Federated Contrastive Learning for Dermatological Disease Diagnosis via On-device Learning

(Invited Paper)

Yawen Wu[1], Dewen Zeng[2], Zhepeng Wang[3], Yi Sheng[3], Lei Yang[4], Alaina J. James[1,5], Yiyu Shi[2], Jingtong Hu[1],
[1]University of Pittsburgh, PA, USA. [2]University of Notre Dame, IN, USA.
[3]George Mason University, VA, USA. [4]University of New Mexico, NM, USA.
[5]University of Pittsburgh Medical Center, PA, USA.
yawen.wu@pitt.edu, dzeng2@nd.edu, zwang48@gmu.edu, ysheng2@gmu.edu,
leiyang@unm.edu, jamesaj@upmc.edu, yshi4@nd.edu, jthu@pitt.edu

2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD) | 978-1-6654-4507-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCAD51958.2021.9643454

***Abstract*—Deep learning models have been deployed in an increasing number of edge and mobile devices to provide healthcare. These models rely on training with a tremendous amount of labeled data to achieve high accuracy. However, for medical applications such as dermatological disease diagnosis, the private data collected by mobile dermatology assistants exist on distributed mobile devices of patients, and each device only has a limited amount of data. Directly learning from limited data greatly deteriorates the performance of learned models. Federated learning (FL) can train models by using data distributed on devices while keeping the data local for privacy. Existing works on FL assume all the data have ground-truth labels. However, medical data often comes without any accompanying labels since labeling requires expertise and results in prohibitively high labor costs. The recently developed self-supervised learning approach, contrastive learning (CL), can leverage the unlabeled data to pre-train a model for learning data representations, after which the learned model can be fine-tuned on limited labeled data to perform dermatological disease diagnosis. However, simply combining CL with FL as federated contrastive learning (FCL) will result in ineffective learning since CL requires diverse data for accurate learning but each device in FL only has limited data diversity. In this work, we propose an on-device FCL framework for dermatological disease diagnosis with limited labels. Features are shared among devices in the FCL pre-training process to provide diverse and accurate contrastive information without sharing raw data for privacy. After that, the pre-trained model is fine-tuned with local labeled data independently on each device or collaboratively with supervised federated learning on all devices. Experiments on dermatological disease datasets show that the proposed framework effectively improves the recall and precision of dermatological disease diagnosis compared with state-of-the-art methods.**

***Index Terms*—Dermatological disease diagnosis, federated learning, contrastive learning, on-device learning**

I. Introduction

Skin diseases are a major global health threat to a tremendous amount of people in the world [1]. These diseases not only injure the physical health including the risk of skin cancer but also can result in psychological problems such as lack of self-confidence and psychological depression due to damaged appearance [2], [3]. Deep learning models have shown great promising in skin diseases diagnosis [3]–[5] and have been widely deployed on mobile devices as mobile dermatology assistants [6]–[8]. These models are trained on a large amount of data with full labels to achieve a high accuracy [9]. When the large-scale datasets for training are not available, the performance of deep learning models will greatly degrade [10]. However, the images of skin disease are usually distributed on mobile devices of patients, which are impractical and even illegal to combine in a single location [10] to form large-scale datasets since data sharing is constrained by the Health Insurance Portability and Accountability Act (HIPAA) [10]. For example, skin disease images can be taken by the cameras of mobile devices and stored for a preliminary self-diagnosis [4], [11]. But patients are usually reluctant to share highly private and sensitive images with the data center. Without large-scale datasets in a single location, it is not doable to perform centralized training for learning an accurate model.

Federated learning (FL) is a distributed learning framework where many mobile devices collaboratively learn a global prediction model without sharing private data [12]. By leveraging FL, distributed data on mobile devices can be used to train an accurate shared model to diagnose skin diseases while keeping data local. Existing FL works assume the local data on devices are fully labeled and use supervised learning for local model updates. However, the assumption of fully labeled data is impractical. For instance, the patients may not want to spend time labeling their skin images captured by cameras of mobile phones. Even voluntary patients may not be able to accurately label all their own images due to the lack of expertise. Therefore, most of the distributed data on devices will be unlabeled, and the deficiency of labels makes supervised FL unrealistic.

