

Prototype-Guided and Lightweight Adapters for Inherent Interpretation and Generalisation in Federated Learning

Samuel Ofosu Mensah¹, Kerol Djoumessi¹, and Philipp Berens^{1,2}

¹ Hertie Institute for AI in Brain Health, University of Tübingen, Germany

² Tübingen AI Center, University of Tübingen, Germany

{samuel.ofosu-mensah, philipp.berens}@uni-tuebingen.de

<http://www.hertie.ai>

Abstract. Federated learning (FL) provides a promising paradigm for collaboratively training machine learning models across distributed data sources while maintaining privacy. Nevertheless, real-world FL often faces major challenges including communication overhead during the transfer of large model parameters and statistical heterogeneity, arising from non-identical independent data distributions across clients. In this work, we propose an FL framework that 1) provides inherent interpretations using prototypes, and 2) tackles statistical heterogeneity by utilising lightweight adapter modules to act as compressed surrogates of local models and guide clients to achieve generalisation despite varying client distribution. Each client locally refines its model by aligning class embeddings toward prototype representations and simultaneously adjust the lightweight adapter. Our approach replaces the need to communicate entire model weights with prototypes and lightweight adapters. This design ensures that each client’s model aligns with a globally shared structure while minimising communication load and providing inherent interpretations. Moreover, we conducted our experiments on a real-world retinal fundus image dataset, which provides clinical-site information. We demonstrate inherent interpretable capabilities and perform a classification task, which shows improvements in accuracy over baseline algorithms³.

Keywords: Federated Learning · Inherent Interpretability · Prototypes.

1 Introduction

Federated Learning (FL) has gained considerable attention due to its ability to collaboratively train machine learning models across distributed data sources while preserving privacy and reducing the need for data centralisation [13,17]. This is particularly relevant in healthcare, where patient privacy and data regulations often prevent direct data sharing between institutions [2,18,20]. FL offers

³ Code is available at <https://github.com/berenslab/FedAdapterProto>

numerous advantages in the medical domain by providing privacy-preserving collaboration among medical research institutions [18], enabling the development of generalisable models using diverse patient data while reducing regulatory challenges. However, medical imaging data presents unique challenges in FL mainly due to statistical heterogeneity across institutions arising from differences in image acquisition protocols, diverse patient populations, imbalance class distributions, and other factors [9,26,27]. As a result, clients in FL often deal with non-identically independent distributed (non-IID) data, leading to inconsistent client updates and degradation of global model performance [8,25].

Since the introduction of FedAvg [17], many FL algorithms have been proposed to address these challenges [3,11,14,15,22,23]. Nevertheless, most methods communicate and aggregate full model parameters resulting in high communication costs. To address heterogeneity and communication constraints, one approach is to communicate representations of local models such as prototypes [21] or compressed versions of the model [24]. Communicating prototypes rather than full model parameters reduces communication overhead, encourages alignment of class representations across clients, and improves tolerance to non-IID data [22]. However, prototypes do not adapt the feature extractor for each client. Also, it is important to mention that the definition of prototypes in FL varies across literature. Most studies define prototypes as the mean value of a backbone’s feature map belonging to a specific class. While these definitions improves communication, they do not provide inherent interpretations. We adopt the definition of prototypes as described in [4]. Another line of work leverages adapters – a small learnable bottleneck – to address representation misalignment caused by heterogeneity [3]. While effective, these approaches do not provide a mechanism for model interpretability. In summary, existing methods either communicate full models or specialised representations, but trade off communication efficiency, personalisation, or interpretability.

To overcome these limitations, we proposed a novel FL framework that communicates only adapter modules [3,5] and prototypes [4]. By exchanging only lightweight adapters and class-representative prototypes, our approach handles statistical heterogeneity with minimal communication overhead [3,22]. The adapters mitigate client-specific features-shift by giving each client a low-rank residual, while prototypes impose a shared interpretable structure to the logit space, helping the global model and therefore also the local models to resist label-shift and offering case-level explanations. Additionally, the use of prototypes offers inherent interpretations into the model’s decisions. To the best of our knowledge, this is the first work to integrate adapter-based fine-tuning with prototype-based learning in FL. Our key contributions are to; 1) develop an FL method that communicates only adapter parameters and prototype parameters, mitigating non-IID discrepancies across clients and 2) provide an FL method that provides inherent interpretations – a feature largely absent in prior FL methods. We carried out our experiments using a real-world retinal fundus image dataset and classified diabetic retinopathy (DR), achieving efficient communication and preserving good performance.

