregation. Calculating the gradient descent of the loss function yields a value for the minimum loss function threshold. If a local edge node's weighted average loss exceeds a certain threshold, the server retrains the local client edge node with an optimal local model received. The server combines and averages the local models to form the global model, which is then distributed to all clients for further training on a global scale.

5. Performance evaluation

Initially, this section provides a thorough introduction to the datasets, evaluation criteria, and implementation specifics used in the simulation experiments. Then, experimental results on these datasets are provided to facilitate a fair evaluation of our proposed algorithm and novel techniques. Finally, several ablation experiments are conducted on the proposed approach to demonstrate the effect of various factors such as dataset proportion, quantity of edge nodes, and training duration. Table 3 presents the simulation parameters used in the experiments.

Table 3: Simulation Parameters

Parameters	Value
Epochs	Centralized-20, Federated-20
Server rounds	Centralized-0, Federated-40
Learning rate	0.001
Batch size	128

Sequence length	20
Training data	Centralized -80 percent of the selected dataset, Federated: 80 percent of dataset split among n number of clients.
Testing data	Centralized-20 percent of a selected dataset, Federated: 20 percent of dataset split among *n* number of clients.
Number of clients in FL	8

5.1. Dataset Description

Pedestrian datasets used in this study are PETA and RAP. The PEdesTrian Attribute dataset (PETA) [58] is a dataset for identifying pedestrian traits, like gender and dress style, at a distance. It is relevant in situations involving video surveillance where it is difficult to get close-ups of the body and face. 19,000 pedestrian photos with 65 attributes make up the collection (61 binary and 4 multi-class). 8705 people may be seen in those pictures [58].

Another dataset for identifying pedestrian attributes is the Richly Annotated Pedestrian (RAP) dataset [59]. It includes 41,585 images gathered from indoor security cameras. Each image has 72 annotations, however, only 51 binary annotations with a positive ratio greater than 1 percent are chosen for examination. There are 8,317 images for testing and 33,268 for the training set. The RAP dataset is the largest pedestrian attribute dataset and is anticipated to significantly advance research into large-scale attribute recognition systems [59].

5.2. Implementation

The Pytorch [60] and Flower [61] platforms are used to implement all settings. The models from [62] are deployed in the centralized experiments, where all experiments are run for 20 epochs. We adhere to FL benchmarks in federated categorization [63]. One server and eight clients are used for our experimentation. The range of clients that need transfer learning goes from 1 to 8. We choose at random 80 % of the datasets for PETA and RAP, divided among n clients, to serve as the training set, and 20 % of the datasets, divided among n clients, to serve as the testing set at the client level. The Flower Python framework was used to create a FL command line interface platform. The platform consists of a host server and multiple clients, and the clients will communicate with the server during the training process. The server sends model parameters to the clients. The clients manage the parameters and conduct the training. The server receives the modified parameters and sends them back after averaging all of the updates. This explains one phase of the FL process; subsequent rounds are described in the same manner. The FedAvg algorithm selects a random subset of the total number of clients participating in the training. The selected clients train the data locally and send back local models to the server for aggregation. A global model for testing is created by the aggregate function FedAvg after each round of local training in Flower.

FedTransferLoss uses the PyTorch communication backend for model aggregation and model updating. In every round for every experiment, we assess both the local and global models. Then, we present the results of the best round on each dataset. To maintain a constant dimension for the model, the pre-processing stage is first completed by resizing each image in the dataset to a single size of (224,224). The images are then transformed into tensors, which PyTorch processes before normalizing each image with a mean and standard deviation of 0.5. Then, establish the locations for the train, test, and validation sets that will be used as inputs for the "datasets" module. As a result, the data may be put onto the GPU because the PyTorch model knows exactly where it is. The batch size is kept at 128. The best pre-trained local model that is currently available is chosen, and training of the model's parameters is disabled because the pre-trained parameters are used to extract features from the dataset. The CNN classifier is then used to replace the model's top fully connected layers. Different attributes, such as gender, age, clothes colour, accessories, etc., are identified using the 3x3 CNN structure with a total of 12 layers. Five convolutional layers and three pooling layers are used to process the input image. One global average pooling layer and three fully connected layers are then used to produce a set of attribute predictions. A sigmoid activation function[1] in the output layer is utilised to handle binary features and generates a probability value between 0 and 1. A softmax activation function [2] in the output layer is utilised for multi-class attribute recognition and generates a probability distribution across all potential attribute values. The

[1] https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

[2] https://machinelearningmastery.com/softmax-activation-function-with-python/

entire batch of data is normalized by batch norm into the number of neurons specified as a parameter. This lessens the model's complexity and keeps it from overfitting. If a GPU is available, this model can also be transferred to it. For the training phase, CrossEntropyLoss() is chosen as the loss function, while the Adam classifier serves as the optimizer. The learning rate for the optimizer is set at 0.001 percent.

