FeSViBS: Federated Split Learning of Vision Transformer with Block Sampling

Faris Almalik, Naif Alkhunaizi, Ibrahim Almakky, and Karthik Nandakumar(✉)

Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
{faris.almalik,naif.alkhunaizi,ibrahim.almakky,
karthik.nandakumar}@mbzuai.ac.ae

Abstract. Data scarcity is a significant obstacle hindering the learning of powerful machine learning models in critical healthcare applications. Data-sharing mechanisms among multiple entities (e.g., hospitals) can accelerate model training and yield more accurate predictions. Recently, approaches such as Federated Learning (FL) and Split Learning (SL) have facilitated collaboration without the need to exchange private data. In this work, we propose a framework for medical imaging classification tasks called **Fe**derated **S**plit learning of **Vi**sion transformer with **B**lock **S**ampling **(FeSViBS)**. The FeSViBS framework builds upon the existing federated split vision transformer and introduces a *block sampling* module, which leverages intermediate features extracted by the Vision Transformer (ViT) at the server. This is achieved by sampling features (patch tokens) from an intermediate transformer block and distilling their information content into a pseudo class token before passing them back to the client. These pseudo class tokens serve as an effective feature augmentation strategy and enhances the generalizability of the learned model. We demonstrate the utility of our proposed method compared to other SL and FL approaches on three publicly available medical imaging datasets: HAM1000, BloodMNIST, and Fed-ISIC2019, under both IID and non-IID settings. Code: https://github.com/faresmalik/FeSViBS.

Keywords: Split learning · Federated learning · Vision transformer · Convolutional neural network · Augmentation · Sampling

1 Introduction

Vision Transformers (ViTs) are self-attention based neural networks that have achieved state-of-the-art performance on various medical imaging tasks [8,24,30]. Since ViTs are capable of encoding long range dependencies between input

F. Almalik and N. Alkhunaizi—Equal contribution

H. Greenspan et al. (Eds.): MICCAI 2023, LNCS 14221, pp. 350–360, 2023.
https://doi.org/10.1007/978-3-031-43895-0_33

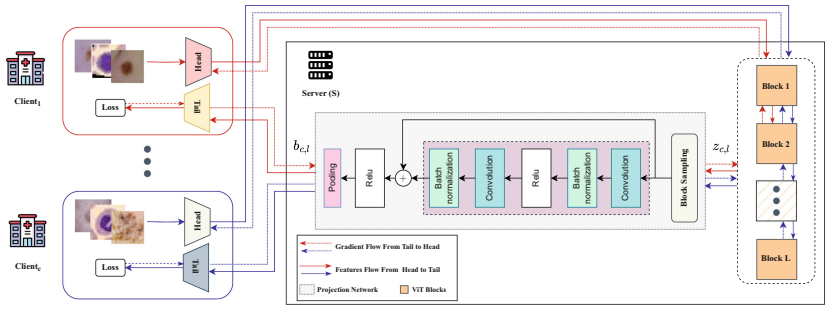

Fig. 1. FeSViBS framework. The server receives smashed representations from the clients, samples a ViT block for each client, uses a projection network to distill patch tokens into pseudo class tokens, which are sent back to the client for final prediction.

sequences [16], they are more robust against distribution shifts and are well-suited for handling heterogeneous distributions [5]. However, training ViT models typically requires significantly more data than traditional Convolutional Neural Network (CNN) models [16], which limits their application in domains such as healthcare, where data scarcity is a challenge. One way to overcome this challenge is to train such models in a collaborative and distributed manner, where large amounts of data can be leveraged from different sites without the need for sharing private data [9,11]. Federated learning and split learning are two well-known approaches for collaborative model training.

Federated Learning (FL) enables clients to collaboratively learn a global model by aggregating locally trained models [14]. Since this can be accomplished without sharing raw data, FL mitigates risks related to private data leakage. Several aggregation rules such as FedAvg [20] and FedProx [19] have been proposed for FL. However, it has been demonstrated that most FL algorithms are vulnerable to gradient inversion attacks [13], which dilute their privacy guarantees. In contrast, Split Learning (SL) divides a deep neural network into components with independently accessible parameters [10]. Since no participant in SL can access the complete model parameters, it has been claimed that SL offers better data confidentiality compared to FL. In particular, the U-shaped SL configuration, where each client has its own feature extraction head and a task-specific tail [27] can further improve client privacy, as it circumvents the need to share the data or labels. Recently, SL frameworks have been proposed for various medical applications such as tumor classification [3] and chext x-ray classification [23].

