POLITECNICO DI TORINO
Repository ISTITUZIONALE

Collaborative Visual Place Recognition through Federated Learning

Original
Collaborative Visual Place Recognition through Federated Learning / Dutto, M., Berton, G., Caldarola, D., Fani, E., Trivigno, G., Masone, C.. - (2024), pp. 4215-4225. (IEEE / CVF Computer Vision and Pattern Recognition Conference Workshop Seattle (USA) 17-18 June 2024) [10.1109/CVPRW63382.2024.00425].

Availability:
This version is available at: 11583/2987725 since: 2024-04-11T07:49:14Z

Publisher:
IEEE/CVF

Published
DOI:10.1109/CVPRW63382.2024.00425

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

01 November 2026

This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Collaborative Visual Place Recognition through Federated Learning

Mattia Dutto*, Gabriele Berton, Debora Caldarola, Eros Fanì, Gabriele Trivigno, Carlo Masone
Politecnico di Torino
name.surname@polito.it
*Corresponding Author: Mattia Dutto

Abstract

Visual Place Recognition (VPR) aims to estimate the location of an image by treating it as a retrieval problem. VPR uses a database of geo-tagged images and leverages deep neural networks to extract a global representation, called descriptor, from each image. While the training data for VPR models often originates from diverse, geographically scattered sources (geo-tagged images), the training process itself is typically assumed to be centralized. This research revisits the task of VPR through the lens of Federated Learning (FL), addressing several key challenges associated with this adaptation. VPR data inherently lacks well-defined classes, and models are typically trained using contrastive learning, which necessitates a data mining step on a centralized database. Additionally, client devices in federated systems can be highly heterogeneous in terms of their processing capabilities. The proposed FedVPR framework not only presents a novel approach for VPR but also introduces a new, challenging, and realistic task for FL research, paving the way to other image retrieval tasks in FL.

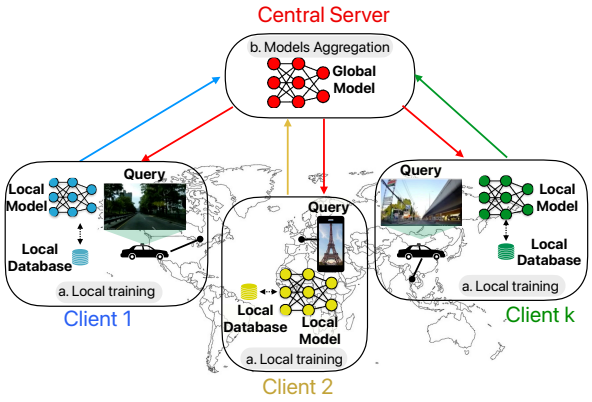


Figure 1. **Federated Visual Place Recognition** (FedVPR): we revisit the training of Visual Place Recognition models from the perspective of Federated Learning, with clients distributed across geographical areas, each possessing heterogeneous computational and communication resources and availability. Instead of relying on a central database for mining, each client builds its own database of geo-tagged images and uses it for local training based on contrastive learning (step a.). Subsequently, it communicates its model weights to the server, where they are aggregated into a new global model (step b.).

1. Introduction

The ability to recognize the place depicted in a picture is of the utmost importance for many modern applications performed by camera-equipped mobile systems. For example, in autonomous driving and mobile robotics, this ability is used for localization in instances where GPS measurement is unavailable or unreliable [49, 64], or in facilitating loop closure within SLAM (Simultaneous Localization and Mapping) pipelines [38]. Additionally, mobile phone applications heavily rely on this functionality for tasks like scene categorization [18] and augmented reality support [60]. Likewise, wearable devices leverage this capability to provide useful information to the user [19]. From a technical perspective, this task is referred to as *Visual Place Recognition* (VPR) [51] and is naturally framed as an image retrieval problem. The query image to be localized is compared via features-space k-nearest neighbor (kNN) [55] to a database of images representing the known or already-visited places. Given that the database samples are usually labeled with geo-tags (such as GPS coordinates), the most similar images retrieved from the database serve as hypotheses of the queried location. This approach entails representing each image with a single vector (*global feature descriptor*) so that the kNN can efficiently compute the similarity between two images, *e.g.*, as an Euclidean distance.

Recent research on VPR has been focusing predominantly on the development of deep neural networks capable of extracting global feature descriptors that are both compact and highly informative for place recognition while leveraging large collections of data from highly heteroge-

neous distributions [2, 3, 10, 12, 34, 39]. However, this centralized formulation assumes the images are readily available on one computer or a central server, which does not suit the distributed nature of the VPR applications previously discussed well. In an ideal scenario where mobile phones, wearable devices, and autonomous vehicles are deployed across numerous cities globally, it becomes crucial to leverage images collected by these diverse distributed devices without transferring their data to a central server, both for cost and privacy-related reasons. Furthermore, it would be beneficial to leverage the onboard computational capabilities of these devices to aid in model training.

In light of these considerations, in this work we question *how to revisit the training of VPR models from the perspective of Federated Learning* (FL) [52], a distributed paradigm where multiple devices (*i.e.*, clients) exchange model parameters updates with a central server to learn a shared global model, without any transfer of local data (see Fig. 1). Some notable challenges make the adaptation of VPR to FL not trivial. Unlike the conventional FL literature that revolves around classification problems [16, 17, 58], VPR lacks a clear division of data into classes. Instead, the collected images are labeled with continuous space annotations (commonly in the form of GPS coordinates), and models are usually trained with contrastive learning techniques [5], which are often performed in conjunction with computationally heavy mining over a large centralized database [5, 27, 37, 45]: in a federated setting, this would be unfeasible due to (i) the low computational capacity of the clients and (ii) the privacy concerns that a centralized database would create. By addressing these challenges, this paper introduces **FedVPR**, the first formulation of VPR in a federated learning paradigm.

Contributions:

- We introduce the first formulation of the VPR task in a federated learning framework. The importance of this formulation is twofold: for the VPR field, it opens up a new research direction with important practical implications; for the FL field, it provides a new downstream task that can broaden the horizon of the research community.
- We propose a new splitting of the worldwide Mapillary Street-Level-Sequences (MSLS) dataset [68] into federated clients, designed to replicate realistic scenarios with varying degrees of statistical heterogeneity across them.
- We deal with clients' data heterogeneity through critical design decisions such as client split, local iteration scheduling, and data augmentation, achieving centralized-level performances while accounting for power and computational requirements.