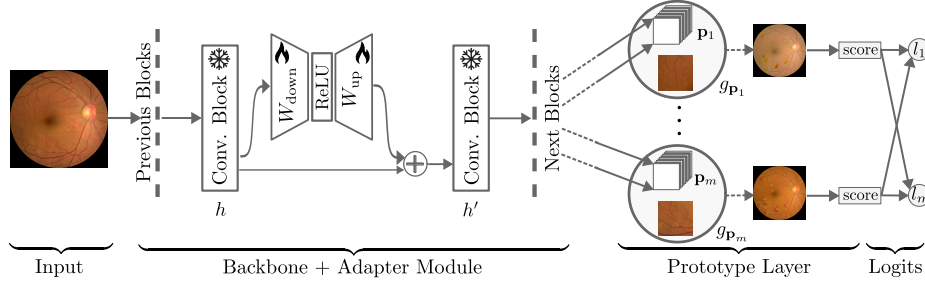


Fig. 1. A depiction of the the model, in which an input image passes through a frozen backbone augmented with adapter modules. Each adapter module compresses a convolutional block, applies a non-linear activation (for example, ReLU), and expand it back, adding the result residually to the next convolutional block. After subsequent blocks, the output representation is compared to a set of learnable prototypes. Each prototype is convolved with the feature map to compute similarities, forming a score that is used to produce the final class logits.

2 Methodology

2.1 Background

In our federated learning (FL) setup, we utilise prototypical network [4] with adapter modules as local models for the clients. We use adapters to act as compressed surrogates of the local models and guide clients to obtain generalisation regardless of varying client distribution. We attach adapter modules to different layers of the backbone model using a residual connection approach to prevent vanish gradient. The prototypes serve as learnable class-specific reference vectors that represent characteristic parts of different classes from the training dataset providing inherent interpretations and simultaneously classify the data. It is important to note that the backbone weight are always frozen and only the adapter modules and the prototype layers are optimised (Figure 1).

Adapters module. The adapter module is a lightweight bottleneck network consisting of a down-sampling convolutional layer $W_{\text{down}} \in \mathbb{R}^{H' \times W' \times D \times r}$ and an up-sampling convolutional layer $W_{\text{up}} \in \mathbb{R}^{H' \times W' \times r \times D}$, that is applied on a feature map $h \in \mathbb{R}^{H \times W \times D}$, where r represents the down-sampled dimension and D is the depth of the feature map ($r < D$). We compute an adapter module as $h' = h + \sigma(hW_{\text{down}})W_{\text{up}}$, where $\sigma(\cdot)$ is a non-linear activation function and $h' \in \mathbb{R}^{H \times W \times D}$ is the resulting feature map. For the rest of the paper, we represent W_{down} and W_{up} as α .

Prototype layer. Suppose that $\mathbf{z} \in \mathbb{R}^{H \times W \times D}$ is the output of feature extractor. The prototype layer contains m learnable prototypes $\mathbf{P} = \{\mathbf{p}_j\}_{j=1}^m$ with dimensions $\mathbf{p}_j \in \mathbb{R}^{h \times w \times D}$, where $1 \leq h \leq H, 1 \leq w \leq W$. Each prototype acts as a template matcher, computing similarity scores across all spatial

positions in $\mathbf{z}$ having the same shape as $\mathbf{p}_j$ through the squared ℓ_2 distance given as $d_j^{(h,w)} = \|\mathbf{z}^{(h,w)} - \mathbf{p}_j\|_2^2$. In our case, we compute the similarity score by negating the distance. Each prototype $\mathbf{p}_j$ constitutes a learnable parameter $W_{\text{proto}} \in \mathbb{R}^{m \times h \times w \times D}$. For the rest of the paper, we represent W_{proto} as ϕ .