Recent studies [21,22] have demonstrated that both FL and SL can be combined to effectively train ViTs. In [22], a framework called FeSTA was proposed for medical image classification. The FeSTA framework involves a hybrid ViT architecture with U-shaped SL configuration - each client has its own CNN head and a multilayer perceptron (MLP) tail, while the shared ViT body resides on a central server. This architecture can be trained using both SL and FL in a potentially task-agnostic fashion, leading to better performance compared to

other distributed learning methods. The work in [21] focuses on privacy and incorporates differential privacy with mixed masked patches sent from the ViT on the server to the clients to prevent any potential data leakage.

In this work, we build upon the FeSTA framework [22] for collaborative learning of ViT. Despite its success, FeSTA requires pretraining the ViT body on a large dataset prior to its utilization in the SL and FL training process. In the absence of pretraining, limited training data availability (a common problem in medical imaging) leads to severe overfitting and poor generalization. Furthermore, the FeSTA framework exploits only the final *cls* token produced by the ViT body and ignores all the other intermediate features of the ViT. It is well-known that intermediate features (referred to as patch tokens) also contain discriminative information that could be useful for the classification task [4].

To overcome the above limitations, we propose a framework called **Fe**derated **S**plit learning of **Vi**sion transformer with **B**lock **S**ampling **(FeSViBS)**. Our primary novelty is the introduction of a *block sampling* module, which randomly selects an intermediate transformer block for each client in each training round, extracts intermediate features, and distills these features into a pseudo cls token using a shared projection network. The proposed approach has two key benefits: (i) it effectively leverages intermediate ViT features, which are completely ignored in FeSTA, and (ii) sampling these intermediate features from different blocks, rather than relying solely on an individual block's features or the final *cls* token, serves as a feature augmentation strategy for the network, enhancing its generalization. The contributions of this work can be summarized as follows:

i. We propose the FeSViBS framework, a novel federated and split learning framework that leverages the features learned by intermediate ViT blocks to enhance the performance of the collaborative system.
ii. We introduce block sampling at the server level, which acts as a feature augmentation strategy for better generalization.

2 Methodology

We first describe the working of a typical split vision transformer before proceeding to describe FeSViBS. Each client $c \in [1, n]$ has access to local private data $(x_c, y_c) \in \{x_c^{(i)}, y_c^{(i)}\}_{i=1}^{N_c}$, where N_c is the number of training samples available at client c, x represents the input data, and y is the class label. Following [22], we assume U-shaped split learning setting, with each client having two local networks called *head* ($\mathcal{H}_{\theta_c}$) and *tail* ($\mathcal{T}_{\psi_c}$), where θ_c and ψ_c are client-specific *head* and *tail* parameters, respectively. The server consists of a ViT *body* ($\mathcal{B}_\Phi$), which includes a stack of L transformer blocks denoted as $\mathcal{B}_{\Phi_1}, \mathcal{B}_{\Phi_2}, \cdots, \mathcal{B}_{\Phi_L}$ and $\mathcal{B}_\Phi(\cdot) = \mathcal{B}_{\Phi_L}(\cdots(\mathcal{B}_{\Phi_2}(\mathcal{B}_{\Phi_1}(\cdot))))$. Here, Φ_l represents the parameters of the l^{th} transformer block and $\Phi = [\Phi_1, \Phi_2, \cdots, \Phi_L]$ denotes the complete set of parameters of the transformer body.