2. Related work

Visual Place Recognition. (VPR) aims to geolocate a given input photo, called *query*, by comparing it to a set of geo-tagged images (*i.e.*, with known GPS position), called *database* [51]. Modern VPR methods leverage deep neural networks to extract global feature descriptors that provide a compact representation to perform a similarity search. An important milestone in this sense is the work by Arandjelović et al. [5], which introduced a learnable aggregation layer called NetVLAD and a training protocol that leveraged street-view imagery through a triplet loss. The triplet loss paradigm and NetVLAD layer have been used with slight modifications in a number of successive papers [27, 29, 37, 45, 53, 54]. Since then, various innovations have been proposed, *e.g.*, in the aggregation layers [3, 7, 56, 57, 65, 72], architecture [37, 67, 72], inference protocol [6, 66], training procedures [2, 10, 12, 27, 33, 39], adaptation techniques [4, 54], post-processing strategies [9, 29, 67, 72], use of foundational models [33, 36], as well as exploitation of temporal information [13, 25, 26, 53, 70].

Recently, the release of increasingly large datasets [2, 10, 68] and the recognition that the expensive mining and large outputs of the traditional "triplet-loss plus NetVLAD" paradigm hampers scalability [11] has led to the emergence of solutions that can learn more efficiently from the data. For example, CosPlace [10] and its derivative works [12, 13] use a mining-less classification proxy for training, whereas GCL [39] overcomes the expensive mining by leveraging graded similarity labels and integrate this with a generalized contrastive loss. Conv-AP [2] and MixVPR [3] instead rely on a multi-similarity loss with more efficient online mining enabled by a curated dataset. Despite their strong performance, these methods are unsuitable for federated learning because they either (i) require the full database to initialize training [10, 12], (ii) require batch sizes of hundreds of images from a curated dataset to converge [2, 3], (iii) rely on large models and need additional similarity labels [39], or (iv) need a re-ranking step [8, 9, 29, 72], all too expensive to be performed on the clients.

These factors make it challenging to apply federated learning to VPR. Indeed, there are a few works that *deploy* a single VPR model among multiple agents, which then localize collaboratively by fusing their descriptors [43] or predictions [38] in a consensus-like mechanism, but without any local or coordinated training. To the best of our knowledge, this is the first work that uses a federated learning approach to *learn* a global VPR model.

Federated learning. Federated Learning (FL) [52] enables learning from private data at the edge and is based on the exchange of model parameters between multiple clients and a central server over several communication rounds. Each client trains a local copy of a common global model independently on its own private data and only sends back the updated parameters, which are then aggregated on the server side. The server remains unaware of any sensitive information, safeguarding privacy. The versatility of FL

has led to its successful application in various domains, ranging from medical imaging [48] to autonomous driving [23, 24, 62] and natural language processing [44, 47, 73]. Despite these accomplishments, the exploration of FL's potential in more complex vision applications remains incomplete [16]. This study marks a significant step in advancing FL, presenting the first reexamination of VPR within the federated setting. Adapting VPR to FL poses a major challenge due to the task's reliance on mining large datasets. Furthermore, as VPR does not hinge on a discrete label space — unlike the majority of FL literature centered around classification methods [16, 17] — this work introduces a novel case study, broadening the horizons of the FL research community. Aiming at studying real-world scenarios, we also take into consideration the challenges arising in realistic federated settings, namely *statistical* and *system heterogeneity* [41, 58, 71]. Differently from standard distributed training scenarios, the clients are heterogeneous in terms of the distribution of the local data and computational capabilities and availability over time. This heterogeneity may arise from factors such as the users' geographical locations or differences in internet access. Due to such distribution shifts, the learning trend becomes inherently noisy and unstable [1, 16, 58] and achieving a target performance necessitates more training rounds, impacting the communication network. Several approaches tackle this issue. Some focus on regularizing local training to prevent local objectives from deviating significantly from the global one [1, 35, 42]. Others aim to virtually equalize the number of samples across clients (*e.g.*, FedVC [32]) or group similar clients together [15, 22, 28, 61]. Following the latter line of works, we further investigate hierarchical FL [14, 20] for VPR. Clients found in close geographical locations are grouped together, with each cluster referring to a distinct server. All the servers communicate with a first-level additional server, orchestrating the overall training process. This approach facilitates a scalable VPR system that can adapt to regional preferences while maintaining a globally consistent framework.

3. Method

3.1. Centralized VPR

The task of VPR is commonly approached as an image retrieval problem. Given a query image, the goal is to find the most similar matches within a geo-tagged database to infer the query's location. When training a VPR model, we aim to learn a function $F_\theta : \mathbb{X} \to \mathbb{D}$ parameterized by $\theta \in \mathbb{R}^p$ which projects each image sample $x \in \mathbb{X}$ into a common embedding space $\mathbb{D}$ of dimensionality d. Intuitively, F_θ should provide an embedding space where different representations of the same place (*e.g.*, the same building seen with different perspectives or illuminations) should be close to each other while simultaneously being far from representations of other places. VPR models are commonly trained with contrastive losses [5, 11, 27, 69], which rely on feeding the model with samples from the same place (*positives*) and samples from different, although potentially similar, locations (*negative*). The most common approach is to use a triplet loss [5], which takes a query (*anchor*), a positive and a negative image, and aims at bringing the query and the positive sample closer in the features space, while pushing away the negative one. However, when using such formulations of the loss, if the chosen negative's embeddings are already far away from the query's ones (a *trivial negative*), the loss will be close to zero, thus leading to uninformative gradients. To circumvent this issue, hard negatives (*i.e.*, negatives close to the query in features space, or visually similar) must be selected, enabling the model to reach higher performances [11]. The hard negatives selection process takes the name of **mining** and is a time-consuming technique performed throughout the training phase to select ever-increasing difficult negatives for each given query. Formally, for a given training query $q \in \mathbb{X}$, we want to obtain a training triplet $(q, \tilde{p}^q, \tilde{n}^q)$, where $\tilde{p}^q$ is its positive image and $\tilde{n}^q$ the negative one. The set of potential positives $\mathcal{P} := \{p_i^q\}$, commonly defined as the images within a threshold $\tau = 25$ meters from the query [5, 37, 72], is retrieved using the GPS label. The set of negatives $\mathcal{N} := \{n_i^q\}$ instead contains all the images geographically far from the query and is obtained following the symmetrically opposite approach. Since GPS labels alone are not enough to determine whether an image actually depicts the same visual content (*e.g.*, close-by images could point in opposite directions), the current estimate of F_θ is used to compute the Euclidean distance $D_\theta(q, p_i^q)$. Thus, the candidate with the highest probability of being a true positive is selected according to the criteria

$$\tilde{p}^q = \underset{p_i^q \in \mathcal{P}}{\operatorname{argmin}} \, D_\theta \left(q, p_i^q\right). \tag{1}$$