Activation map and logits. The similarity computations produce activation maps that preserve spatial information, indicating where in the image prototypical parts are detected. These can be upsampled to create heatmaps showing which image regions most strongly activate each prototype. Next, logits are obtained through a weighted combination of the similarity scores. This allows the model to both classify instances and provide inherent interpretations.

2.2 Problem setting

Let there be N clients, indexed by $i \in \{1, \dots, N\}$. Each client i has local data $\mathcal{D}_i = \{(x_j^{(i)}, y_j^{(i)})\}_{j=1}^{|\mathcal{D}_i|}$ of size $|\mathcal{D}_i|$. The total data is $|\mathcal{D}| = \sum_{i=1}^N |\mathcal{D}_i|$. In standard FedAvg [17], the aim is to train a global model with parameters θ . At each communication round $r \in \{1, \dots, R\}$, the server broadcasts the current global model θ^r to each client i . Each client initialises its local model with the weights of the shared global model ($\theta_i^r \leftarrow \theta^r$) and performs local updates for a number of epochs E on local dataset $\mathcal{D}_i$. After local updates, the server collects local θ_i^r and updates the global model with θ^{r+1} obtained using a weighted average given by, $\theta^{r+1} = \sum_{i=1}^N \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \theta_i^r$. FedAvg aims to minimise

$$\min_{\theta} \sum_{i=1}^N \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \mathcal{L}_S(\mathcal{F}(\theta; x), y), \quad (1)$$

where $\mathcal{L}_S$ is a loss function evaluated on data from client i and $\mathcal{F}$ is the model parameterised by θ .

2.3 Proposed method

We modify the above procedure to address statistical heterogeneity through adapters and prototypes. For this setup, each client has a prototypical part network (ProtoPNet) with three groups of parameters. For each client i , we have ω_i representing the backbone parameters, followed by the adapter module parameters denoted as α_i , which are positioned at the end of every convolutional block of the backbone, and ϕ_i for the prototype parameters. We denote $\theta_i = (\alpha_i, \phi_i)$ as the communicating parameters from the clients to the server. We use the ResNet-50 architecture as our backbone in this work. Every ResNet model has four convolutional blocks that are made of several convolutional operations, thus our setup has four adapter modules. During training, we communicate only adapters and prototypes rather than the full model, improving communication overhead. Globally, we maintain a reference set of parameters θ^g , containing prototypes ϕ^g and adapters α^g , which are obtained via weight aggregation and sent back to the clients for update (Figure 2).

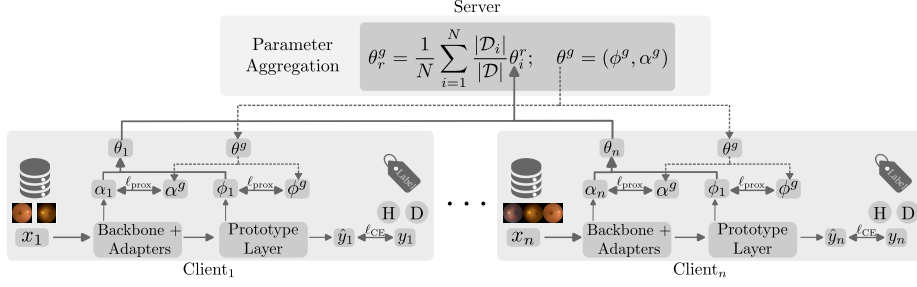


Fig. 2. Illustration of our FL setup for adapter- and prototype-based learning. Each client i receives from the server a shared global parameter set, comprising of adapter parameters and prototype parameters. Locally, the client refines the adapters and prototypes by minimising a combination of loss function and regularisers to align local parameters to the global ones. The updated parameters are sent back to the server, which performs parameter aggregation to obtain a new global model. This process is repeated for a number of rounds to enable each client to personalise adapters and prototypes for its local data distribution while collaboratively improving the global model.