5.3. Evaluation metrics

The most often used evaluation measures for the pedestrian attribute detection task are Accuracy, Precision, Recall, and loss function [8]. These measurements aid in determining which training algorithm produces better results and is more effective. We need more than one evaluation metric since it is impossible to assess an algorithm's effectiveness and performance using only one metric.The standard indicators, including True Positive (*TP*), True Negative (*TN*), False Positive (*FP*), and False Negative (*FN*), are used to assess these metrics. For example, *TP* and *TN* measure the number of correctly identified tests, while *FP* and *FN* reveal the number of tests that the machine learning model misclassified.

Accuracy indicates how frequently the trained model was overall correct [4]. For instance, in Re-Id applications, model accuracy refers to the number of times the model made accurate predictions to locate the missing person. Precision talks about how precise/accurate the training model is out of those predicted positives, and how many of them are true positive [64]. When the costs of false positives are large, precision is an effective measurement to use. For instance, a false positive in Re-Id applications denotes a model misclassification based on user-specified attributes. If the precision is not high for the Re-Id apps, the correct identification of the missing person may not happen. Recall determines how many actual positives the training model actually captures by classifying it as Positive (true positive) [65]. When a false negative has a significant cost, recall is the model statistic used to choose the best model. For example, in Re-Id applications, if the missing person's image (true positive) is predicted to not be missing (false Negative). If the missing individual comes into contact with dangers that could endanger their lives, the cost of a false negative will be exceedingly significant.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

This work also makes use of loss function and stochastic gradient descent (SGD), which can be crudely applied to the federated optimization problem.

For each client k, the average gradient on the global model of the current global model w_T is defined as follows.

$$F_k(w) = 1/n_k \sum_{i \epsilon P_k} f_i(w) \tag{7}$$

$$g_k = \nabla F_k(w_t) \tag{8}$$

where P_k - Set of data points on client k and $f_i(w)$- Loss with model parameters w.

The update is subsequently implemented by the central server after aggregating these gradients as defined below.

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} n_k/ng_k \tag{9}$$

where w_t - provides the local model's gradients in round t, η - learning rate, and n_k- Number of data points on client k.

6. Results and Discussion

In order to be fair, we should contrast our approach with one that is comparable to it and takes into account joint training of Re-Id and attribute recognition. As a result, we select a previous study by [13] that uses CNN for centralized training. The proposed model will be validated and its performance will be assessed on PETA and RAP datasets in comparison to FedAvg and centralized learning. The outcomes are also contrasted with the variable dataset ratio/fixed dataset ratio and the fixed dataset ratio/variable number of clients stated in Section 6.3.

6.1. Comparison of FedTransferLoss with Centralized training and FedAvg

A comparison of the centralized training, FedAvg and proposed method, FedTransferLoss for PETA is shown in Table 3. As expected, centralized training has higher performance with an accuracy 79.95 % compared to other methods. The proposed method, FedTransferLoss achieved better accuracy of 75.38 % than FedAvg (72.94 %). Despite taking longer to train than FedAvg (983s), the proposed method (1638s) offers the highest accuracy and lowest loss, hence the long training time is acceptable. Hence FedTransferLoss enhances the local model's edge performance when there is insufficient or erroneous data. By choosing the best local models for the edge server's global model aggregation, the proposed algorithm guarantees excellent detection accuracy in fewer communication rounds.

Table 4: Comparison of evaluation metrics based on PETA among different training algorithms

Training type	**Accuracy (%)**	**Recall**	**Precision**	**Loss**	**Training time (seconds)**
Centralized training	**79.95**	80.23	78.39	21.05	3240s
FedAvg	72.94	62.38	78.34	28.38	983s
FedTransferLoss	**75.38**	**76.39**	73.89	25.24	**1638s**

Table 4 compares the suggested approach, FedTransferLoss for RAP, with the centralized training, FedAvg. Similar to the PETA dataset, the results for the RAP dataset are overwhelmingly favorable. As anticipated, centralized training performs better than other approaches, with an accuracy rate of 91.12 %. The suggested method, FedTransferLoss, has a greater accuracy rate of 89.78 % than FedAvg (83.27 %). A significant contributor to the improvement in accuracy over PETA is the quantity of the RAP dataset. The proposed method (3648s) delivers the highest accuracy and lowest loss, hence the long training time is acceptable even though it takes longer to train than FedAvg (2289s).