On the other hand, hard negatives are selected as those who are closest to the query in the embedding space (therefore visually similar), while still being geographically far.

$$\tilde{n}^q = \underset{n_i^q \in \mathcal{N}}{\operatorname{argmin}} \, D_\theta \left(q, n_i^q\right). \tag{2}$$

Finally, the loss for the selected triplet $(q, \tilde{p}^q, \tilde{n}^q)$ is

$$\mathcal{L}_\theta = \max\left(D_\theta^2\left(q, \tilde{p}^q\right) - D_\theta^2\left(q, \tilde{n}^q\right) + m, 0\right), \tag{3}$$

where m is the margin hyperparameter.

Finally, while the above procedure (triplet loss with negative mining) is not the only way to train VPR models, its computational affordability makes it suitable for a federated

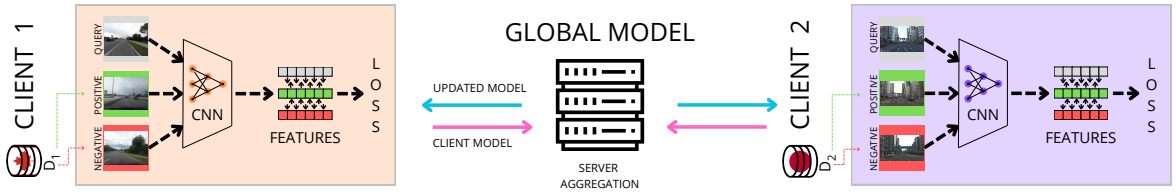


Figure 2. **FedVPR training.** At each round t, the server sends the current global model to a set of active clients, *e.g. client 1* and *client 2* in the figure. Each client i has access to its own local dataset $\mathcal{D}_i$, whose distribution is highly influenced by the user's geographical positions (hence the country flags on the local datasets). Differently from centralized VPR, in FedVPR the mining happens exploiting the client's previously collected images. Thus, given a query image, local optimization is based on a contrastive loss, which relies on a positive and negative images extracted from $\mathcal{D}_i$. Since each local dataset follows a different distribution, the resulting updated parameters vary from client to client (*orange* vs. *purple* updates). Lastly, the local parameters are sent back to the server, where they are aggregated with FedAvg.

learning scenario. On the contrary, latest methods that skip mining altogether either require to know the entire database surface area before starting training [10], or are computationally expensive due to large batch sizes [2, 3] or large models [39], both of which are unsuitable options within a federated scenario.

3.2. Federated Visual Place Recognition (FedVPR)

In this work, we consider a realistic scenario in which the goal is to build a VPR system that will be deployed to a set of clients. While it is expected that some data will be available to pre-train a model in a centralized fashion, in general, it is not easy to obtain large-scale annotated datasets for place recognition. As an example, while large datasets exist available for research purposes, they are usually scraped from Google Street View, which does not allow commercial usage, and collecting a dataset that covers large geographical areas independently is highly expensive. On the other hand, the clients to which the model is deployed are a convenient source of heterogeneous and relevant data, which are fundamental for training a robust feature extractor for the task at hand. However, in the spirit of Federated Learning, the client's privacy must not be violated. Possible examples of the described scenario can be a company deploying a fleet of self-driving vehicles, wanting to improve the performances of its localization-and-mapping pipeline, a swarm of drones, or even general-purpose content-based image retrieval. We frame our analysis in this realistic scenario, demonstrating the criticalities of developing a distributed learning framework. Fig. 2 summarizes FedVPR.

Federated framework. In the standard federated framework, a central server communicates with a set of clients $\mathcal{C}$ over T communication rounds. Clients commonly are edge devices, *e.g.*, smartphones, autonomous vehicles, and IoT sensors. In our setting, each client $k \in \mathcal{K}$ has access to a privacy-protected dataset $\mathcal{D}_k$ made of N_k images $x \in \mathbb{X}$ associated with a GPS location. The training goal is to learn a global shared model $F_\theta : \mathbb{X} \rightarrow \mathbb{D}$ parameterized by $\theta \in \mathbb{R}^p$ without violating the users' privacy. At each round t, the server sends the current global model θ^t to a subset of selected clients $\mathcal{C}^t \in \mathcal{C}$, which trains it using their local data.

Local mining. As detailed in Sec. 3.1, the **mining** process is crucial for learning as it ensures that the network is fed with informative samples during training. Differently from the centralized scenario where the model has access to the whole dataset for training, here we encounter the challenge of performing mining without (*i*) increasing the communication costs by exchanging continuous information between clients and server, *ii*) downloading enormous quantities of data on resource-constrained devices and (*iii*) exchanging data with other clients or the server, which could result in privacy leaks. In FedVPR, to avoid the aforementioned bottlenecks and any privacy concerns, the mining is limited to the database images previously collected by each client. The local data collection likely satisfies the requirement of having access to *hard negative* samples (*i.e.*, visually similar images of different places) to successfully train the feature extractor. However, its limited variability is an important factor that can slow down convergence and ultimately affects performances. More details on this matter are presented in the experimental section.