Local loss function Each client i minimises a local loss composed of four components. First, we compute the cross-entropy loss defined as $\ell_{\text{CE}} = -\sum_c \mathbf{1}\{y = c\} \log(\hat{y}_c)$ between the predicted logits $\hat{y}_c$ for class c and ground-truth labels y . Second, to prevent overfitting, we apply an ℓ_2 -norm regularisation to the adapter module parameters, defined as $\ell_{\text{adapter}} = \beta \|\alpha_i\|_2^2$, where β is a weighting coefficient for the adapter regulariser. Third, we enforce prototype regularisation by encouraging each instance to move closer to its correct-class prototypes and away from wrong-class prototypes. Correct-class prototypes are representatives of the correct class being classified, while wrong-class prototypes represent instances that are not in the correct class. Specifically, we minimise the distance to correct-class prototypes ($\ell_{\text{clst}} = \sum_{j \in Y_+} \min_{(h,w)} d_j^{(h,w)}$) and maximise the distance to wrong-class prototypes ($\ell_{\text{sep}} = \sum_{j \notin Y_+} \max_{(h,w)} (-d_j^{(h,w)})$). We further regularise prototypes using an ℓ_1 -norm on the prototype-to-class weights, resulting in $\ell_{\text{proto}} = \lambda_{\text{clst}} \ell_{\text{clst}}(d) - \lambda_{\text{sep}} \ell_{\text{sep}}(d) + \gamma \|\phi_i\|_1$, with weighting coefficients λ_{clst} , λ_{sep} , and γ . Lastly, we align local adapter and prototype parameters to global references using a proximal penalty: $\ell_{\text{prox}} = \frac{\mu_1}{2} \|\alpha_i - \alpha^g\|^2 + \frac{\mu_2}{2} \|\phi_i - \phi^g\|^2$, where μ_1 and μ_2 control regularisation strength. Thus, the total local loss for each client i is $\ell_i(\theta_i; x, y) = \ell_{\text{CE}}(y, \hat{y}) + \ell_{\text{proto}}(d) + \ell_{\text{adapter}}(\alpha_i) + \ell_{\text{prox}}(\phi_i, \alpha_i; \phi^g, \alpha^g)$.

Local update procedure and global aggregation Within each communication round r , client i receives $\theta_i^r = (\phi_{r-1}^g, \alpha_{r-1}^g)$ from the server. Each client updates for a number of epochs iterating $\theta_i \leftarrow \theta_i - \eta \nabla_{\theta_i} \ell_i(\theta_i; x, y)$, where η is the learning rate. After updates, client i communicates ϕ_i and α_i back to the server. For every communication round r , we perform global update by averaging the parameters θ_i using the expression $\theta_r^g = \frac{1}{N} \sum_{i=1}^N \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \theta_i^r$.

3 Experimental details

Datasets. We conducted our experiments on retinal fundus images from Eye-PACS [7]. For this study, we used images that showed both the macula and optic disk. In addition, we selected images labelled as "*Excellent*", "*Good*", and "*Adequate*" in quality. The dataset contains clinical site information, which provided us with distribution differences for the study. From the available clinical site information, we focused on the five most frequent sites – four of which served as distinct training client data (each split 80%/20% for training and validation) and one for testing. As a result, the dataset span unique patients and varying instance of images across the selected sites. Furthermore, we selected DR as our target for the study. Specifically, the five sites contained 8,962, 7,601, 5,099, 4,080 and 3,104 images, with 76.21%, 55.27%, 80.81%, 83.82% and 81.51% of them classified as healthy, respectively. We preprocessed the images by using a tightly centred-circle cropping technique [19] and resizing them to 512×512 dimension. In addition, we applied colour jitter transformations, random flips and random rotations to further augment the data.

Implementation details. We focus on DR as a binary task by classifying images as healthy or diseased. We utilised the ResNet-50 model [10] pretrained on ImageNet [6] as the backbone and employed transfer learning approach across 100 communication rounds. Each local client was trained for one epoch before they were communicated to the server. We did this to prevent local clients from overfitting on their corresponding local data. We set the local batch size at 16 due to GPU memory constraints. We employed the Adam optimiser [12] at a learning of 0.0001 and balanced the data using a class-balanced sampler method [1]. We used accuracy as our performance metrics. All experiments were implemented using the PyTorch framework on an NVIDIA A100 GPU.