Contrastive learning (CL), a recently developed self-supervised learning approach, can learn effective visual representations on data without using labels [13]. By combining CL with FL as federated contrastive learning (FCL), the conventional supervised learning on local devices can be replaced by CL pre-training without using labels. After that, the pre-trained model can be used as the initialization to fine-tune for the diagnosis task with limited labels. In this way, an accurate dermatological disease diagnosis model can be

This work was supported in part by NSF CNS-2122320.

learned by using distributed data with limited labels.

However, simply combining CL into FL cannot achieve optimal performance. This is because existing CL approaches [13]–[15] are originally developed for centralized training on large-scale datasets, assuming sufficiently diverse data is available for training. More specifically, different from supervised learning with cross-entropy loss, in which each image is used independently from other images, CL relies on diverse data to learn the correlation between different images. Without large data diversity, the performance of CL performance will greatly degrade, which also result in a low accuracy on the skin disease diagnosis after fine-tuning with labeled data.

To address this challenge, we propose an on-device FCL framework to enable effective FL with limited labels. This framework has two stages. The first stage is federated self-supervised pre-training. Feature sharing is proposed to improve the data diversity of local contrastive learning while avoiding raw data sharing. Data features encoded in vectors are shared among devices, such that diverse and accurate contrastive information is provided to each device. By leveraging the shared features, representations of higher quality are learned on local devices, which improves the quality of the aggregated model in FL.

The second stage is fine-tuning with limited labeled data. By using the pre-trained model in the first stage as a good initialization, the second stage learns the task of dermatological disease diagnosis by either fine-tuning independently on each device, or fine-tuning collaboratively on all devices by supervised federated learning with limited labels.

In summary, the main contributions of this paper include:

- **Federated contrastive learning (FCL) framework.** We propose an on-device FCL framework to enable effective learning with limited labels for dermatological disease diagnosis. FCL pre-trains the model on distributed unlabeled data to provide a good initialization, followed by fine-tuning with a limited number of labeled data to perform the disease diagnosis.
- **Feature sharing for better local learning.** We propose a feature sharing method to improve the data diversity of local contrastive learning while avoiding raw data sharing for privacy. The shared features provide more diverse features to contrast with during local learning on each mobile device for better representations.
- **More accurate diagnosis and better label efficiency.** Experiments on dermatological disease datasets with various skin colors show superior diagnostic accuracy and label efficiency over state-of-the-art techniques.

II. Background and Related Work

A. Contrastive Learning

Contrastive learning (CL) is a powerful self-supervised method to learn visual representations from unlabeled data [16]–[20]. CL pre-trains a model and provides high generalization performance for downstream tasks such as classification and segmentation [13], [14], [21]. CL learns representations by performing a proxy task of discriminating the image identities. In the learning process, CL minimizes a contrastive loss evaluated on pairs of feature vectors extracted from data augmentations (e.g. cropping, rotation, and color distortion) of the image [22]. By optimizing the contrastive loss, CL maximizes the agreement of representations between transformations of the same identity and minimizes the agreement between different images [23]. As shown in Fig. 1, for an unlabeled input image x, two random transformations $t \sim T$ and $t' \sim T$ are applied to x to produce x_q and x_{k^+}, both of which are then fed into the model f and projection head g to get representations q and k^+. Let Q be a memory bank with K representation vectors stored. By using every feature n in the memory bank Q as negatives, a positive pair q and k^+ are contrasted with every n by the following contrastive loss function.

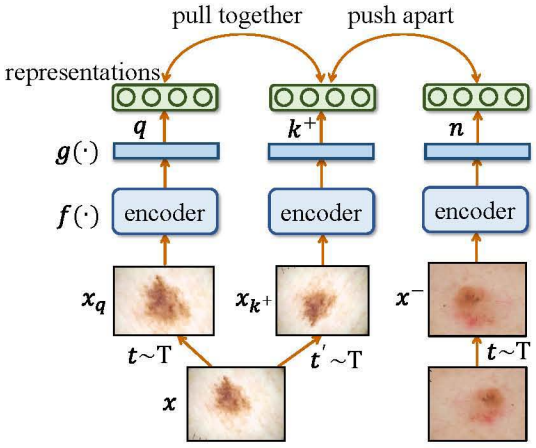

Fig. 1: Illustration of contrastive learning. Two separate data augmentations operators are sampled from the same family of augmentations ($t \sim T$ and $t' \sim T$) and then applied to one image x. The representations of the two transformed versions are pushing close to each other and apart from the representations of other images.

$$\ell_q = -\log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{n \in Q} \exp(q \cdot n/\tau)}. \quad (1)$$

By optimizing the model f to minimize the loss function, effective visual representations can be learned by f.