At the end of local training, each client k sends the updated θ_k^t to the server. The global training objective is solved by aggregating the received updates on the server side. The de-facto standard aggregation algorithm is FedAvg [52], which averages the clients' parameters as

$$\theta^{t+1} \leftarrow \sum_{k \in \mathcal{C}^t} \frac{N_k}{N} \theta_k^t, \tag{4}$$

where $N = \sum_{k \in \mathcal{C}^t} N_k$. As showed by [58], Eq. (4) is equivalent to apply SGD (Stochastic Gradient Descent) [59] with a pseudo-gradient $\Delta\theta^t := \sum_{k \in \mathcal{C}^t} {}^{N_k}/_{N} (\theta^t - \theta_k^t)$ and server learning rate $\eta_s = 1$, where each round is a different optimization step. Thus, Eq. (4) can be generalized to $\theta^{t+1} \leftarrow \theta^t - \textsc{ServerOpt}(\theta^t, \Delta\theta^t, \eta_s, t)$, with ServerOpt being any gradient-based optimizer.

The realistic scenario considered in this work gives us the opportunity to study not only the challenges linked with

the VPR task but also those derived from the federated framework. In particular, in the experiments discussed in Sec. 5, we investigate the effect of *statistical heterogeneity* on the task of VPR in real-world settings where clients' data is usually non-*i.i.d.* with respect to the overall data distribution. Namely, given two clients i and j, their local datasets likely follow a different distribution P, *i.e.*, $P_i \neq P_j$. Consequently, local optimization paths usually lead towards distinct local minima, straying from the global one, a phenomenon referred to as *client drift* [35]. The resulting learning trends are slowed down, unstable, and subject to catastrophic forgetting.

Hierarchical FL. Lastly, since in our setting, the clients' data distributions are linked with the users' geographical locations (*e.g.*, a device likely spends most time within a single region), we additionally explore the hierarchical FL (H-FL) setup [14]. In H-FL, we group clients into K clusters according to their geographical proximity. A specialized model F_{θ_c} is assigned to each cluster c. Once every T_c rounds, the cluster-specific models are aggregated. This implies the existence of multiple servers. We explore a dual-level framework: the first-tier servers handle inter-clusters interactions (*e.g.*, among cities or continents), while second-tier ones manage the intra-cluster exchanges (*e.g.*, between users living in the same city, or continent).

4. Decentralizing the MSLS dataset for FL

Our experiments center on the Mapillary Street-Level-Sequences (MSLS) dataset [68], geographically distributed across 30 cities worldwide, mimicking a FL scenario. The dataset is split into non-overlapping train, validation, and test sets. Each set comprises distinct cities: Amsterdam and Manila for validation, San Francisco and Copenhagen for testing, and the remaining cities for training. Similar to other VPR datasets, each subset is further divided into databases and queries. Queries represent images to be localized, while databases act as the system's prior knowledge of the area. Notably, the dataset excels due to its rich diversity. It encompasses a vast number of cities captured by various users, resulting in a wide range of cameras, weather conditions, times of day, and scenarios across both urban and rural environments. These characteristics perfectly align with the demands of our use case.

4.1. Proposed FL datasets

This work's first contribution lies in proposing three novel splits for the MSLS dataset. These splits mimic real-world scenarios with varying data distributions across devices. Users are grouped based on geographical *proximity*, *similarity* in city features (*e.g.*, architecture), or *randomness*. Tab. 1 summarizes the datasets' characteristics. Additional analyses can be found in **??**.

Table 1. Characteristics of the proposed FL+VPR datasets and associated FedAvg performances.

FL dataset	Radius (m)	Sequences per client	Images per client	Number of clients	R@1 (%)
Centralized	-	-	-	-	66.0 $\pm$ 0.4
Random	-	64 $\pm$ 1	3655 $\pm$ 676	700	40.2 $\pm$ 0.0
Clustering	-	36 $\pm$ 32	2018 $\pm$ 1266	678	57.3 $\pm$ 1.2
Proximity	1000	17 $\pm$ 18	897 $\pm$ 808	1303	51.7 $\pm$ 1.7
	2000	33 $\pm$ 48	1834 $\pm$ 2050	713	61.0 $\pm$ 0.6
	4000	75 $\pm$ 148	4270 $\pm$ 6515	316	**66.1** $\pm$ **0.3**

Proximity. This split emulates user movements within a neighborhood or a proximal geographical area. While clients in smaller towns may explore different localities, users in large cities like Tokyo or San Francisco are inclined to stay within their neighborhoods. The MSLS dataset is first divided geographically, with each city representing a separate entity. Within each city, clients are formed iteratively. An initial query image is chosen from a sequence available in that city. All other geographically close sequences, *i.e.*, within a given radius from the coordinates of the selected image, are then grouped with the chosen query image. This group is considered a valid client only if it contains at least two queries and two database sequences. The resulting number of training clients depends on the chosen radius, which we select in $\{1000, 2000, 4000\}$ meters. Twelve clients are randomly selected from the pool of validated training clients to serve as the validation set. The test set is kept on the server side.

Clustering. The proximity split assumes similar features (*e.g.*, architecture) in nearby areas. However, distant neighborhoods might share more similarities (*e.g.*, busy streets, shops) than geographically close ones. To capture such nuances, the clustering split utilizes the K-means algorithm [50] at the city level, grouping images based on their visual and environmental characteristics. To ensure a balanced number of clients while capturing similarities, we determine the value of K (number of clusters) for each city individually. We use the number of clients obtained in the proximity split with a radius of 2000 meters as a reference point, *i.e.*, $K = 713$. The same selection criterion of the proximity split is then applied to define the valid clients. 12 clients are maintained for validation, and the test set is on the server.

Random. Following the approach of [23, 24, 31], we introduce a *random* split of MSLS to emulate a uniform distribution and facilitate the understanding of the effects of statistical heterogeneity induced by domain shift. Each dataset's client includes images from all cities, and validation is conducted on the same local dataset. If a city does not have enough data for all clients, we duplicate existing sequences until each client can access at least one sequence from each city. Any remaining sequences are redistributed

Table 2. **Centralized baselines**. Comparison of different model architectures (*left*) and pooling layers (*right*) in terms of recall. The average pooling layer is used for the architecture comparison on the left, and ResNet18 truncated on the right.