During training, the client performs a forward pass of the input data through the head to produce an embedding $h_c = \mathcal{H}_{\theta_c}(x_c) \in \mathbb{R}^{768 \times M}$ of its local data,

which is typically organized as M *patch tokens* representing different patches of the input image. These embeddings (*smashed* representations) are then sent to the server. The ViT appends an additional token called the class token (*cls* $\in \mathbb{R}^{768\times 1}$) and utilizes the self-attention mechanism to obtain a representation $b_c = \mathcal{B}_\Phi(h_c) \in \mathbb{R}^{768\times 1}$, which is typically the *cls* token resulting from the last transformer block. This *cls* token is returned to the client for further processing. The *tail* at each client projects the received class token representation b_c into a class probability distribution to get the final prediction $\hat{y}_c = \mathcal{T}_{\psi_c}(b_c)$. This marks the end of the forward pass. Subsequently, the backpropagation starts with computing loss $\ell_c(y_c, \hat{y}_c)$, where $\ell_c(.)$ represents the client's loss function between the true labels y_c and predicted labels $\hat{y}_c$. The gradient of this loss is propagated back in the reverse order from the client's *tail*, server's *body*, to the client's *head*. We refer to this setting as Split Learning of Vision Transformer (**SLViT**), where each client optimizes the following objective in each round:

$$\min_{\theta_c,\Phi,\psi_c} \frac{1}{N_c} \sum_{i=1}^{N_c} \ell_c\big(y_c^{(i)}, \mathcal{T}_{\psi_c}(\mathcal{B}_\Phi(\mathcal{H}_{\theta_c}(x_c^{(i)})))\big) \tag{1}$$

In FeSTA [22], an additional federation step was introduced. After every few SL rounds, the local (client-specific) *heads* and *tails* are aggregated in a *unifying round* using FedAvg [20] to produce global parameters $\bar{\theta}$ and $\bar{\psi}$. Note that the above framework completely ignores all the intermediate features (*patch* tokens) extracted from various ViT blocks. In [4], it was demonstrated that these patch tokens are also discriminative and valuable for classification tasks. Hence, we aim to exploit these intermediate features to further enhance the performance.

2.1 FeSViBS Framework

The proposed FeSViBS method is illustrated in Fig. 1 and detailed in Algorithm 1. The working of the FeSViBS framework is very similar to FeSTA, except for one key difference. During the forward pass of SLViT and FeSTA, the server always returns the *cls* token from the last ViT block. In contrast, a FeSViBS server samples an intermediate block $l \in \{1, 2\ldots, L\}$ for each client c in each round and extracts the intermediate features $z_{c,l}$ from the chosen l^{th} block as follows:

$$z_{c,l} = \mathcal{B}_{\Phi_l}\Big(\mathcal{B}_{\Phi_{l-1}} \ldots \mathcal{B}_{\Phi_1}\big(\mathcal{H}_{\theta_c}(x_c)\big)\Big) \tag{2}$$

where $z_{c,l} \in \mathbb{R}^{768\times M}$. The server then projects the extracted intermediate features into a lower dimension using a *projection network* $\mathcal{R}$ (shared across all blocks) to obtain the final representation $b_{c,l} = \mathcal{R}_\pi(z_{c,l})$, where $b_{c,l} \in \mathbb{R}^{768\times 1}$. This final representation $b_{c,l}$ can be considered as a *pseudo class token* and the role of the projection network is to distill the discriminative information contained in the intermediate features into this pseudo class token. The primary motivation for block sampling is to effectively leverage intermediate ViT features that are better at capturing local texture information (but are lost when

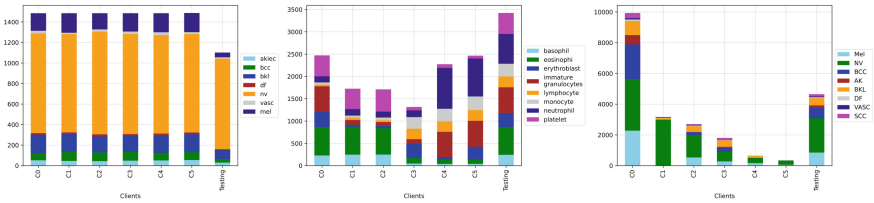

Fig. 2. Distribution of: (**left**) HAM10000, (**middle**) BloodMNIST, and (**right**) FedISIC2019. Each stacked bar represents the number of samples, and each color represents each class. The last bar in each figure represents the testing set.

only the final *cls* token is used). Stochasticity in the block selection serves as a feature augmentation strategy, thereby aiding the generalization performance.