4 Results and discussion

4.1 Site-wise model performance

We presented classification accuracy results for four clients under different FL algorithms (Table 1). Notably, the FedAdapter model with proximal constraint performed best, achieving the highest average accuracy (87.58%). By contrast, the FedAdapter without proximal constraint achieved moderate accuracy levels suggesting that its inclusion during local training can significantly enhance global performance. The notably low AUC scores for FedProx and FedNova, despite their reasonable accuracy, indicate poor calibration and limited discriminative capability. On the other hand, the prototype variants showed promising performance but generally fell short of the top-performing solutions. Although not the best performing algorithm, our proposed method ranks closely behind the FedAdapter only outperforming it by 0.23% for Client 1. The results obtained by the proposed model demonstrates consistency among clients, suggesting that the proposed method can handle heterogeneous client data within a federated

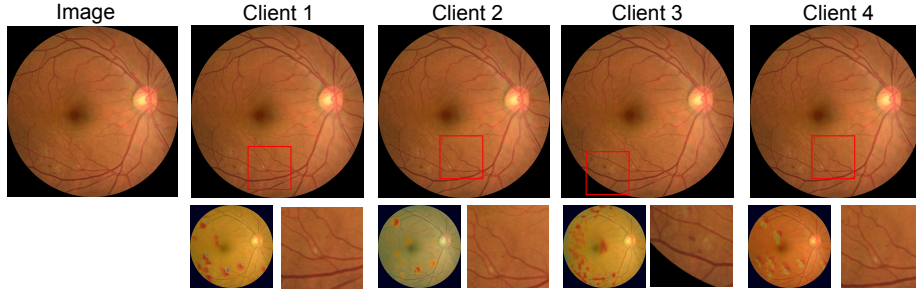


Fig. 3. Visualising key disease retinal features from the proposed method. An input image is fed to the clients to obtain local prototypes highlighting disease markers on the retinal fundus images, indicated in red bounding box in the first row, along with their corresponding activation maps in the second row. The hotter colours on the activation map denotes positive activations where as cooler colours indicate negative activations. Also, the prototype part of the image is enlarged in the second row for clarity.

setting. This design therefore underlines the importance of merging local updates with global objectives, while preserving adaptation to their own data.

4.2 Explainability

Nonetheless, the benefits of our proposed method is apparent as it achieved good performance and provided inherent interpretation to model decisions. We visualised the prototypes by drawing a red bounding box at areas of the retinal fundus images where local prototypes fires strongly. In Figure 3, we observed that all four clients placed their bounding box in almost the same area, showing a degree of agreement on what part of the image signals disease. This suggests that, even though each model was trained on a different dataset, they learned to focus on similar disease makers. We observed that most prototypes for DR detection mainly contained lesions-based features with normal or blood vessel background. These are typical features of DR. The activation maps also highlighted several hot regions on the retinal fundus image indicating possible regions of interests.

Table 1. Performance comparison of federated learning algorithms across four clients.

Algo	# Params. ($\times 10^3$)	Client 1	Client 2	Client 3	Client 4	Avg. Acc	AUC
FedAvg	25,600	78.82	66.59	65.08	67.57	69.52	75.62
FedProx	25,600	75.11	72.74	72.55	72.55	73.35	48.06
FedNova	25,600	76.82	76.92	76.41	76.51	76.66	48.77
FedAdapter *	2,530	65.57	71.61	74.82	76.06	72.02	79.18
FedAdapter	2,530	86.71	87.88	87.84	87.88	87.58	88.41
FedProto	41	59.21	60.69	61.63	61.45	60.75	62.45
FedPrototypical	1,317	70.26	74.52	74.90	76.43	74.03	77.29
Ours	3,847	86.94	87.67	86.33	86.32	86.82	86.93

* indicate FedAdapter without proximal constraint.