Table 5: Comparison of evaluation metrics based on RAP among different training algorithms

Training type	**Accuracy %**	**Recall**	**Precision**	**Loss**	**Training time (seconds)**
Centralized training	**91.12**	69.31	80.77	7.02	5939s
FedAvg	83.27	64.67	81.38	16.23	2289s
FedTransferLoss	**89.78**	**68.45**	**81.23**	11.34	**3648s**

Figure 4 presents the comparison of the accuracy and loss function of centralized learning and FL platforms for RAP. As discussed earlier, after 40 rounds of learning, the accuracy of the proposed method-FedTransferLoss shows convergence rate near to the centralized learning.

Table 6: Comparison of results of proposed method with other existing works

Methods	**Type**	**Accuracy for PETA (%)**	**Accuracy for RAP (%)**
Sudowe et al. [66]	DL based	73.66	62.61

Liu et al. [67]	DL based	74.62	53.30
Raghavendra et al. [68]	DL based	74.79	80.71
Zhuang et al. [41]	Federated learning based	65.2	82.2
Zhang et al. [17]	Federated learning based	70.6	79.1
Ours	Federated learning based	**75.38**	**89.78**

Comparing our proposed FL approach to earlier DL and FL approaches, the test set accuracy was greater than existing works(see Table 6). In particular, our method attained an average accuracy of 75.38% on the PETA dataset and 89.78% on the RAP dataset, which is high compared to earlier methods. Overall, especially for the RAP dataset, the findings considerably outperformed the state-of-the-art. The magnitude of the RAP dataset is a major contributing factor to this difference.

6.2. Experiment for reducing the number of false negatives in FedTransferLoss

Experiment using re-weighting is conducted to reduce the number of false negatives by modifying the loss function used during training. The concept behind this method is to give the positive class more weight in the loss function so that the model will be penalized more severely if it predicts the negative class wrongly while the true label is positive [69]. The loss function is changed to emphasise correctly predicting positive samples, or instances where the model successfully detected a person in a variety of camera perspectives. This can be accomplished by giving positive samples in the loss function more weight while giving negative samples less weight. FedTransferLoss technique is used to train the updated model and distribute it among local edges. This procedure can be repeated with various weightings of the positive and negative samples to achieve the best tradeoff between lowering false negatives and preserving overall accuracy. This experiment ultimately enable to evaluate the efficacy of re-weighting as a method for enhancing the performance of Re-ID applications.

Table 7: Re-weighting experiment with different weights for positive and negative samples.

Weight for positive sample	**Weight for negative sample**	**False Negative rate (%)**	**Accuracy (%)**	**Loss**
0.5	0.5	30	75.38	25.24
0.6	0.4	28	77.81	23.19
0.7	0.3	26	78.34	21.66
0.8	0.2	24	**78.69**	21.31

Table 7 depicts the result of re-weighting experiment with different weights for PETA dataset. In order to reduce false negatives while preserving overall accuracy, we experimented with different weights for positive and negative samples. We discovered that the greatest performance of 78.69% was obtained with a weight of 0.8 for positive samples and 0.2 for negative ones. In particular, in situations where privacy and communication overhead are key considerations, this experiment shows the value of applying re-weighting in a FL setting for boosting the performance of Re-Id models.

6.3. Experiment on fixed dataset ratio/variable number of clients in FedTransferLoss

Here, we are examining the effects of fixed data size and variable number of edge clients on learning outcomes during the FL process. The number of clients who require transfer learning changes from 1-8 as the overall amount of data used for FL remains constant.The RAP dataset has been split into two parts, with 33,268 images designated for training and 8,317 images designated for testing. Meanwhile, the PETA dataset has been divided into 9,500 training samples, 1,900 validation samples, and 7,600 testing samples. Although the total time required for transfer learning increases as the number of clients increases, it still remains less compared to the time needed for centralized training.