The architecture of the projection network is shown in Fig. 1 and it resembles a simple ResNet [12] block with skip connection. The pseudo class token is then sent to the client's *tail* to obtain the final prediction $\hat{y}_c = \mathcal{T}_{\psi_c}(b_{c,l})$ and complete the forward pass. Each client uses $\hat{y}_c$ along with the true labels y_c to compute the loss $\ell_c(y_c, \hat{y}_c)$. The gradients of the client's *tail* are then calculated and sent back to the server, which then carries out the back-propagation through the *projection network* and relevant blocks of the ViT *body* (only those blocks involved in the corresponding forward pass). Next, the server sends the gradients back to the client to propagate them through the *head* and end the back-propagation step. Hence, the client's optimization problem is:

$$\min_{\theta_c, \Phi_{1:l,c}, \pi, \psi_c} \frac{1}{N_c} \sum_{i=1}^{N_c} \ell_c\big(y_c^{(i)}, \mathcal{T}_{\psi_c}(b_{c,l}^{(i)})\big). \tag{3}$$

In the FeSViBS framework, the *heads* and *tails* of all the clients are assumed to have the same network architecture. Within each collaboration round, all the clients perform the forward and backward passes. While the parameters of the relevant head and tail as well as the shared projection network are updated after every backward pass, the parameters of the ViT body are updated only at the end of a collaboration round after aggregating updates from all the clients. The above protocol until this step is referred to as **SViBS**, because there is still no federation of the *heads* and *tails*. Similar to FeSTA, we also perform aggregation of the local *heads* and *tails* periodically in unifying rounds, resulting in the final FeSViBS framework. While in SViBS, the clients can initialize their heads and tails independently, FeSviBS requires a common initialization by the server and sharing of aggregated head and tail parameters after a unifying round.

3 Experimental Setup

Datasets. We conduct our experiments on three medical imaging datasets. The first dataset is HAM10000 [26], a multi-class dataset comprising of 10,015 dermoscopic images from diverse populations. HAM10000 includes 7 imbalanced

Algorithm 1. FeSViBS

Require: Local data at client c (x_c, y_c). Server initializes the body parameters (Φ), Projection Network parameters (π), client head and tail parameters $(\bar{\theta}, \bar{\psi})$

for rounds $r = 1, 2, \ldots, R$ **do**
  for client $c \in [1, n]$ **do**
    if $r = 1$ **or** $(r-1) \in$ Unifying Rounds **then**
      $(\theta_c, \psi_c) \leftarrow (\bar{\theta}, \bar{\psi})$
    end if
    Client c: $h_c \leftarrow \mathcal{H}_{\theta_c}(x_c)$
    Server:
      Sample a ViT block (l) for client c
      $b_{c,l} \leftarrow \mathcal{R}_\pi\Big(\mathcal{B}_{\Phi_l}\big(\mathcal{B}_{\Phi_{l-1}} \ldots \mathcal{B}_{\Phi_1}(h_c)\big)\Big)$
    Client c:
      Compute $\ell_c\big(y_c, \mathcal{T}_{\psi_c}(b_{c,l})\big)$ and Backprop.
      Update (θ_c, ψ_c) with suitable optimizer
    Server:
      Update (π) with suitable optimizer, Compute and store $\Phi_{1:l,c}$
  end for
  Server:
    Update body: $\Phi \leftarrow \frac{1}{n}\sum_c \Phi_{1:l,c}$
  if $r \in$ Unifying Rounds **then**
    $(\bar{\theta}, \bar{\psi}) \leftarrow (\frac{1}{n}\sum_c \theta_c, \frac{1}{n}\sum_c \psi_c)$
  end if
end for

categories of pigmented lesions; we randomly perform 80%/20% split for training and testing, respectively. The second dataset [2] termed "BloodMNIST" is a multi-class dataset consisting of 17,092 blood cell images for 8 different imbalanced cell types. We followed [29] and split the dataset into 70% training, 10% validation, and 20% testing. Finally, the Fed-ISIC2019 dataset consists of 23,247 dermoscopy images for 8 different melanoma classes. This dataset was prepared by FLamby [25] from the original ISIC2019 dataset [6,7,26] and the data was collected from 6 centers, with significant differences in population characteristics and acquisition systems, representing real-world domain shifts. We use 80%/20% split for training and testing, respectively. The training samples in all datasets are divided among 6 clients, whereas the testing set is shared among them all. The distribution of each dataset is depicted in Fig. 2. Note that Fed-ISIC2019 and BloodMNIST are non-IID, whereas HAM10000 is IID.