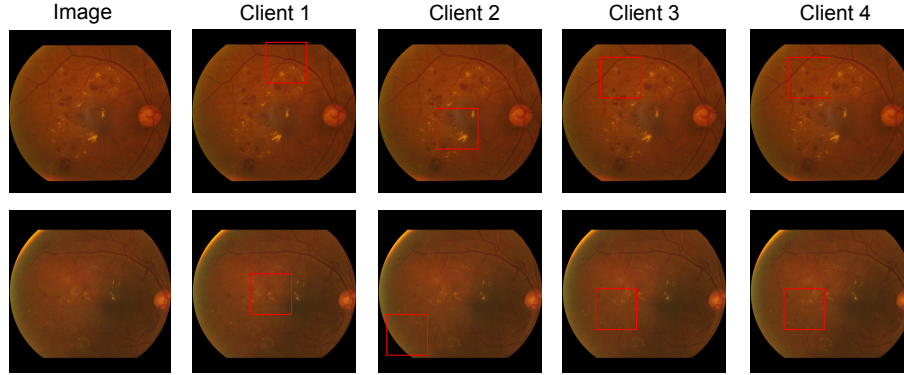


Fig. 4. Demonstrating interpretability on external dataset with two randomly selected diseased images.

4.3 Performance on external dataset

We evaluated our method on the APTOS dataset to assess its generalisability. It contained 3660 retinal fundus images with DR labels. We classified whether an image was healthy or diseased and obtained an accuracy of 96.39%, 95.38%, 93.88% and 95.57% for clients 1, 2, 3 and 4 respectively. A key observation is that Clients 3 and 4 exhibit similar activation regions (Figure 4). Their consistency could mean that they likely learned a common set of prototypical DR features that generalise well. In contrast, Clients 1 and 2 exhibit more variation in selected regions.

5 Conclusion

We presented a novel FL framework that communicates only adapter modules and prototypes to help address statistical heterogeneity and provided inherent interpretations from the model. Regardless of the differences in distribution per client, the proposed method achieved consistent high accuracy scores and maintained interpretability by highlighting disease-relevant retinal regions across clients. Moreover, the ability of our proposed method to classify DR on an external data with high confidence suggests that it has learned core features of the disease and, thus, offer a promising implication for practical purposes, especially in healthcare. While the global diagnostic performance is robust, we observe that prototype-based explanations can differ between client models. In future, integrating model heterogeneity to the proposed model could address scenarios where clients have different network structure. We will include human-validation of interpretations and study how alternative adapter architectures and prototype hyperparameters influence accuracy, interpretability, and communication cost in future work.

Moreover, our target users are resource-constrained or bandwidth-constrained nodes such as 1) community hospitals in low- and middle-income countries with

internet connectivity but with typical upstream rates of < 5 Mbps and expensive data plans, 2) on-premises Picture Archiving and Communication Systems that are permitted to exchange parameters but not pixels or 3) mobile medical screening units that connect only intermittently (e.g. nightly sync over 4G).

Acknowledgments. This project was supported by the Hertie Foundation and by the Deutsche Forschungsgemeinschaft under Germany’s Excellence Strategy with the Excellence Cluster 2064 “Machine Learning – New Perspectives for Science”, project number 390727645. PB is a member of the Else Kröner Medical Scientist Kolleg “Clin-brAIIn: Artificial Intelligence for Clinical Brain Research”.

Disclosure of Interests. The authors declare no competing interests.