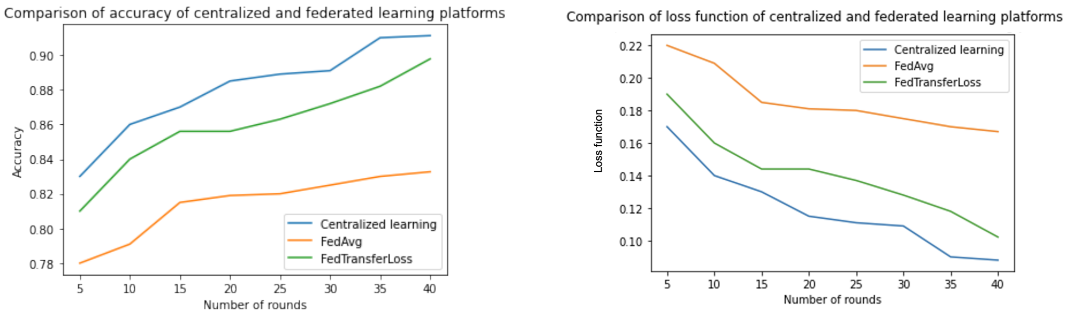


Figure 4: Comparison of accuracy and loss function of centralized and federated learning platforms for RAP dataset.

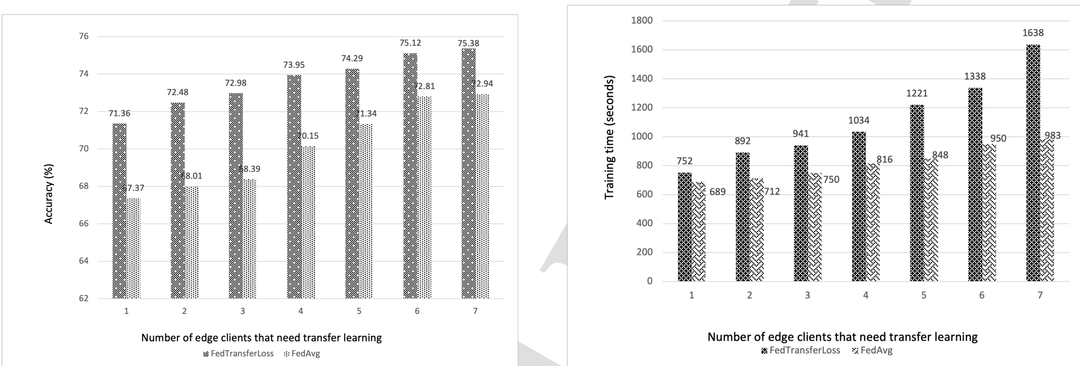


Figure 5: Comparison of accuracy and training time with variable number of edge clients in FedtransferLoss and FedAvg for PETA dataset.

The task can be distributed more evenly as the number of clients rises, but there is a cost associated with communication and coordination with the server, as well as a longer time required to aggregate the model weights, which increases training time.

Figure 5 and Figure 6 illustrate the impact of variable edge clients and fixed dataset ratio in terms of accuracy and training time in FedTransferLoss and FedAvg for PETA and RAP dataset. The graphs clearly show that as the number of edge clients increases, convergence slows down, even when the same amount of data is used for FL. This makes sense because there will be less data samples available for each client as a result of more participants splitting the overall dataset, which would lengthen training time but increases the overall accuracy of FedTransferLoss compared to FedAvg..

6.4. Experiment on Variable dataset ratio/Fixed number of clients in FedTransferLoss

The learning results during the FL process are being studied here in relation to a fixed number of clients who require transfer learning and a variable dataset ratio. In this experiment, there are a total of four edge clients and a fixed number of two nodes that require transfer learning. The RAP dataset is divided into 8, 317 test photos and 33, 268 training images. The PETA dataset is partitioned into 9,500 samples for training, 1,900 samples for validation, and 7,600 samples for testing. For a fixed number of edge clients, the learning performance in terms of testing accuracy improves as the dataset percentage utilized for FL increases. As the dataset quantity is reduced, training time gets lower.