However, existing CL works are developed for centralized training on large-scale datasets consisting of millions [24] or even billions [13], [25] of images. The large-scale datasets provide sufficiently large data diversity for training. However, in FL each device only has a limited amount of data with limited diversity. Since CL relies on the contrast with different data to achieve high performance, without large data diversity, the performance of CL will greatly degrade.

B. Federated Learning

Federated learning (FL) aims to collaboratively learn a global model for distributed devices while keeping data local devices for privacy [26]. In FL, the training data are distributed among devices, and each device has a subset of the training data. In a

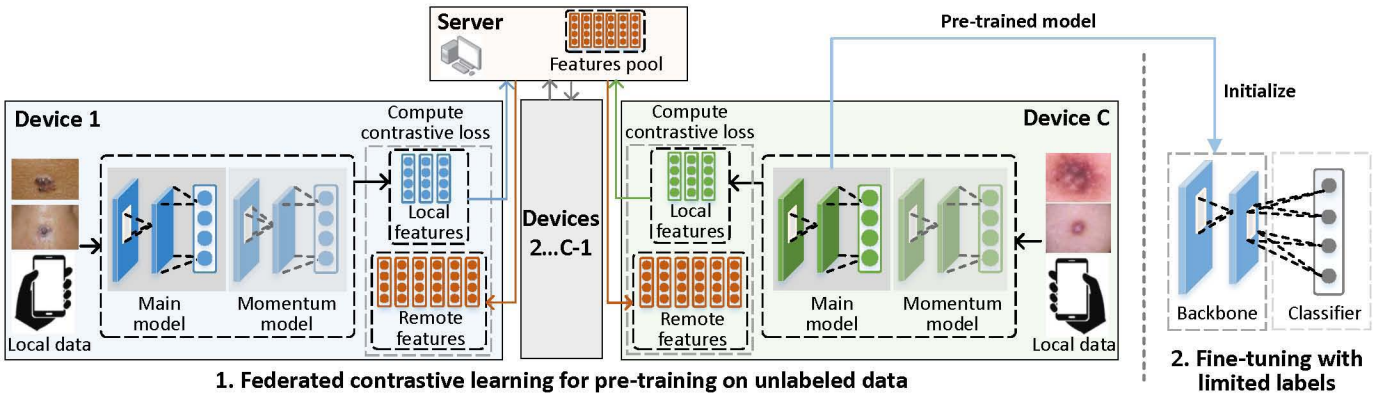

Fig. 2: Federated contrastive learning with feature sharing for pre-training the model with unlabeled data, by which good data representations are learned. After pre-training, the learned model is used as the initialization for fine-tuning with limited labels for dermatological disease classification.

typical FL algorithm FedAvg [26], learning is performed round-by-round by repeating the local learning and model aggregation process until convergence. In one round, the server activates a subset of devices and sends them the latest model. Then the activated devices perform learning on local data. More specifically, in communication round t, the server activates a subset of devices C^t and downloads the latest global model with parameters θ^t to them. Each device $c \in C^t$ learns on private dataset D_c by minimizing the local loss ℓ_c to get the updated local parameters θ_c^{t+1}. The locally updated models are aggregated into the global model by averaging the local parameters $\theta^{t+1} \leftarrow \sum_{c \in C^t} \frac{|D_c|}{\sum_{i \in C^t} |D_i|} \theta_c^{t+1}$. This learning process continues until convergence.

The problem with these FL works is that they assume the devices have ground-truth labels for all the data. However, this assumption is not realistic in dermatological disease diagnosis due to the high labeling cost and the requirement of expertise for accurate labeling. Therefore, to apply FL to dermatological disease diagnosis, one approach to effectively use limited labels to achieve high model accuracy is needed.