References

1. Alhamoud K., Ghunaim Y., Alfarrar M., Hartvigsen T., Torr P., Ghanem B., Bibi A., Ghassemi M.: FedmedICL: Towards holistic evaluation of distribution shifts in federated medical imaging. In International Conference on Medical Image Computing and Computer-Assisted Intervention, 383–393. Cham: Springer Nature Switzerland (2024)
2. Brisimi T.S., Chen R., Mela T., Olshevsky A., Paschalidis I.C., Shi W.: Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 59–67 (2018)
3. Cai D., Wu Y., Wang S., Lin F.X., Xu M.: FedAdapter: Efficient federated learning for modern NLP. *arXiv preprint arXiv:2205.10162* (2022)
4. Chen C., Li O., Tao D., Barnett A., Rudin C., Su J.K.: This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems* (2019)
5. Chen H., Zhang Y., Krompass D., Gu J., Tresp V.: FedDAT: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, 11285–11293 (2024)
6. Deng J., Dong W., Socher R., Li L.J., Li K., Fei-Fei L.: Imagenet: A large-scale hierarchical image database. *IEEE conference on computer vision and pattern recognition*, 248–255 (2009)
7. Cuadros J., Bresnick G.: EyePACS: an adaptable telemedicine system for diabetic retinopathy screening. *Journal of diabetes science and technology* (2009)
8. Gong X., Sharma A., Karanam S., Wu Z., Chen T., Doermann D., Innanjan A.: Ensemble attention distillation for privacy-preserving federated learning. In Proceedings of the IEEE/CVF international conference on computer vision, 15076–15086 (2021)
9. Guan H., Yap P.T., Bozoki A., Liu M.: Federated learning for medical image analysis: A survey. *Pattern Recognition* (2024)
10. He K., Zhang X., Ren S., Sun J.: Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778 (2016)
11. Huang W., Ye M., Shi Z., Li H., Du B.: Rethinking federated learning with domain shift: A prototype view. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 16312–16322 (2023)

12. Kingma D.P., Ba J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Li T., Sahu A.K., Talwalkar A., Smith V.: Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 50–60 (2020)
14. Li T., Sahu A.K., Zaheer M., Sanjabi M., Talwalkar A., Smith V.: Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 429–450 (2020)
15. Liang P.P., Liu T., Ziyin L., Allen N.B., Auerbach R.P., Brent D., Salakhutdinov R., Morency L.P.: Think locally, act globally: Federated learning with local and global representations. arXiv preprint arXiv:2001.01523 (2020)
16. Loftus T.J., Ruppert M.M., Shickel B., Ozrazgat-Baslanti T., Balch J.A., Efron P.A., Upchurch Jr G.R., Rashidi P., Tignanelli C., Bian J., Bihorac A.: Federated learning for preserving data privacy in collaborative healthcare research. *Digital Health* (2022)
17. McMahan B., Moore E., Ramage D., Hampson S., y Arcas B.A.: Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282 (2017)
18. Rieke N., Hancox J., Li W., Milletari F., Roth H.R., Albarqouni S., Bakas S., Galtier M.N., Landman B.A., Maier-Hein K., Ourselin S.: The future of digital health with federated learning. *NPJ digital medicine* (2020)
19. Müller S., Heidrich H., Koch L.M., Berens P.: Fundus Circle Cropping. URL: https://github.com/berenslab/fundus_circle_cropping (2023)
20. Sadilek A., Liu L., Nguyen D., Kamruzzaman M., Serghiou S., Rader B., Ingerman A., Mellem S., Kairouz P., Nsoesie E.O., MacFarlane J.: Privacy-first health research with federated learning. *NPJ digital medicine* (2021)
21. Snell J., Swersky K., Zemel R.: Prototypical networks for few-shot learning. *Advances in neural information processing systems* (2017)
22. Tan Y., Long G., Liu L., Zhou T., Lu Q., Jiang J., Zhang C.: Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI conference on artificial intelligence*, 8432–8440 (2022)
23. Wang J., Liu Q., Liang H., Joshi G., Poor H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 7611–7623 (2020)
24. Wu C., Wu F., Lyu L., Huang Y., Xie X.: Communication-efficient federated learning via knowledge distillation. *Nature communications* (2022)
25. Wu X., Liu X., Niu J., Wang H., Tang S., Zhu G., Su H.: Decoupling general and personalized knowledge in federated learning via additive and low-rank decomposition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 7172–7181 (2024)
26. Yang T., Yu X., McKeown M.J., Wang Z.J.: When federated learning meets medical image analysis: A systematic review with challenges and solutions. *APSIPA Transactions on Signal and Information Processing* (2024)
27. Yoo J.H., Jeong H., Lee J., Chung T.M.: Open problems in medical federated learning. *International Journal of Web Information Systems*, 77–99 (2022)