Figure 7 and Figure 8 present the impact of variable dataset ratio and fixed number of edge clients that need transfer learning in terms of accuracy and training time in FedTransferLoss and FedAvg for PETA and RAP datasets. With the same number of transfer learning clients, it is evident from the graphs that the greater the dataset fraction utilized

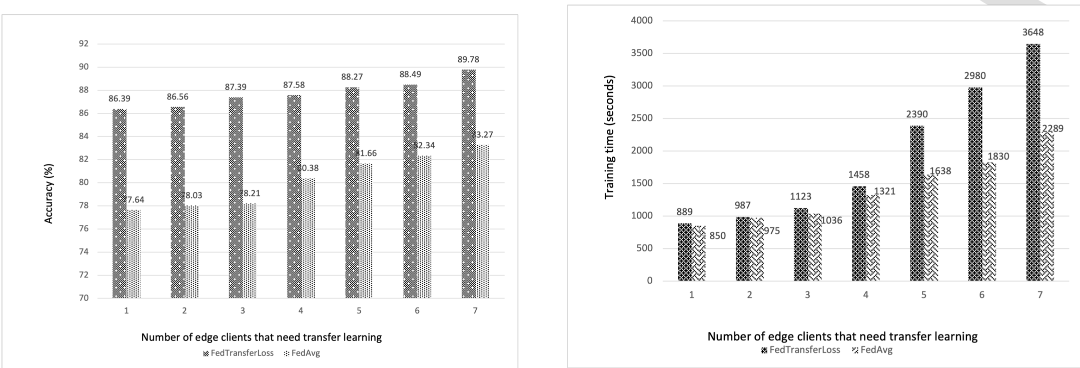


Figure 6: Comparison of accuracy and training time with variable number of edge clients in FedtransferLoss and FedAvg for RAP dataset.

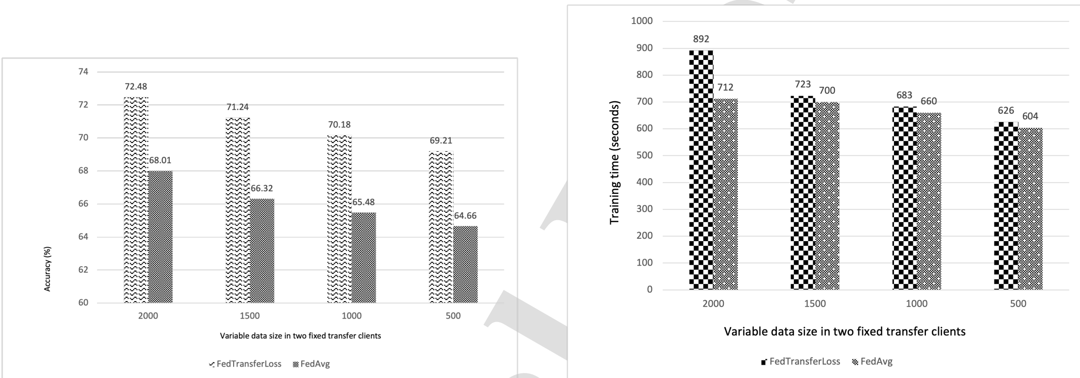


Figure 7: Comparison of accuracy and training time with variable dataset ratio in FedTransferLoss and FedAvg for PETA dataset.

for FL, the better the learning performance in terms of testing accuracy for FedTransferLoss compared to FedAvg. Accuracy will be lower if there are not enough data samples.

7. Conclusion and Future work

This study presents an all-in-edge architecture for attribute-based Re-Id that is based on deep learning. As tasks are pushed closer to the edge, bandwidth and cloud resource costs are reduced, data privacy is increased, and the transmission time of offloading data is decreased. The biggest drawback of this strategy is the extra communication required between edge nodes and endpoints. The proposed study utilizes collaborative learning, in which edge nodes learn on their own but can collaborate with nearby nodes by sharing information as necessary, to minimize communication and computational demands among edge nodes. The study employs two concepts of collaborative learning, which are FL and transfer learning.

There is significant debate over the aggregation process within the FL framework. The server will use an aggregation method to generate the system machine learning model once it has received all of the parameters from all machine learning training models. If certain nodes delivered corrupted training models, malicious parameters, or low-quality training models to the server, the accuracy could suffer. FedTransferLoss, which incorporates transfer learning in global aggregation process, is presented as a solution to this constraint. It is evident from the simulation experiments that FedTransferLoss gives greater accuracy than the standard FedAvg approach. Additionally, the accuracy and training time relationships between the ratio of datasets and the number of edge clients are also examined.