III. Overview of Federated Contrastive Learning

The overview of the proposed federated contrastive learning (FCL) framework is shown in Fig. 2. This framework has two stages. In the first stage, the model is collaboratively pre-trained by distributed devices on unlabeled data to extract visual representations. During learning, there is a server and many devices. The server is used to coordinate the learning process, aggregate the locally updated models and forward the shared features. The devices perform CL on local data as well as local and remote features to update local models. In the second stage, the model pre-trained in the first stage is used as the initialization for fine-tuning with limited labels. This can be achieved by existing supervised learning methods independently on each device, or collaboratively by supervised federated learning [26]. In the rest of this paper, we focus on the first stage of FCL on unlabeled data for pre-training.

To achieve better contrastive learning on each device, one can share raw data (i.e. skin images) with other devices to improve the data diversity for local learning. However, since skin images are highly sensitive and private, sharing them will cause serious privacy concerns for patients. To improve data diversity while keeping raw data local for privacy, we proposed to share features (i.e. encoded vectors). As shown in Fig. 2, FCL is performed round-by-round. In one round, each device encodes its data as local features and uploads them to the server. Local models are also uploaded for model aggregation. Then, the server de-identifies the uploaded features and sends these anonymous features to devices as remote features. The uploaded models are aggregated by averaging the weights of all models following [26], which serves as the initial model for the next round. After that, the aggregated model and the remote features are downloaded to devices. The local models on devices are updated by leveraging both local and remote features on each device, which provides more accurate and diverse contrastive information for computing the local contrastive loss. Finally, the updated local models and features are uploaded to the server to initialize the next round. This iterative learning process continues until the model convergence.

IV. Contrastive Learning with Remote and Local Features

In this section, we will present how to perform contrastive learning on each device with both local and remote features in each round of FCL, by which a good initialization for fine-tuning can be learned.

A. Local Model Architecture

The local contrastive learning process on a device is shown in Fig. 3. We use the Momentum Contrast (MoCo) model architecture [13] since it uses a separate memory bank to store negative features, which can be filled with both local and remote negatives in the FCL setting. Other popular contrastive learning models such as SimCLR [14] uses features of data in the same mini-batch as negatives, which tightly couples the negative features with the mini-batch data and is not suitable to leverage remote features in FCL. Different from SimCLR, MoCo decouples the negative features with the mini-batch data

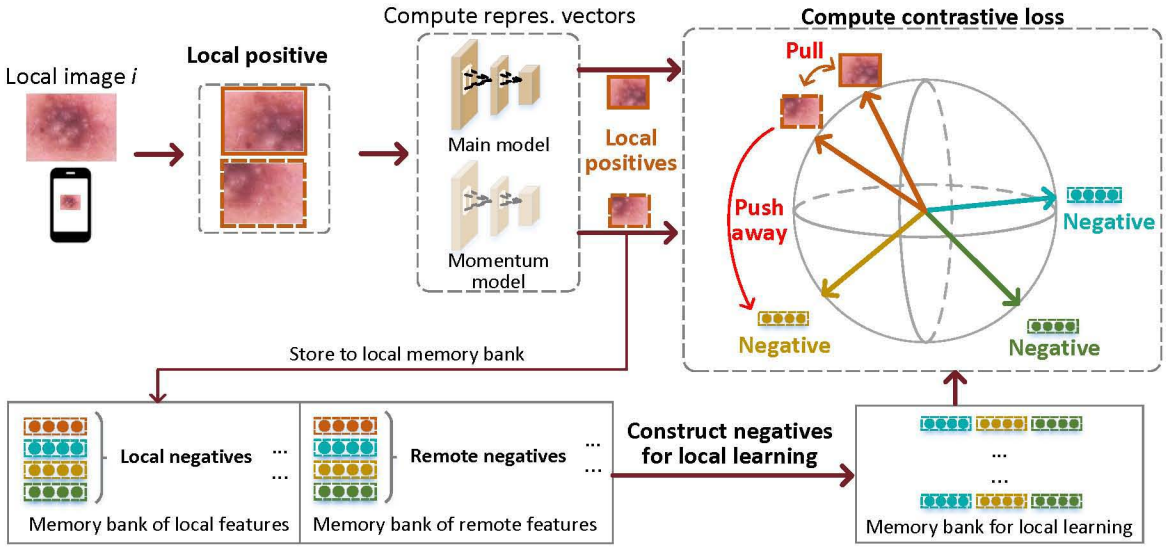

Fig. 3: Local contrastive learning with shared features on one device in FCL. The remote features from other devices serve as negatives to improve the data diversity for better-learned representations. During learning, the features of local positives are pushed close to each other and apart from the remote negatives.

by an independent memory bank and can leverage the remote features as negatives for local contrastive learning.