Server's Network. For the server's body, we chose the ViT-B/16 model from timm library [28] which includes $L = 12$ transformer blocks, embedding dimension $D = 768$, 12 attention heads, and divides the input image into patches each of size 16×16 with $M = 196$ patches. We limit the block sampling to the first 6 ViT blocks. Additionally, the projection network has two convolution layers with a skip connection, which takes an input of dimension 768×196 and projects it into a lower dimension of 768.

Table 1. Average balanced accuracy for different methods. Centralized, FedAvg, FedProx, SCAFFOLD, MOON, and FeSTA have one global unified model for all clients. For local, SLViT, and SViBS, we report the standard deviation (stdev) across clients. For FeSViBS, we report stdev over stochastic sampling of ViT blocks during inference.

	Dataset		
	HAM10000	BloodMNIST	Fed-ISIC2019
Centralized	0.615	0.957	0.614
Local	0.494 ± 0.024	0.785 ± 0.017	0.290 ± 0.113
SLViT	0.540 ± 0.029	0.826 ± 0.018	0.293 ± 0.133
SViBS (ours)	0.570 ± 0.011	0.836 ± 0.014	0.330 ± 0.042
FedAvg [20]	0.564	0.894	0.476
FedProx [19]	0.568	0.892	0.472
SCAFFOLD [15]	0.290	0.880	0.330
MOON [18]	0.570	0.903	0.450
FeSTA [22]	0.638	0.929	0.430
FeSViBS (ours)	**0.682 ± 0.021**	**0.936 ± 0.002**	**0.534 ± 0.005**

Clients' Networks. Each client has two main networks: head and tail. We followed timm library's implementation of Hybrid ViTs (h-ViT) to design each client's head, which is a ResNet-50 [12] with a convolution layer added to project the features extracted by ResNet-50 to a dimension of 768×196. The tail is a linear classifier. Also, we unify the clients' networks (head and tail) every 2 rounds using FedAvg. We conduct our experiments for 200 rounds with Adam optimizer [17], a learning rate of 1×10^{-4}, and 32 batch size with a cross-entropy loss calculated at the tail. The code was implemented using PyTorch 1.10 and the models were trained using Nvidia A100 GPU with 40 GB memory.

4 Results and Analysis

Following [25], we used balanced accuracy in all experiments to evaluate the performance of the classification task across all datasets. This metric defines as the average recall on each class. In Table 1, we compare the performance of FeSViBS and SViBS frameworks with other SOTA methods. FeSViBS consistently outperforms other methods on the three datasets with both IID and non-IID settings. More specifically, for HAM10000 (IID), FeSViBS outperforms all other methods with a **4.4%** gain in performance over FeSTA and approximately **11%** over FedAvg and FedProx ($\mu = 0.006$). In the non-IID settings with both BloodMNIST and Fed-ISIC2019, FeSViBS maintains a high performance compared to other methods. Under extreme non-IID settings (Fed-ISIC2019), our approach demonstrated a performance improvement of **10.4%** compared to FeSTA and **5.8%** over FedAvg and FedProx, demonstrating the robustness of FeSViBS.

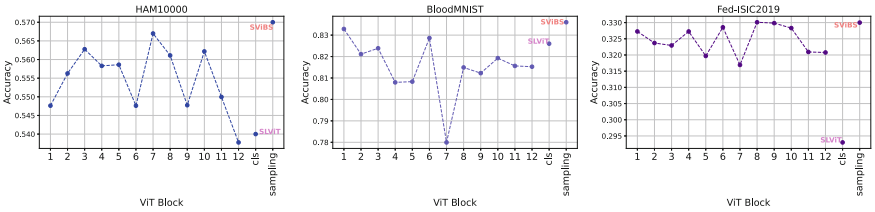

Fig. 3. Performance of each ViT block, sending *cls* token (SLViT), and SViBS. Sampling from blocks 1 to 6 (SViBS) showed better performance than individual blocks.