Backbone	R@1	Total	Trained
ResNet18 trunc.	42.9 ± 2.5	**2.8M**	**2.1M**
ResNet18	$\mathbf{60.1 \pm 0.3}$	11.2M	10M
VGG16	46.3 ± 0.5	14.7M	7M

Pooling	R@1
SPOC [7]	42.9 ± 2.5
MAC [57]	59.4 ± 0.6
GeM [56]	$\mathbf{68.0 \pm 0.3}$

among clients. The test set remains on the server side.

5. Experiments and Results

5.1. Implementation details

This section provides the main implementation details used in our experiments. Additional information can be found in **??**. The used model architecture, unless specified otherwise, is a ResNet18 [30] truncated at the third convolutional layer, with GeM pooling [56]. In local training, we use a batch size of 2 triplets per iteration, with each triplet comprising 5 negative examples for each query alongside its positive counterpart. The learning rate is 1×10^{-5} with Adam as both local optimizer and optimizer in the centralized runs. On the server side, unless otherwise specified, FedAvg [52] is used for model aggregation. In the centralized experiments, the training continues until the model converges, incorporating an early-stopping mechanism based on validation accuracy. In contrast, in the FL framework, each round engages 5 clients, with each client running a single local epoch. This process is repeated across a total of $T = 300$ rounds. Validation is conducted using a subset of 12 clients that do not participate in the training phase. Lastly, testing is directly handled by the server.

The Hierarchical Federated Learning (H-FL) [20, 46] experiments propose two hierarchy types, delineated by geographical proximity: *City* and *Continental* levels, where clients within the same city or continent respectively are aggregated to form the cluster-specific models. This results in 21 clusters in the former case and 4 in the latter. We employ the classical SGD server optimizer in our hierarchical experiments and select 5 clients from each cluster per round.

All experiments are conducted using an image resolution of 288×384 pixels, which provides a good trade-off between speed and results. Notably, we refrain from employing any form of data augmentation in our methodology. To ensure the robustness and reliability of the results, all experimental outcomes are averaged over 3 distinct runs.

5.2. Centralized baselines

In FL, the choice of network architectures is restricted by the communication bottleneck and the constrained computational capabilities of individual clients. Consequently, lightweight backbones with fewer parameters are preferred

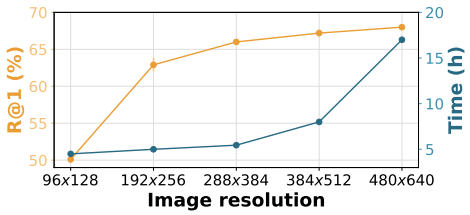


Figure 3. **Centralized setting**. Comparison of R@1 (%) and computational time (hours) when varying the image resolution. Resolution greatly affects training time, and an optimal trade-off can be attained with minimal performance drops.

to alleviate communication and computation burdens. Table 2 presents a comparison of backbone architectures and aggregation layers in a centralized setting, which serve as a baseline for the FL experiments. We consider three networks pre-trained on ImageNet [21]: ResNet18 [30] truncated at the third layer, ResNet18 with fine-tuning limited to the last block, and VGG16 [63]. Regarding aggregation layers, we test different pooling strategies, namely SPOC [7], MAC [57], and GeM [56]. These methods apply a pooling operation on the feature maps provided by the backbone, obtaining a single embedding for each image, whose dimensions are determined by the number of channels of the chosen backbone. NetVLAD [5] is another popular aggregation layer for VPR, which usually grants robust performances [11]. Nevertheless, some drawbacks make its adoption impractical in a federated scenario. Firstly, it outputs large descriptors (ranging from 16k to 64k dimensionality, depending on the backbone), which clients would have to store and use for nearest neighbor search. Moreover, it requires to initialize a set of centroids (*visual words*) using a diverse set of images [11], which would have to be provided by the server, increasing the communication costs and eventually raising privacy concerns. Thus, in our analysis, we consider simpler pooling layers to be viable alternatives.

Based on the results in Tab. 2, we select ResNet18 truncated as our architecture, capitalizing on the benefits of a reduced parameter count. The adoption of GeM as the pooling layer is justified by the observed performance improvements. However, GeM increases the computational time, which is a restriction in FL settings, where edge devices have access to limited resources. To mitigate this, we compare various image resolutions, considering both training time and performance metrics in Fig. 3. We settle on the 288×384 resolution as the optimal choice, striking the best balance between recall and training time. With this resolution, we get a final centralized recall of 66.0 ± 0.4.

5.3. FL baselines

Splits comparison. With the centralized baselines established, we conduct an analysis of the vanilla FedAvg algorithm across the different introduced datasets, as illus-

Table 3. Comparison of the vanilla baseline FedAvg with Hierarchical FL methods and various server optimizers. Notation: **CC** for continent-level middle servers in H-FL, **C** for city-level middle servers, SGDm for SGD with server-side momentum, T rounds, C clients participating at each round.

Algorithm	Server Optimizer	R@1 (%)
FedAvg	SGD	61.0 $\pm$ 0.6
	SGDm	61.2 $\pm$ 1.4
	Adam	61.1 $\pm$ 1.2
	AdaGrad	61.6 $\pm$ 0.3
H-FL (**CC**)	SGD	46.9 $\pm$ 1.3
FedAvg $T = 75$, $C = 20$	SGD	55.6 $\pm$ 0.8
H-FL (**C**)	SGD	33.3 $\pm$ 0.1
FedAvg $T = 15$, $C = 105$	SGD	44.2 $\pm$ 0.4

trated in Tab. 1. Interestingly, the performance of FedAvg on the Random FL dataset significantly lags behind that of other ones, despite the scenario closely resembling uniform splits as seen in prior works [23, 24, 31]. However, as highlighted in [11], optimal performance necessitates hard negatives to be situated within a distance range of 25 meters to a few kilometers from the query, a condition not met in this dataset where images within the same client can belong to various locations worldwide. Our experiments on the Proximity and Clustering FL datasets yielded comparable performance. Interestingly, the Proximity split achieved slightly better results on average. This difference is likely because images within clients of the Proximity split have closer GPS coordinates compared to those in the Clustering split. Additionally, clients in the Proximity FL datasets with radii of 2000m and 4000m also contain a larger number of images compared to their counterparts in the Clustering datasets. The proximity experiment employing a radius of 4000m demonstrates performance levels roughly akin to the centralized baseline. However, opting for a larger radius results in fewer clients, each possessing a greater number of images and sequences, thereby resembling a cross-silo scenario [40]. Conversely, reducing the radius yields a larger number of clients but with a markedly limited quantity of images and sequences per client. Given all these considerations, we shift our focus solely to the **proximity** FL dataset with a **2000m radius** in the upcoming experiments. This choice strikes a balance between the number of clients and the volume of data per client.