On each device, there are two models, the main model and the momentum model, and they have different functions. The main model is the target model for learning and will be used as the initialization for the fine-tuning in the second stage. The momentum model is used to generate features for local contrastive learning as local negatives and for sharing as remote features. The generated features will be stored in the memory bank of local features. In the learning process, the main model is updated by the contrastive loss, and the momentum model is updated as the exponential moving average of the main model with a momentum coefficient.

B. Memory Banks of Local and Remote Features

To perform contrastive learning on each device, the memory banks of local features and remote features need to be constructed such that they can provide features for computing the local loss. Each device has a memory bank of local features and a memory bank of remote features. Local features are generated by feeding local images to the momentum model to produce the feature vectors. The feature vectors are then stored in the memory bank of local features. On device i, let $Q_{l,i}$ be the memory bank of local features with K features stored. $Q_{l,i}$ is used as local negatives and maintained following a first-in-first-out principle. The oldest features in the local memory bank will be dropped when new features are generated.

The remote features are collected from other clients. At the beginning of each round of FCL, the local features are uploaded to the server and the remote features will be downloaded to fill the memory bank of remote features. More specially, the memory bank of remote features $Q_{r,i}$ on device i is filled as follows.

$$Q_{r,i} = \{Q_{l,c} \mid 1 \leq c \leq |C|, c \neq i\}. \tag{2}$$

where C is the set of all devices.

C. Learning with Remote Features

Constructing negatives from memory banks for local learning. By leveraging memory banks of local features $Q_{l,i}$ and remote features $Q_{r,i}$, the negatives $Q_{\text{CL},i}$ for computing the contrastive loss is constructed as follows. For conciseness, we leave out the device index i in $Q_{\text{CL},i}$.

On device i, at the beginning of each round of FCL, Q_{CL} is initialized as the memory bank of local features $Q_{l,i}$. During each mini-batch of local learning, a mini-batch data x of size B is fed into the momentum model to generate the features $q_{l,i,B}$. Then B remote features are sampled uniformly from remote features $Q_{r,i}$ as follows.

$$q_{r,i,B} = \{Q_{r,i,j} \mid j \sim \mathcal{U}(|Q_{r,i}|, B)\}. \tag{3}$$

where $j \sim \mathcal{U}(|Q_{r,i}|, B)$ means B indices are sampled uniformly from the range $[0, |Q_{r,i}| - 1]$ and $Q_{r,i,j}$ is the j-th feature in $Q_{r,i}$.

After learning a mini-batch, the oldest features in Q_{CL} will be replaced by the latest features q_{update}, which are constructed as follows.

$$q_{\text{update}} = \{q_{l,i,B} \cup q_{r,i,B}\}. \tag{4}$$

Removing local negatives for more accurate learning. While Q_{CL} updated by Eq.(4) contains remote features for improved data diversity, Q_{CL} also contains local features. For better local learning, we propose to completely avoid using local features as negatives during local learning. The intuition is that local features can share certain levels of similarity with the data that is being learned because they are from the same patient. Using local features as negatives can degrade the learned representations since it pushes the representations of the data being learned apart from the local features, which could have been clustered for better representations.

To solve this problem, we eliminate the use of local features during local contrastive learning. At the beginning of each round of FCL, Q_{CL} is initialized as the memory bank of remote features $Q_{r,i}$ in Eq.(2) instead of $Q_{l,i}$. The latest features for updating Q_{CL} are also simplified as follows.

$$q_{\text{update}} = \{q_{r,i,B}\}. \tag{5}$$

Compared with Eq.(4), the local negatives are removed in Eq.(5). In this way, better representations can be learned during FCL, which also results in a higher accuracy of the diagnostic model after fine-tuning.

Loss function. On device i, by using the constructed Q_{CL}, the feature q of one image being learned is compared with all features in Q_{CL}, and the contrastive loss for q is defined as follows.

$$\ell_{q,k^+,Q_{\text{CL}}} = -\log \frac{\exp(q \cdot k^+/\tau)}{\exp(q \cdot k^+/\tau) + \sum_{n \in Q_{\text{CL}}} \exp(q \cdot n/\tau)}. \tag{6}$$

where the operator $\cdot$ is the dot product between two vectors and τ is the temperature to control the distribution concentration degree [27]. By minimizing the loss, the representations of local data can be effectively learned.