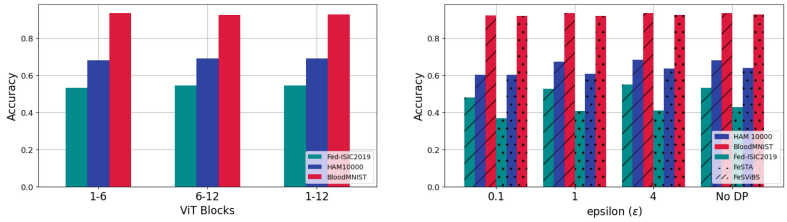

Fig. 4. FeSViBS performance with: **Left** different set of ViT blocks. **Right**: Differential Privacy with different ϵ values along with the original FeSViBS.

We investigate the impact of sampling intermediate blocks in SViBS, by analysing the individual performance of intermediate features from specific blocks during training. The results in Fig. 3 demonstrate that the majority of individual blocks outperform the vanilla split learning setting (SLViT), which is dependent on the *cls* token. On the other hand, SViBS shows dominant performance across datasets, where the sampling of ViT blocks provides augmented representations of the input images at different rounds and improves the generalizability. From Table 1, we also observe that the variance of the accuracy achieved by FeSViBS due to stochastic block sampling during inference is very low.

5 Ablation Study

Set of ViT Blocks. To study the impact of ViT blocks from which the intermediate features are sampled on the overall performance of FeSViBS, we carry out experiments choosing different sets of blocks. The results depicted in Fig. 4 (left) show consistent performance for different sets of blocks across different datasets. This indicates that implementing FeSViBS with the first 6 ViT blocks would reduce the computational cost without compromising performance.

FeSViBS with Differential Privacy. Differential Privacy (DP) [1] is a widely-used approach that aims to improve the privacy of local client's data by adding noise. We conduct experiments where we add Gaussian noise to the client's

head output (h_c). In such a scenario, DP makes it more challenging for a malicious/curious server to infer the client's input from the smashed representations. With different ϵ values, the results in Fig. 4 (right) show that FeSViBS maintains its performance even under a small ϵ value ($\epsilon = 0.1$), while also outperforming FeSTA under the same constraints.

Number of Unifying Rounds. We investigated the impact of reducing communication rounds (unifying rounds) on FeSViBS performance. However, our results showed that performance was maintained even with decreasing the number of communication rounds.

Computational and Communication Overhead. Except for MOON and SCAFFOLD, all methods in Table 1 share the same h-ViT architecture, resulting in similar computational costs. SViBS and FeSViBS require training an additional projection network but avoid needing a complete ViT forward/backward pass. Centralized and local training methods have no communication cost. For other methods, the communication cost per client per collaboration round: **(i) FedAvg/FedProx:** $\sim$ 97M, **(ii) SLViT/SViBS:** $\sim$ 197M values for HAM10000 dataset, and **(iii) FeSTA/FeSViBS:** $\sim$ 197M values +12M parameters per client per unifying round. Thus, the proposed method has a marginally higher communication overload than SL and twice the communication burden as FL.

6 Conclusion and Future Directions

We proposed a novel Federated Split Learning of Vision Transformer with Block Sampling (FeSViBS), which utilizes FL, SL and sampling of ViT blocks to enhance the performance of the collaborative system. We evaluate FeSViBS framework under IID and non-IID settings on three real-world medical imaging datasets and demonstrate consistent performance. In the future, we aim to (i) extend our work and evaluate the privacy of FeSViBS under the presence of malicious clients/server, (ii) evaluate FeSViBS in the context of natural images and (iii) extend the current framework to multi-task settings.