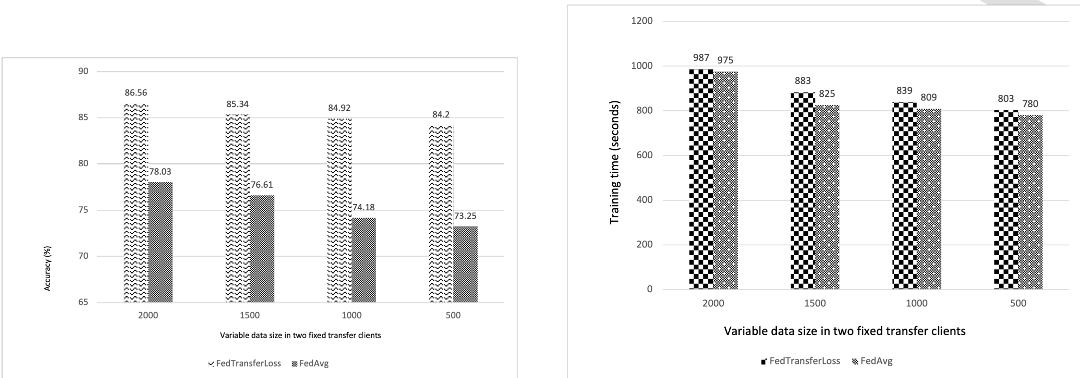


Figure 8: Comparison of accuracy and training time with variable dataset ratio in FedTransferLoss and FedAvg for RAP dataset.

To enhance the effectiveness of learning, the future work will incorporate more classifiers and deep learning models. We also intend to investigate averaging methods for FL as well as heterogeneous features and distribution space for transfer learning. In-depth research to increase the viability and acceptance of the transfer learning principle for Re-Id applications is another promising direction.

References

[1] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):2872–2893, 2021.

[2] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 95:151–161, 2019.

[3] Kejun Wang, Haolin Wang, Meichen Liu, Xianglei Xing, and Tian Han. Survey on person re-identification based on deep learning. *CAAI Transactions on Intelligence Technology*, 3(4):219–227, 2018.

[4] Xiaofei Wang, Yiwen Han, Victor CM Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(2):869–904, 2020.

[5] Castro Elizondo Jose Ezequiel, Martin Gjoreski, and Marc Langheinrich. Federated learning for privacy-aware human mobility modeling. *Frontiers in Artificial Intelligence*, 5:867046, 2022.

[6] En Li, Zhi Zhou, and Xu Chen. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, pages 31–36, 2018.

[7] Yansong Gao, Minki Kim, Sharif Abuadbba, Yeonjae Kim, Chandra Thapa, Kyuyeon Kim, Seyit A Camtepe, Hyoungshick Kim, and Surya Nepal. End-to-end evaluation of federated learning and split learning for internet of things. *arXiv preprint arXiv:2003.13376*, 2020.

[8] Muhammad Mansoor Ashraf, Muhammad Waqas, Ghulam Abbas, Thar Baker, Ziaul Haq Abbas, and Hisham Alasmary. Feddp: A privacy-protecting theft detection scheme in smart grids using federated learning. *Energies*, 15(17):6241, 2022.

[9] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8):1738–1762, 2019.

[10] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2018.

[11] Rodolfo Quispe and Helio Pedrini. Top-db-net: Top dropblock for activation enhancement in person re-identification. In *2020 25th International conference on pattern recognition (ICPR)*, pages 2980–2987. IEEE, 2021.

[12] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3415–3424, 2017.

[13] Yanming Chen, Tianbo Yang, Chao Li, and Yiwen Zhang. A binarized segmented resnet based on edge computing for re-identification. *Sensors*, 20(23):6902, 2020.

[14] Mohammadreza Baharani, Shrey Mohan, and Hamed Tabkhi. Real-time person re-identification at the edge: A mixed precision approach. In *International Conference on Image Analysis and Recognition*, pages 27–39. Springer, 2019.

[15] Svebor Karaman and Andrew D Bagdanov. Identity inference: generalizing person re-identification scenarios. In *European Conference on Computer Vision*, pages 443–452. Springer, 2012.

[16] Zichuan Xu, Jiangkai Wu, Qiufen Xia, Pan Zhou, Jiankang Ren, and Huizhi Liang. Identity-aware attribute recognition via real-time distributed inference in mobile edge clouds. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3265–3273, 2020.

[17] Chong Zhang, Xiao Liu, Jia Xu, Tianxiang Chen, Gang Li, Frank Jiang, and Xuejun Li. An edge based federated learning framework for person re-identification in uav delivery service. In *2021 IEEE International Conference on Web Services (ICWS)*, pages 500–505. IEEE, 2021.

[18] Bipin Gaikwad and Abhijit Karmakar. End-to-end person re-identification: Real-time video surveillance over edge-cloud environment. *Computers and Electrical Engineering*, 99:107824, 2022.