Baselines. Table 3 presents a comparative analysis between FedAvg with various server-side optimizers and baselines sourced from the Hierarchical Federated Learning (H-FL) literature [20, 46]. As described in Sec. 5.1, we distinguish between H-FL at city (C) and continent level (CC). To ensure a fair comparison with H-FL, which selects 5 clients from each cluster per round, we additionally run FedAvg with 20 clients per round and $T = 75$, and 105 participating clients for $T = 15$ rounds. Both H-FL experiments exhibit a reduction in recall by approximately 10%. We posit that this substantial decline in performance stems from clusters tending to overfit the local distributions, thereby diminishing the meaningfulness of aggregation compared to training with all clients collectively. Concerning the server optimizers, AdaGrad demonstrates slightly superior performance compared to others. As a result, we opt to employ the standard SGD without momentum for the other baselines.

Table 4. Addressing the clients' quantity heterogeneity. We compare the R@1 (%) of the FedAvg baseline (grey background) with the ones of FedAvg and FedVC [32] with a fixed number of iterations per client per round. B is the local mini-batch size.

Local Iterations	Rounds	FedAvg	FedVC
$\min\left(\lfloor \lvert\mathcal{D}_k\rvert / B \rfloor, 2500\right)$	300	61.0 $\pm$ 0.6	-
125	3200	66.6 $\pm$ 0.8	62.3 $\pm$ 1.1
250	1600	66.0 $\pm$ 1.7	65.9 $\pm$ 1.0
500	800	66.4 $\pm$ 1.6	**67.7 $\pm$ 0.4**
1000	400	61.7 $\pm$ 2.4	66.8 $\pm$ 0.5
2000	200	58.8 $\pm$ 1.8	65.2 $\pm$ 0.9
4000	100	57.3 $\pm$ 2.5	60.6 $\pm$ 1.0

5.4. Data Quantity Skewness in FedVPR

Table 1 highlights significant variations in the number of sequences or images among clients (*i.e.*, *quantity heterogeneity*), particularly evident in the Proximity split with a radius of 2000m - our reference federated dataset. This section analyzes how this phenomenon affects the final performance.

In heterogeneous settings, an increased number of local training steps (updates within a client over a batch of data) fosters client drift and destructive interference during aggregation [16, 35]. Thus, a larger local dataset leads to more updates and potentially negatively impacts the training process. Motivated by these insights, Tab. 4 investigates how the data quantity skewness and the number of local training iterations affect performances of algorithms trained within the FedVPR framework, focusing on FedAvg and the state-of-the-art algorithm FedVC (Federated Virtual Clients) [32]. FedVC specifically addresses variations in client data sizes by splitting large datasets into smaller clients and replicating smaller ones. This ensures all participating *virtual* clients contribute roughly the same amount of data during each training round. To prevent knowledge loss, larger clients are resampled with higher probability.

Given a fixed amount of total iterations I_{tot}, we either vary the local iterations I_{loc}, or the training rounds T such that $I_{loc} \times T \times |\mathcal{C}^t| = I_{tot}$. With larger I_{loc}, smaller datasets are used multiple times within a client, while fewer iterations might lead to an incomplete view of larger datasets. We set $I_{tot} = 2,000,000$, $|\mathcal{C}|^t = 5$ and vary I_{loc} and T.

This analysis reveals that reducing the influence of data imbalances can improve performance by up to 5%. However, there exists a trade-off between communication rounds and final accuracy. As shown in Tab. 4, increasing T (more

Table 5. **Augmentation**. Comparison of data augmentation strategies. The baseline represents training without augmentation, while the Client-specific color jitter strategy aims at simulating the system heterogeneity typical of FL Sec. 5.5.

Augmentation	R@1 (%)
Baseline	61.0 ±0.6
Client-specific color jitter	53.5 ±2.5
Color jitter	64.7 ±0.9
Color jitter + random resize crop	65.7 ±0.6

communication) while reducing the local steps leads to performance improvement. Conversely, excessively increasing local computation at the expense of the number of rounds deteriorates the final performance. Finally, FedVC's sampling strategy, which favors larger clients, consistently improves performance when each device performs more than 500 local updates per round. However, for fewer local updates (125 and 250), FedVC shows a decrease in accuracy.

5.5. Heterogeneity of Local Augmentations

Table 5 illustrates the performance variations with different levels of data augmentation during training using FedAvg. Varying levels of color jitter among clients (*client-specific color jitter*) simulate significant statistical heterogeneity. This heterogeneity could arise from various sources, as client devices capture data with varying camera qualities and under diverse environmental conditions (lighting, weather). As expected, this experiment results in a severe performance degradation (-7.5 points w.r.t. the baseline). To isolate the effects of color jitter augmentation from statistical heterogeneity, we apply the same color jitter augmentation uniformly across all clients, revealing an improvement of nearly 4 percentage points in R@1 w.r.t. the baseline. An additional random resized crop yields an increase in performance of 1 percentage point. We posit that the substantial benefits observed with stronger data augmentation primarily stem from the relatively small local datasets. Without robust augmentation, clients tend to overfit on local data, culminating in a meaninglessly aggregated model at the end of each round.

5.6. Impact of Data Distribution on Local Mining

As discussed in Sec. 3.1, the mining algorithm is a crucial factor affecting performances in VPR. In centralized training, the model can access the entire database to select negative examples. On the other hand, in FL, clients can only rely on their local collection of images, which come from a limited geographical area, thus limiting the negative sampling distribution. To study the extent to which this limitation represents an issue for FL, we run centralized experiments with limited available images during mining (Tab. 6). For each query, the server accesses negative images only within the closest N database sequences.

Table 6. **Local mining**. In a centralized scenario, we constrain the mining procedure to select negatives within the closest database sequences, rather than the global database, to study its effect. This setting emulates a federated scenario.

Mining	# Sequences	DB size	R@1
Baseline	-	1k	66.0 ±0.4
Local	333	28k	**68.8 ±0.2**
Local	20	3k	58.0 ±0.9

The results confirm our expectation that focusing on an overly restricted geographical range of images can hinder performance in VPR. Interestingly, having an extremely wide geographical spread of images doesn't necessarily lead to better learning either. Limiting the image range to a moderate level ($\approx$ 333 sequences, *i.e.*, neighborhood to city scale) leads to a slight performance improvement. This suggests that focusing on a localized area can be beneficial for VPR, motivating FedVPR. However, excessively restricting the range to a very local level (20 sequences) proves detrimental, resulting in performance even lower than a FL approach (cf. Tab. 3). This is likely because most clients in the federated setting have access to a wider variety of images.

These findings challenge the traditional assumption that geographical diversity is essential for VPR and suggest that a balance might exist between geographical scope and training data diversity for optimal VPR results.

6. Conclusion

In this work, we introduced **FedVPR**, a novel Federated Learning framework specifically designed for Visual Place Recognition (VPR) tasks. This approach addresses the growing need for distributed VPR solutions in applications like autonomous vehicles and mobile augmented reality. We analyzed the unique challenges of federated VPR compared to classification tasks and demonstrated that FedVPR can achieve performance comparable to a centralized model while minimizing resource consumption on individual devices. Our exploration identified key design choices impacting the performance-cost trade-off, including the number of local iterations, data augmentation strategies, and image resolution. These simple yet effective tools effectively mitigate statistical heterogeneity, validating the feasibility of the FedVPR setting. We believe this work opens a new avenue for VPR research while offering a realistic and valuable task for federated learning research.

Acknowledgments. This project was supported by CINI (Consorzio Interuniversitario Nazionale per l'Informatica). Computational resources were provided by HPC@POLITO.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization, 2021. 3

[2] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Gsv-cities: Toward appropriate supervised visual place recognition. *Neurocomputing*, 513:194–203, 2022. 2, 4

[3] Amar Ali-bey, Brahim Chaib-draa, and Philippe Giguère. Mixvpr: Feature mixing for visual place recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2998–3007, 2023. 2, 4

[4] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5958–5964. IEEE, 2019. 2

[5] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1437–1451, 2018. 2, 3, 6

[6] Bruno Arcanjo, Bruno Ferrarini, Michael Milford, Klaus D McDonald-Maier, and Shoaib Ehsan. An efficient and scalable collection of fly-inspired voting units for visual place recognition in changing environments. *IEEE Robotics and Automation Letters*, 7(2):2527–2534, 2022. 2

[7] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv:1510.07493*, 2015. 2, 6

[8] Giovanni Barbarani, Mohamad Mostafa, Hajali Bayramov, Gabriele Trivigno, Gabriele Berton, Carlo Masone, and Barbara Caputo. Are local features all you need for cross-domain visual place recognition?, 2023. 2

[9] Gabriele Berton, Carlo Masone, Valerio Paolicelli, and Barbara Caputo. Viewpoint invariant dense matching for visual geolocalization. In *IEEE International Conference on Computer Vision*, pages 12169–12178, 2021. 2

[10] Gabriele Berton, Carlo Masone, and Barbara Caputo. Rethinking visual geo-localization for large-scale applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4878–4888, 2022. 2, 4

[11] Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. Deep visual geo-localization benchmark, 2023. 2, 3, 6, 7

[12] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. Eigenplaces: Training viewpoint robust models for visual place recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11080–11090, 2023. 2

[13] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. Jist: Joint image and sequence training for sequential visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1310–1317, 2024. 2

[14] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020. 3, 5

[15] Debora Caldarola, Massimiliano Mancini, Fabio Galasso, Marco Ciccone, Emanuele Rodolà, and Barbara Caputo. Cluster-driven graph federated learning over multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2749–2758, 2021. 3

[16] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, pages 654–672. Springer, 2022. 2, 3, 7

[17] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019. 2, 3

[18] Mike Chatzidakis, Junye Chen, Oliver Chick, Eric Circlaeays, Sowmya Gopalan, Yusuf Goren, Kristine Guo, Michael Hesse, Omid Javidbakht, Vojta Jina, Kalu Kalu, Anil Katti, Albert Liu, Richard Low, Audra McMillan, Joey Meyer, Steve Myers, Alex Palmer, David Park, Gianni Parsa, Paul Pelzl, Rehan Rishi, Michael Scaria, Chiraag Sumanth, Kunal Talwar, Karl Tarbe, Shan Wang, and Mayank Yadav. Learning iconic scenes with differential privacy. *Expo Talk at International Conference on Machine Learning*, 2023. 1

[19] Ruiqi Cheng, Kaiwei Wang, Jian Bai, and Zhijie Xu. Unifying visual localization and scene recognition for people with visual impairment. *IEEE Access*, 8:64284–64296, 2020. 1

[20] Shunfeng Chu, Jun Li, Kang Wei, Yuwen Qian, Kunlun Wang, Feng Shu, and Wen Chen. Design of two-level incentive mechanisms for hierarchical federated learning, 2023. 3, 6, 7

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[22] Moming Duan, Duo Liu, Xinyuan Ji, Yu Wu, Liang Liang, Xianzhang Chen, Yujuan Tan, and Ao Ren. Flexible clustered federated learning for client-level data distribution shift. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2661–2674, 2021. 3

[23] Eros Fanì, Marco Ciccone, and Barbara Caputo. Feddrive v2: an analysis of the impact of label skewness in federated semantic segmentation for autonomous driving. *arXiv preprint arXiv:2309.13336*, 2023. 3, 5, 7

[24] Lidia Fantauzzo, Eros Fanì, Debora Caldarola, Antonio Tavera, Fabio Cermelli, Marco Ciccone, and Barbara Caputo. Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11504–11511. IEEE, 2022. 3, 5, 7

[25] Sourav Garg and Michael Milford. Seqnet: Learning descriptors for sequence-based hierarchical place recognition.

IEEE Robotics and Automation Letters, 6(3):4305–4312, 2021. 2

[26] Sourav Garg, Madhu Vankadari, and Michael Milford. Seqmatchnet: Contrastive learning with sequence matching for place recognition & relocalization. In *5th Annual Conference on Robot Learning*, 2021. 2

[27] Yixiao Ge, Haibo Wang, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-supervising fine-grained region similarities for large-scale image localization. In *Computer Vision – ECCV 2020*, pages 369–386, Cham, 2020. Springer International Publishing. 2, 3

[28] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020. 3

[29] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 14141–14152, 2021. 2

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[31] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. 5, 7

[32] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated visual classification with real-world data distribution, 2020. 3, 7

[33] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. *arXiv preprint arXiv:2311.15937*, 2023. 2

[34] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition, 2023. 2

[35] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020. 3, 5, 7

[36] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 2023. 2

[37] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geolocalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3251–3260, 2017. 2, 3

[38] Pierre-Yves Lajoie and Giovanni Beltrame. Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems. *IEEE Robotics and Automation Letters*, 9(1):475–482, 2024. 1, 2

[39] María Leyva-Vallina, Nicola Strisciuglio, and Nicolai Petkov. Data-efficient large scale place recognition with graded similarity supervision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 23487–23496, 2023. 2, 4

[40] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020. 7

[41] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 3

[42] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020. 3

[43] Yiming Li, Zonglin Lyu, Mingxuan Lu, Chao Chen, Michael Milford, and Chen Feng. Collaborative visual place recognition, 2023. 2

[44] Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. Fednlp: Benchmarking federated learning methods for natural language processing tasks. *arXiv preprint arXiv:2104.08815*, 2021. 3

[45] Liu Liu, Hongdong Li, and Yuchao Dai. Stochastic attraction-repulsion embedding for large scale image localization, 2019. 2

[46] Lumin Liu, Jun Zhang, Shenghui Song, and Khaled B. Letaief. Hierarchical federated learning with quantization: Convergence analysis and system design. *IEEE Transactions on Wireless Communications*, 22(1):2–18, 2023. 6, 7

[47] Ming Liu, Stella Ho, Mengqi Wang, Longxiang Gao, Yuan Jin, and He Zhang. Federated learning meets natural language processing: A survey. *arXiv preprint arXiv:2107.12603*, 2021. 3

[48] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1013–1023, 2021. 3

[49] Zhuoqun Liu, Fan Guo, Heng Liu, Xiaoyue Xiao, and Jin Tang. CMLocate: A cross-modal automatic visual geolocalization framework for a natural environment without gnss information. *IET Image Processing*, 17(12):3524–3540, 2023. 1

[50] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 5

[51] Carlo Masone and Barbara Caputo. A survey on deep visual place recognition. *IEEE Access*, 9:19516–19547, 2021. 1, 2

[52] 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. 2, 4, 6

[53] Riccardo Mereu, Gabriele Trivigno, Gabriele Berton, Carlo Masone, and Barbara Caputo. Learning sequential descriptors for sequence-based visual place recognition. *IEEE*

Robotics and Automation Letters, 7(4):10383–10390, 2022. 2

[54] Gabriele Moreno Berton, Valerio Paolicelli, Carlo Masone, and Barbara Caputo. Adaptive-attentive geolocalization from few queries: a hybrid approach. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2917–2926, 2021. 2

[55] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2): 1883, 2009. 1

[56] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN Image Retrieval with No Human Annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 2, 6

[57] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016. 2, 6

[58] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *International Conference on Learning Representations*, 2021. 2, 3, 4

[59] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 4

[60] Paul-Edouard Sarlin, Mihai Dusmanu, Johannes L. Schönberger, Pablo Speciale, Lukas Gruber, Viktor Larsson, Ondrej Miksik, and Marc Pollefeys. LaMAR: Benchmarking localization and mapping for augmented reality. In *European Conference on Computer Vision*, pages 686–704, Cham, 2022. Springer Nature Switzerland. 1

[61] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32 (8):3710–3722, 2020. 3

[62] Donald Shenaj, Eros Fanì, Marco Toldo, Debora Caldarola, Antonio Tavera, Umberto Michieli, Marco Ciccone, Pietro Zanuttigh, and Barbara Caputo. Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning, 2022. 3

[63] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 6

[64] Lauri Suomela, Jussi Kalliola, Harry Edelman, and Joni-Kristian Kämäräinen. Placenav: Topological navigation through place recognition. *arXiv preprint arXiv:2309.17260*, 2023. 1

[65] Giorgos Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. *CoRR*, abs/1511.05879, 2016. 2

[66] Gabriele Trivigno, Gabriele Berton, Juan Aragon, Barbara Caputo, and Carlo Masone. Divide&classify: Fine-grained classification for city-wide visual geo-localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11142–11152, 2023. 2

[67] Ruotong Wang, Yanqing Shen, Weiliang Zuo, Sanping Zhou, and Nanning Zheng. Transvpr: Transformer-based place recognition with multi-level attention aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13648–13657, 2022. 2

[68] Frederik Warburg, Søren Hauberg, Manuel López-Antequera, Pau Gargallo, Yubin Kuang, and Javier Civera. Mapillary street-level sequences: A dataset for lifelong place recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2623–2632, 2020. 2, 5

[69] Mubariz Zaffar, Sourav Garg, Michael Milford, Julian Kooij, David Flynn, Klaus McDonald-Maier, and Shoaib Ehsan. VPR-Bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change. *International Journal of Computer Vision*, 129(7): 2136–2174, 2021. 3

[70] Junqiao Zhao, Fenglin Zhang, Yingfeng Cai, Gengxuan Tian, Wenjie Mu, Chen Ye, and Tiantian Feng. Learning sequence descriptor based on spatio-temporal attention for visual place recognition. *IEEE Robotics and Automation Letters*, 9(3):2351–2358, 2024. 2

[71] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021. 3

[72] Sijie Zhu, Linjie Yang, Chen Chen, Mubarak Shah, Xiaohui Shen, and Heng Wang. R2former: Unified retrieval and reranking transformer for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19370–19380, 2023. 2, 3

[73] Xinghua Zhu, Jianzong Wang, Zhenhou Hong, and Jing Xiao. Empirical studies of institutional federated learning for natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 625–634, 2020. 3