V. Experiments

Datasets. The proposed methods are evaluated on four datasets of different skin colors, including the ISIC 2019 challenge dataset [30] mainly for white skins, AtlasDerm [31] and Dermnet [32] mainly for brown skins, and DarkDerm mainly for dark skins collected by us. Since these four datasets have a different number of diagnostic categories, to form a unified classification task, we use their intersection of five diseases, including basal cell carcinoma (BCC), dermatofibroma (DF), melanoma (MEL), melanocytic nevus (NV), and squamous cell carcinoma (SCC). ISIC dataset consists of about 25k dermoscopic images among nine different diagnostic categories, and we use a subset with about 21k images in the above five classes. AtlasDerm has about 11k images in 560 categories, and we use its subset in the above five classes with 618 images. Dermnet consists of images in 23 types of dermatology diseases, and we use a subset in the above five classes with 276 images. DarkDerm is established with 216 images in the above five categories. In the pre-processing, the images are resized with bi-linear interpolation such that the dimension of the shorter edge is 72 pixels while keeping the original aspect ratio.

Federated setting. We use 10 devices for FL and distribute datasets based on skin colors to simulate different patients. The ISIC dataset is randomly split into 7 partitions and each partition is distributed to one of the first 7 devices. The AtlasDerm, Dermnet, and DarkDerm datasets are assigned to one of the following three devices, respectively. On each device, the assigned dataset is randomly split into a training set and a test set with 60% and 40% data, respectively. The test set is not used in any stage of pre-training or fine-tuning. We use ResNet-18 [33] as the backbone model, and use a 2-layer MLP projection head to project the representations to 128-dimensional features.

Evaluation. We use the proposed FCL method to pre-train the model by the distributed devices without using labels. Then the pre-trained model is used as the initialization for fine-tuning with limited labels. We consider two practical settings for fine-tuning, *local fine-tuning* and *federated fine-tuning*. In local fine-tuning, each device independently fine-tunes its model with its limited labeled data after pre-training. In federated fine-tuning, devices collaboratively fine-tune the pre-trained model with limited labeled data by supervised federated learning. During fine-tuning, we evaluate with different fractions of labels, where the percentage of labeled data in the training set is $L \in \{10\%, 20\%, 40\%, 80\%\}$ on each device. Following [3], we use two metrics for evaluation, including the mean recall of each class (i.e. balanced multiclass accuracy used as the primary metric for the ISIC 2019 challenge [30]) and the mean precision of each class. We report the mean recall and mean precision on the test set of all devices.

Training details. The pre-training by the proposed FCL is performed for 100 communication rounds, and FedAvg [26] is employed as the model aggregation algorithm on the server in each round. The ratio of active devices per round is 1.0 and the number of local training epochs before each aggregation is 1. The batch size is 128 and the initial learning rate is 0.03 with a cosine decay schedule. In the fine-tuning stage, the model is trained for 20 epochs in local fine-tuning or 100 rounds in federated fine-tuning. In local fine-tuning, Adam optimizer is used with a batch size 256, a learning rate of 0.0001 with a decay factor of 0.2 at epoch 12 and 16. In federated fine-tuning, Adam optimizer is used with a batch size 128 and a learning rate of 0.0001. The training is performed on one Nvidia RTX 2080Ti GPU.

Baselines. We compare the proposed techniques with four baselines for pre-training. *Random init* uses random model initialization for fine-tuning. *Local CL* pre-trains the model by contrastive learning independently on each device with unlabeled data. *Rotation* [29] is a self-supervised learning approach for pre-training by predicting the rotation angles of images. *SimCLR* [14] is a SOTA contrastive learning based approach. We combine these two methods with the FL framework FedAvg [26] as *FedRotation* and *FedSimCLR*.

A. Local Fine-tuning

We compare the performance of different methods by local fine-tuning with limited labels. The results are shown in Table I, and both recall and precision are reported. The proposed methods effectively improve the recall and precision with different fractions of labeled data. First, with 10%, 20%, 40% and 80% labeled data on each device for fine-tuning, the proposed approaches outperform the best-performing baseline by 0.30%, 1.64%, 1.53%, 1.67% for recall, and 2.29%, 1.29%, 1.86%, 1.17% for precision, respectively. Second, the proposed approaches effectively use limited labels for fine-tuning. With 40% labels, the proposed approaches achieve 37.03% recall and 34.95% precision, which are on par with or even better than the best-performing baseline with 2× labels (37.58% recall and 33.85% precision).