[19] Tianshu Hao, Yunyou Huang, Xu Wen, Wanling Gao, Fan Zhang, Chen Zheng, Lei Wang, Hainan Ye, Kai Hwang, Zujie Ren, et al. Edge aibench: towards comprehensive end-to-end edge computing benchmarking. In *International Symposium on Benchmarking, Measuring and Optimization*, pages 23–30. Springer, 2018.

[20] Chengjin Yu, Yanping Zhang, Mithun Mukherjee, and Jaime Lloret. Augmented semi-supervised learning for salient object detection with edge computing. *IEEE Wireless Communications*, 2022.

[21] He Li, Kaoru Ota, and Mianxiong Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. *IEEE network*, 32(1):96–101, 2018.

[22] Zakaria Maamar, Thar Baker, Mohamed Sellami, Muhammad Asim, Emir Ugljanin, and Noura Faci. Cloud vs edge: Who serves the internet-of-things better? *Internet Technology Letters*, 1(5):e66, 2018.

[23] Zakaria Maamar, Thar Baker, Noura Faci, Emir Ugljanin, Mohammed Al Khafajiy, and Vanilson Burégio. Towards a seamless coordination of cloud and fog: illustration through the internet-of-things. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 2008–2015, 2019.

[24] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[25] Mingzhe Chen, Nir Shlezinger, H Vincent Poor, Yonina C Eldar, and Shuguang Cui. Communication-efficient federated learning. *Proceedings of the National Academy of Sciences*, 118(17):e2024789118, 2021.

[26] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

[27] Jin Kyu Kang, Hyung Gil Hong, and Kang Ryoung Park. Pedestrian detection based on adaptive selection of visible light or far-infrared light camera image by fuzzy inference system and convolutional neural network-based verification. *Sensors*, 17(7):1598, 2017.

[28] Xiujun Shu, Xiao Wang, Xianghao Zang, Shiliang Zhang, Yuanqi Chen, Ge Li, and Qi Tian. Large-scale spatio-temporal person re-identification: Algorithms and benchmark. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[29] Chengyan Zhong, Xiaoyu Jiang, and Guanqiu Qi. Video-based person re-identification based on distributed cloud computing. *Journal of Artificial Intelligence and Technology*, 1(2):110–120, 2021.

[30] Zhedong Zheng, Liang Zheng, and Yi Yang. Pedestrian alignment network for large-scale person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(10):3037–3045, 2018.

[31] Bipin Gaikwad and Abhijit Karmakar. Smart surveillance system for real-time multi-person multi-camera tracking at the edge. *Journal of Real-Time Image Processing*, 18(6):1993–2007, 2021.

[32] Hyuk-Jin Jeong, Hyeon-Jae Lee, Chang Hyun Shin, and Soo-Mook Moon. Ionn: Incremental offloading of neural network computations from mobile devices to edge servers. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 401–411, 2018.

[33] Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. Distributed deep neural networks over the cloud, the edge and end devices. In *2017 IEEE 37th international conference on distributed computing systems (ICDCS)*, pages 328–339. IEEE, 2017.

[34] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[35] Swarna Priya Ramu, Parimala Boopalan, Quoc-Viet Pham, Praveen Kumar Reddy Maddikunta, Thien Huynh-The, Mamoun Alazab, Thanh Thi Nguyen, and Thippa Reddy Gadekallu. Federated learning enabled digital twins for smart cities: Concepts, recent advances, and future directions. *Sustainable Cities and Society*, 79:103663, 2022.

[36] Jed Mills, Jia Hu, and Geyong Min. Communication-efficient federated learning for wireless edge intelligence in iot. *IEEE Internet of Things Journal*, 7(7):5986–5994, 2019.

[37] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.

[38] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–23, 2022.

[39] Yogesh Kumar and Ruchi Singla. Federated learning systems for healthcare: perspective and recent progress. *Federated Learning Systems*, pages 141–156, 2021.

[40] Weiming Zhuang, Yonggang Wen, Xuesen Zhang, Xin Gan, Daiying Yin, Dongzhan Zhou, Shuai Zhang, and Shuai Yi. Performance optimization of federated person re-identification via benchmark analysis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 955–963, 2020.

[41] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Joint optimization in edge-cloud continuum for federated unsupervised person re-identification. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 433–441, 2021.