References

1. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)
2. Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., Rodellar, J.: A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. Data Brief **30**, 1–5 (2020)
3. Ads, O.S., Alfares, M.M., Salem, M.A.M.: Multi-limb split learning for tumor classification on vertically distributed data. In: 2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 88–92. IEEE (2021)

4. Almalik, F., Yaqub, M., Nandakumar, K.: Self-ensembling vision transformer (SEViT) for robust medical image classification. In: Wang, L., Dou, Q., Fletcher, P.T., Speidel, S., Li, S. (eds.) Medical Image Computing and Computer Assisted Intervention. MICCAI 2022. LNCS, vol. 13433, pp. 376–386. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16437-8_36
5. Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., Veit, A.: Understanding robustness of transformers for image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10231–10241 (2021)
6. Codella, N.C.F., et al.: Skin lesion analysis toward melanoma detection: a challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC). In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 168–172 (2018). https://doi.org/10.1109/ISBI.2018.8363547
7. Combalia, M., et al.: Bcn20000: Dermoscopic lesions in the wild. arXiv:1908.02288 (2019)
8. Dai, Y., Gao, Y., Liu, F.: TransMed: transformers advance multi-modal medical image classification. Diagnostics **11**(8), 1384 (2021)
9. Dayan, I., et al.: Federated learning for predicting clinical outcomes in patients with COVID-19. Nat. Med. **27**(10), 1735–1743 (2021)
10. Gupta, O., Raskar, R.: Distributed learning of deep neural network over multiple agents. J. Netw. Comput. Appl. **116**, 1–8 (2018)
11. Ha, Y.J., Lee, G., Yoo, M., Jung, S., Yoo, S., Kim, J.: Feasibility study of multi-site split learning for privacy-preserving medical systems under data imbalance constraints in COVID-19, x-ray, and cholesterol dataset. Sci. Rep. **12**(1), 1534 (2022)
12. 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, pp. 770–778 (2016)
13. Huang, Y., Gupta, S., Song, Z., Li, K., Arora, S.: Evaluating gradient inversion attacks and defenses in federated learning. Adv. Neural. Inf. Process. Syst. **34**, 7232–7241 (2021)
14. Kairouz, P., et al.: Advances and open problems in federated learning. Found. Trends ® Mach. Learn. **14**(1–2), 1–210 (2021)
15. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: stochastic controlled averaging for federated learning. In: International Conference on Machine Learning, pp. 5132–5143. PMLR (2020)
16. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: a survey. ACM Comput. Surv. (CSUR) **54**, 1–41 (2021)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10713–10722 (2021)
19. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proc. Mach. Learn. Syst. **2**, 429–450 (2020)
20. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)

21. Oh, S., et al.: Differentially private cutmix for split learning with vision transformer. In: First Workshop on Interpolation Regularizers and Beyond at NeurIPS 2022 (2022)
22. Park, S., Kim, G., Kim, J., Kim, B., Ye, J.: Federated split vision transformer for COVID-19 CXR diagnosis using task-agnostic training. In: 35th Conference on Neural Information Processing Systems, NeurIPS 2021, pp. 24617–24630 (2021)
23. Poirot, M.G., Vepakomma, P., Chang, K., Kalpathy-Cramer, J., Gupta, R., Raskar, R.: Split learning for collaborative deep learning in healthcare (2019). https://doi.org/10.48550/ARXIV.1912.12115, https://arxiv.org/abs/1912.12115
24. Shamshad, F., et al.: Transformers in medical imaging: a survey. arXiv preprint arXiv:2201.09873 (2022)
25. du Terrail, J.O., et al.: FLamby: datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. In: Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2022). https://openreview.net/forum?id=GgM5DiAb6A2
26. Tschandl, P., Rosendahl, C., Kittler, H.: The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci. Data **5**(11), 180161 (2018). https://doi.org/10.1038/sdata.2018.161
27. Vepakomma, P., Gupta, O., Swedish, T., Raskar, R.: Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564 (2018)
28. Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019). https://doi.org/10.5281/zenodo.4414861
29. Yang, J., Shi, R., Ni, B.: MedMNIST classification decathlon: a lightweight AutoML benchmark for medical image analysis. In: IEEE 18th International Symposium on Biomedical Imaging (ISBI), pp. 191–195 (2021)
30. Zheng, S., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6881–6890 (2021)