TABLE I: Results of **local fine-tuning** by the proposed approaches and baselines. The model is pre-trained by different approaches without using labels and then fine-tuned with limited labeled data independently on each device. L is the label fraction on each device for fine-tuning. The recall and precision averaged over all devices are reported, and on each device, the recall and precision are averaged over all classes. With different label fractions, consistent improvements by the proposed approaches over the baselines are observed.

	L=10%		L=20%		L=40%		L=80%	
Methods	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Random init	21.56	17.35	23.13	20.79	24.88	23.69	23.92	26.06
Local CL [28]	26.57	26.54	28.82	28.20	31.46	30.49	34.27	30.71
FedRotation [29]	23.45	25.27	25.05	25.05	29.39	28.48	32.87	28.15
FedSimCLR [14]	30.11	31.25	32.77	31.67	35.50	33.09	37.58	33.85
Proposed	**30.41**	**33.54**	**34.41**	**32.96**	**37.03**	**34.95**	**39.25**	**35.02**

TABLE II: Results of **federated fine-tuning** by the proposed approaches and baselines. The model is pre-trained by different approaches without using labels and then fine-tuned with limited labeled data collaboratively by all devices. L is the label fraction for fine-tuning on each device. The recall and precision averaged over all classes are reported. With different label fractions, consistent improvements by the proposed approaches over the baselines are observed.

	L=10%		L=20%		L=40%		L=80%	
Methods	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Random init	43.15	38.97	45.63	40.41	50.73	44.66	55.61	46.35
Local CL [28]	43.41	39.59	45.69	41.39	50.35	45.58	55.47	47.70
FedRotation [29]	43.18	39.57	44.36	40.27	50.69	44.10	54.27	46.70
FedSimCLR [14]	45.89	41.44	48.92	43.39	54.17	46.71	58.64	48.27
Proposed	**48.03**	**42.87**	**51.50**	**45.71**	**55.13**	**48.73**	**59.23**	**50.21**

B. Federated Fine-tuning

We compare the performance of different methods by federated fine-tuning on all devices with limited labels. The results are shown in Table II, and both recall and precision are reported. First, the proposed methods outperform the baselines by a large margin with different fractions of labeled data. With 10%, 20%, 40% and 80% labeled data on each device for collaborative fine-tuning, the proposed approaches outperform the best-performing baseline by 2.14%, 2.58%, 0.96%, 0.59% for recall, and 1.43%, 2.32%, 2.02%, 1.94% for precision, respectively. Second, the proposed approaches effectively improve the labeling efficiency. For instance, with 10% labels, the proposed approaches achieve a similar performance as the best-performing baseline with 2× labels (48.03% vs. 48.92% for recall and 42.87% vs. 43.39% for precision).

C. Ablation Study

We perform an ablation study to evaluate the effectiveness of each of the proposed approaches. We evaluate approaches by federated fine-tuning with 10% labels, and the results are shown in Fig. 4. For example, without feature exchange, the recall is 44.72%. By enabling feature exchange, the recall is improved to 47.45%. By further removing local negatives, the recall is improved to 48.03%. This result shows that each of the proposed approaches effectively improves the learned representations and improves the recall and precision for the dermatological disease diagnosis.

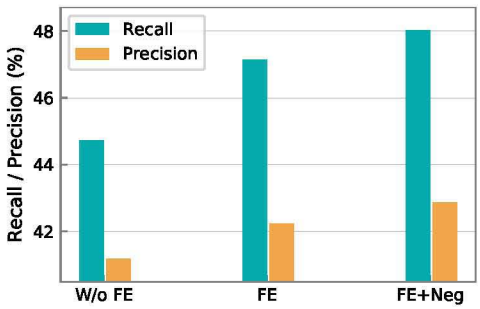

Fig. 4: Ablation study. W/o FE is the naive approach without the proposed feature exchange, and FE is the approach with feature exchange enabled. FE+Neg further removes the local negatives. Results by federated fine-tuning with 10% labeled data are reported. Each of the proposed approaches effectively improves the recall and precision.

VI. Conclusion

This work aims to enable federated contrastive learning for dermatological disease diagnosis via on-device learning. In the learning process, devices first collaboratively pre-train a model by using distributed unlabeled data and then fine-tune the model with limited labels. Feature exchange is proposed to improve the data diversity for better contrastive learning on each device. Local negatives are further removed for better clustering of learned representations. Experimental results on dermatological disease datasets of different skin colors show the effectiveness of the proposed approaches for dermatological disease diagnosis.

REFERENCES

[1] A. K. Verma, S. Pal, and S. Kumar, "Classification of skin disease using ensemble data mining techniques," *Asian Pacific journal of cancer prevention: APJCP*, vol. 20, no. 6, p. 1887, 2019.

[2] B. Ahmad, M. Usama, C.-M. Huang, K. Hwang, M. S. Hossain, and G. Muhammad, "Discriminative feature learning for skin disease classification using deep convolutional neural network," *IEEE Access*, vol. 8, pp. 39 025–39 033, 2020.

[3] Z. Wu, S. Zhao, Y. Peng, X. He, X. Zhao, K. Huang, X. Wu, W. Fan, F. Li, M. Chen *et al.*, "Studies on different cnn algorithms for face skin disease classification based on clinical images," *IEEE Access*, vol. 7, pp. 66 505–66 511, 2019.

[4] J. Velasco, C. Pascion, J. W. Alberio, J. Apuang, J. S. Cruz, M. A. Gomez, B. Molina Jr, L. Tuala, A. Thio-ac, and R. Jorda Jr, "A smartphone-based skin disease classification using mobilenet cnn," *arXiv preprint arXiv:1911.07929*, 2019.

[5] Y. Gu, Z. Ge, C. P. Bonnington, and J. Zhou, "Progressive transfer learning and adversarial domain adaptation for cross-domain skin disease classification," *IEEE journal of biomedical and health informatics*, vol. 24, no. 5, pp. 1379–1393, 2019.

[6] "Using ai to help find answers to common skin conditions," https://blog.google/technology/health/ai-dermatology-preview-io-2021/.

[7] "Free artificial intelligence (ai) dermatology search," https://www.firstderm.com/ai-dermatology/.

[8] "Your smartphone gets even smarter with dermexpert," https://www.visualdx.com/clinical-solutions/derm-expert/.

[9] Y. Wu, Z. Wang, Y. Shi, and J. Hu, "Enabling on-device cnn training by self-supervised instance filtering and error map pruning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3445–3457, 2020.

[10] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[11] X. Sun, J. Yang, M. Sun, and K. Wang, "A benchmark for automatic visual classification of clinical skin disease images," in *European Conference on Computer Vision*. Springer, 2016, pp. 206–222.

[12] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[13] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[15] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," 2020.

[16] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," *arXiv preprint arXiv:1906.05849*, 2019.

[17] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6707–6717.

[18] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" *arXiv preprint arXiv:2005.10243*, 2020.

[19] Y. Wu, Z. Wang, D. Zeng, Y. Shi, and J. Hu, "Enabling on-device self-supervised contrastive learning with selective data contrast," *arXiv preprint arXiv:2106.03796*, 2021.

[20] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, "Debiased contrastive learning," *arXiv preprint arXiv:2007.00224*, 2020.

[21] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," *arXiv preprint arXiv:2006.10029*, 2020.

[22] C.-H. Ho and N. Vasconcelos, "Contrastive learning with adversarial examples," *arXiv preprint arXiv:2010.12050*, 2020.

[23] M. Kim, J. Tack, and S. J. Hwang, "Adversarial self-supervised contrastive learning," *arXiv preprint arXiv:2006.07589*, 2020.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[25] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten, "Exploring the limits of weakly supervised pretraining," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 181–196.

[26] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[27] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[28] K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu, "Contrastive learning of global and local features for medical image segmentation with limited annotations," *arXiv preprint arXiv:2006.10511*, 2020.

[29] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[30] "Skin lesion analysis towards melanoma detection." [Online]. Available: http://challenge2019.isic-archive.com

[31] D. B. C. Samuel Freire Da Silva, "Atlasderm dermatology diseases." [Online]. Available: http://www.atlasdermatologico.com.br

[32] "Skin disease atlas." [Online]. Available: http://www.dermnet.com

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.