[42] Lei Zhang, Guanyu Gao, and Huaizheng Zhang. Spatial-temporal federated learning for lifelong person re-identification on distributed edges. *arXiv preprint arXiv:2207.11759*, 2022.

[43] Fengxiang Yang, Zhun Zhong, Zhiming Luo, Shaozi Li, and Nicu Sebe. Federated and generalized person re-identification through domain and feature hallucinating. *arXiv preprint arXiv:2203.02689*, 2022.

[44] Guile Wu and Shaogang Gong. Decentralised learning from independent multi-domain labels for person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2898–2906, 2021.

[45] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In *Proceedings of the second workshop on distributed infrastructures for deep learning*, pages 1–8, 2018.

[46] Wei Ying, Yu Zhang, Junzhou Huang, and Qiang Yang. Transfer learning via learning to transfer. In *International Conference on Machine Learning*, pages 5085–5094. PMLR, 2018.

[47] Arpita Gupta, Pratik Pawade, and Ramadoss Balakrishnan. Deep residual network and transfer learning-based person re-identification. *Intelligent Systems with Applications*, page 200137, 2022.

[48] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database, 2009. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2020.

[49] Sk Miraj Ahmed, Aske R Lejbolle, Rameswar Panda, and Amit K Roy-Chowdhury. Camera on-boarding for person re-identification using hypothesis transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12144–12153, 2020.

[50] Guoan Cheng, Junyu Shi, Hao Wang, Long Chen, Jiaxi Guo, and Shengke Wang. A study on pedestrian re-identification based on transfer learning. In *2021 6th International Conference on Image, Vision and Computing (ICIVC)*, pages 112–118. IEEE, 2021.

[51] Chia-Yu Chen, Jiamin Ni, Songtao Lu, Xiaodong Cui, Pin-Yu Chen, Xiao Sun, Naigang Wang, Swagath Venkataramani, Vijayalakshmi Viji Srinivasan, Wei Zhang, et al. Scalecom: Scalable sparsified gradient compression for communication-efficient distributed training. *Advances in Neural Information Processing Systems*, 33:13551–13563, 2020.

[52] Sawsan AbdulRahman, Hanine Tout, Hakima Ould-Slimane, Azzam Mourad, Chamseddine Talhi, and Mohsen Guizani. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet of Things Journal*, 8(7):5476–5497, 2020.

[53] https://www.ekkono.ai/wp-content/uploads/2020/12/swp-federated-learning-ekkono-solutions-may. 2020.

[54] Yuhao Zhou, Qing Ye, and Jiancheng Lv. Communication-efficient federated learning with compensated overlap-fedavg. *IEEE Transactions on Parallel and Distributed Systems*, 33(1):192–205, 2021.

[55] Yunfan Ye, Shen Li, Fang Liu, Yonghao Tang, and Wanting Hu. Edgefed: Optimized federated learning based on edge computing. *IEEE Access*, 8:209191–209198, 2020.

[56] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[57] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

[58] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 789–792, 2014.

[59] Dangwei Li, Zhang Zhang, Xiaotang Chen, Haibin Ling, and Kaiqi Huang. A richly annotated dataset for pedestrian attribute recognition. *arXiv preprint arXiv:1603.07054*, 2016.

[60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[61] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al. Flower: A friendly federated learning framework. 2022.

[62] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[63] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[64] https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. 2021.

[65] Guan'an Wang, Shaogang Gong, Jian Cheng, and Zengguang Hou. Faster person re-identification. In *European conference on computer vision*, pages 275–292. Springer, 2020.

[66] Patrick Sudowe, Hannah Spitzer, and Bastian Leibe. Person attribute recognition with a jointly-trained holistic cnn model. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 87–95, 2015.

[67] Kai Yu, Biao Leng, Zhang Zhang, Dangwei Li, and Kaiqi Huang. Weakly-supervised learning of mid-level features for pedestrian attribute recognition and localization. *arXiv preprint arXiv:1611.05603*, 2016.

[68] S Raghavendra, SK Abhilash, Venu Madhav Nookala, S Kaliraj, et al. Efficient deep learning approach to recognize person attributes by using hybrid transformers for surveillance scenarios. *IEEE Access*, 11:10881–10893, 2023.

[69] Pengyu Xie, Xin Xu, Zheng Wang, and Toshihiko Yamasaki. Sampling and re-weighting: Towards diverse frame aware unsupervised video person re-identification. *IEEE Transactions on Multimedia*, 24:4250–4261, 2022.

Conflict of Interest Statement

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: