A feasibility study for federated learning in the context of classification of biomarkers from optical coherence tomography (OCT) images

Arunodhayan Sampathkumar and Danny Kowerko

Junior Professorship of Media Computing,
Chemnitz University of Technology,
09107 Chemnitz, Germany,
{firstname.lastname}@informatik.tu-chemnitz.de

Abstract.: Recent advancement in machine learning and deep learning requires centralized data for training. Federated Learning (FL) is a machine learning approach that deals with collaboratively training a model while keeping the training data decentralized. The introduction of FL reduces the privacy risk (data sharing) and the cost of memory usage from the traditional centralized approach. We developed a scalable baseline FL framework based on PyTorch, incorporated in a docker container. We distributed our training data equally to our client servers and deployed the docker container to train our models. This research paper focuses on creating a baseline FL workflow for OCT biomarkers classification to lower the risk inherent to centralized medical data. The best prediction accuracy (macro average F1-score) obtained from the FL approach (72.5%) is closer when compared to our centralized approach (73.6%).

Key words: Federated Learning, Deep Learning, OCT Biomarkers, Cryptography, Computer Vision

1 Introduction

Federated Learning (FL) is an emerging technology for decentralized learning of machine learning models in a network of remote devices. In Germany for instance, "it is not possible to pool routine data from different hospitals for research purposes without the consent of the patients" [LB20].

Federated Learning (FL)- FL learns a lot from raw data instead of proxy data. The principle of FL incorporates minimizing data collection, reducing privacy risks, e.g. when handling medical data, and reducing the cost when compared with traditional centralized machine learning which requires data centers to store the dataset [KMA+19]. In this article, we design a workflow of a scalable baseline FL framework to train a medical image image-based classifier for ophthalmic biomarkers using decentralized data to reduce patient privacy risk.

Dataset- The dataset used in this research is high-resolution images of the central retina from Optical Coherence Tomography (OCT) [PKL+22]. OCT helps in diagnosing and monitoring Age-related Macular Degeneration (AMD), Diabetic Macular Edema (DME) and Retinal Vein Occlusion (RVO) [NSSK16], [SOS+19]. The 16 OCT biomarkers used in this classification process are listed in Table 1. The detailed description of data preprocessing is discussed in section 3.2.

No.	Abbreviation	Full Biomarker Name
1	FAVF	Fully Attached Vitreous Face
2	IRHRF	IntraRetinal Hyperreflective Foci
3	IRF	IntraRetinal Fluid
4	DRT/ME	Diffuse Retinal Thickening or Macular Edema
5	PAVF	Partially Attached Vitreous Face
6	VB	Vitreous Debris
7	PTH	Preretinal Tissue Hemorrhage
8	EZ	Ellipsoid Zone
9	IRH	Intra Retinal Hemorrhages
10	SRF	SubRetinal Fluid
11	ATRL	Atrophy Thinning of Retinal Layers
12	SHRM	Subretinal HyperReflective Material
13	DRIL	Disruption of the Retinal Inner Layers
14	VMT	Vitreo Mascular Traction
15	RPE	Retinal Pigment Epithelium
16	PED	Pigment Epithelial Detachment

Table 1: List of OCT biomarkers and abbreviations used in this study

Privacy- Compared to centralized data, training in FL has a distinct privacy advantage. Even holding medical data without the patient's information can still put the patient's privacy at risk. The information transmitted from FL clients to the FL server holds very minimal information updates such as weights of the trained model, train log files, hyperparameter settings, etc. when compared to raw data available in centralized data centers. The updates from FL client systems are encrypted using cryptography and can be transmitted to the FL servers in a secured environment [EL01]. The detailed description of the cryptography method used in designing the scalable FL framework is discussed in section 3.3.

Some general notation and definitions are defined in this research paper:

- **FL** - Federated Learning
- **OCT** - Optical Coherence Tomography
- **CUDA** - Computer Unified Device Architecture
- **cuDNN** - NVIDIA CUDA Deep Neural Network

- **AMD** - Age-related Macular Degeneration
- **DME** - Diabetic Macular Degeneration
- **RVO** - Retinal Vein Occulsion
- **ELM** - External Limiting Membrane
- **PC-1** - Client system-1
- **PC-2** - Client system-2
- **PC-3** - Server system
- **AES** - Advanced Encryption Standards
- **DES** - Data Encryption Standards
- **3DES** - Triple Data Encryption Standards
- **HMAC** - Keyed-Hashing for Message Authentication
- **SHA-256** - Secure Hashing Algorithm, 256-Bits
- **HDD** - Hard Disk Drive

2 Related Work

The number of research and development in federated learning has increased rapidly during the past few years [GDG+17]. The goal of research communities is to develop, analyze, and learn from distributed data without exploiting user data privacy. Researchers from Google have performed a federated learning approach on Gboard (Google Keyboard) to predict emoji's [RMRB19], actions to be made by the users [YAE+18] and discovering of new words [CMOB19] on mobile applications. Apple researchers enhance their Siri recognition by training different copies of a speaker recognition model across all its user's devices, using only the audio data available locally. It then sends just the updated models back to a central server to be combined into a master model. In this way, raw audio of users' Siri requests never leaves their iPhones and iPads, but the assistant continuously gets better at identifying the right speaker [GSvD+20]. Large-scale machine learning, particularly from data centers, has motivated the development of distributed optimization methods, namely Federated Learning(FL) [HYF+18]. Federated learning works without the need to store massive datasets in a centralized cloud, hence reducing data privacy and storage costs [SFM+16]. FL was designed to enhance the intelligence of user interaction on mobile devices by providing a decentralized computing strategy to train a neural network model [MMRyA16]. Mobile devices have referred a client, which generates large volumes of raw user data, were trained locally and shared the updated model to the server, group of these weighted models was aggregated to create a global model. Hence, the created global model can be used as pre-trained weights to train on other client mobile devices [SS15]. The models from the client devices don't share meta information, hence avoiding data privacy. Cryptography techniques were introduced to have secure communication between the client and the server [AS00], [VYJ08].

3 Proposed Method

The proposed method comprises 5 subsections starting with the experimental setup followed by data preprocessing, model architecture, symmetric cryptography, and implementation.

3.1 Experimental Setup

This section describes the hardware and software configurations. The FL setup comprises 3 different local clients and servers, namely PC-1, PC-2, and PC-3, along with the local cloud setup to share the trained model and the corresponding files as presented in Figure 2. Table 2 describes the hardware setup.

	PC-1	**PC-2**	**PC-3**
CPU	i9-9900K@ 3.60GHz	i9-9900K @ 3.60GHz	i7-7700 @ 3.60GHz
RAM	126 GB	126 GB	32 GB
GPU	Titan RTX (24GB)	Geforce RTX 3060 (12GB)	Geforce GTX 1050Ti (4GB)
Hard-disk type	SSD	SSD	HDD
Operating System	Ubuntu 20.04 LTS	Ubuntu 20.04 LTS	Ubuntu 20.04 LTS
No of cores	16	16	8
Location			

Table 2: Hardware configurations for the FL setup

The local cloud setup was accomplished by connecting our HDD to Nextcloud which acts as our local cloud to communicate and transmit data between our local clients and server. All the data are encrypted symmetrically. A detailed description of cryptography-based symmetric encryption is discussed in section 3.3.

The Software configuration is listed below:-

1. *Programming Language:*
   - Python 3.8
2. *Libraries:*
   - CUDA 11.3 tool kit for GPU acceleration

- Cudnn 7.5.4
- Pytorch-1.7.0
- Torchvision-0.7.0
- Docker

3. *Dependencies:*
   - numpy 1.14.2
   - scipy 1.0.0
   - python-openCV
   - matplotlib 2.2.0
   - libblas-dev liblapack-dev
   - cmake 3.5.1
   - cython 0.29

3.2 Dataset Preprocessing

Table 3 presents a detailed description of the dataset. There are 6424 OCT slices where 16 OCT biomarkers (classes) are documented [MKY+22]. Since different OCT slices possess different image resolutions, the OCT slices were scaled to a size of 224×224 pixels. Figure 1 represents biomarkers on the OCT slice.

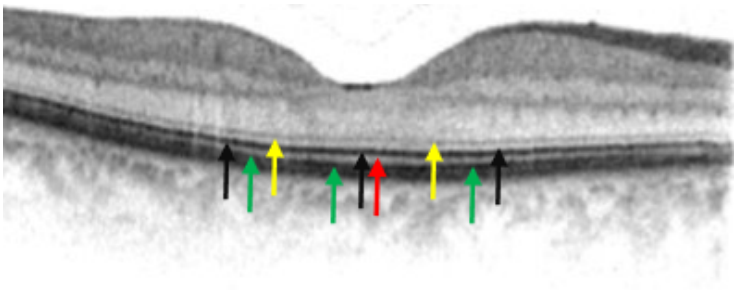

Fig. 1: Subfoveal elevation and attenuation of EZ (black arrows), subretinal (green arrows), clinical pigmentary changes (red arrow), ELM (yellow arrow)

3.3 Symmetric Cryptography

Cryptography is the art of transforming readable text into unreadable text (cipher text) to ensure data privacy. There are two types of cryptography, symmetric key cryptography (secret key) and asymmetric key cryptography (public key). This research paper deals with the symmetric cryptography method, where a common key (secret key) was shared between the participants for both encryption and decryption purposes. There are several

OCT biomarkers	No of Images	Split 1	Split 2
FAVF	951	476	475
IRHRF	937	468	469
IRF	904	452	452
DRT/ME	843	422	421
PAVF	658	329	329
VB	657	328	329
PTH	502	251	251
EZ	338	169	169
IRH	242	121	121
SRF	152	76	76
ATRL	119	59	60
SHRM	65	33	32
DRIL	26	13	13
VMT	10	5	5
RPE	10	5	5
PED	10	5	5

Table 3: Ophthalmic dataset overview of 16 OCT biomarkers. The blue color indicates the dataset split used for PC-1, and the brown color indicates the dataset split used for PC-2

algorithms for symmetric key cryptography namely AES, DES, 3DES, etc. A Python-based library named Fernet was used which incorporates the AES algorithm [CBPA14]. Fernet guarantees that a message encrypted using the technique cannot be manipulated or read without the secret key.

Important features of the Fernet module are:

- secured mechanism for generating keys
- secured encryption algorithm - AES with CBS mode and PKCS7 padding
- randomly allocating secured salt values
- signing a message using HMAC and SHA256 to detect any attempts to change it

3.4 Model Architecture

The model architecture used here were EfficientNet-B0, EfficientNet-B4, EfficientNet-B5, EfficientNet-v2s, Tf-Mixnet-s [TL19b], and Seresnext50_32x4d [HSS17]. The EfficientNet model architectures are described in Table 4. The EfficientNet network is based on the inverted bottleneck residual blocks of MobileNetV2, and Seresnext50_32x4d is similar to Resnet50, where each present block is replaced by Seresnext block, which consists of a Fully Connected layer (FC) as discussed in Tables 4 and 6, in addition to squeeze-and-excitation blocks. [TL19a].

Stage *(i)*	Operator *(Fi)*	Resolution (*Hi* $\times$ *Wi*)	Channels (*Ci*)	No of layers (*Li*)
1	Conv, k 3×3	224×224	32	1
2	Conv, k 3×3	112×112	16	1
3	Conv, k 3×3	112×112	24	2
4	Conv, k 5×5	56×56	40	2
5	Conv, k 3×3	28×28	80	3
6	Conv, k 5×5	14×14	112	3
7	Conv, k 5×5	14×14	192	4
8	Conv, k 3×3	7×7	320	1
9	Conv 1×1 ,Pooling, FC	7×7	1280	1

Table 4: EfficientNet-B0 network, other EfficientNet-B4 & B5 varies with resolution (*Hi* $\times$ *Wi*) [TL19a]. Conv indicates convolutional layer, FC denotes Fully Connected, and k denotes kernel.

Blocks	Operator	Kernel	Channels
SEBlock-1	Conv	1×1	64
	Conv	3×3	64
	Conv	1×1	256
	FC	1×1	[16,256]
SEBlock-2	Conv	1×1	128
	Conv	3×3	128
	Conv	1×1	512
	FC	1×1	[32,512]
SEBlock-3	Conv	1×1	256
	Conv	3×3	256
	Conv	1×1	1024
	FC	1×1	[64,1024]
SEBlock-4	Conv	1×1	512
	Conv	3×3	512
	Conv	1×1	2048
	FC	1×1	[128,2048]

Table 5: The SEBlocks of Seresnext50_32x4d are illustrated, where Conv represents the convolutional layer, and FC represents Fully Connected layers.

Stage *(i)*	Operator *(Fi)*	Kernel ($K \times K$)	Channels (*Ci*)	No of layers (*Li*)
1	Conv	7×7	64	1
2	Max Pooling	3×3	64	1
3	SEBlock-1	-	[16,256]	3
4	SEBlock-2	-	[32,512]	4
5	SEBlock-3	-	[64,1024]	6
6	SEBlock-4	-	[128,2048]	3
7	Global average pooling	1×1	-	1
8	Dense	-	1000	1
9	Dense	-	16	1

Table 6: Configuration of the Seresnext50_32x4d architecture, the SEBlock is discussed in Table 5. Conv stands for convolutional layer.

3.5 Implementation

Figure 2 represents the structure of the scalable FL framework. Table 7 represents the experimental setup.

The server PC-3 orchestrates the training process, and the steps are discussed in detail.

1. **Client and Server:** The client systems are PC-1 and PC-2, the server system is PC-3, and the docker container was created and deployed to install the software dependencies on the client and server systems [Sam22].
2. **Broadcast:** The client systems-1 & 2 downloads the model weights and training scripts from the Nextcloud, which was synced to the server system.
3. **Experiments:** Table 7 describes the experimental design for the FL approach. The model architecture is discussed in section 3.4. The client systems 1 & 2 were trained for 20 epochs with a learning rate 0.0001 and cosine-annealing as the learning rate scheduler.
4. **Schedule**: The experiments were automated with the help of Slack API. A Slack FL-Bot was created to interact with the client and server systems as presented in Figure 3. In Exp-1 and Exp-2 from Table 7, the client systems (PC-1 & PC-2) and server system (PC-3) were connected to Slack, the PC-1 starts training, PC-2 waits for the message (training has been completed, hence saves the model in Nextcloud) from Slack to start it's training and vice-versa. In Exp-3, the server system (PC-3) waits for the message from Slack for the training to complete and performs model averaging.
5. **Model Averaging:** The server system collects the updated models from Nextcloud. Model averaging is a technique where multiple individually trained models are combined. Each individual model has its own strengths and weaknesses hence, the final model is compiled with the strengths of the individual models. The weighted averag-

ing technique was applied, this technique calculates the contribution of each trained model and is weighted proportionally to its capability or skill.

6. **Model update:** The server system updates the averaged model computed from the client systems.

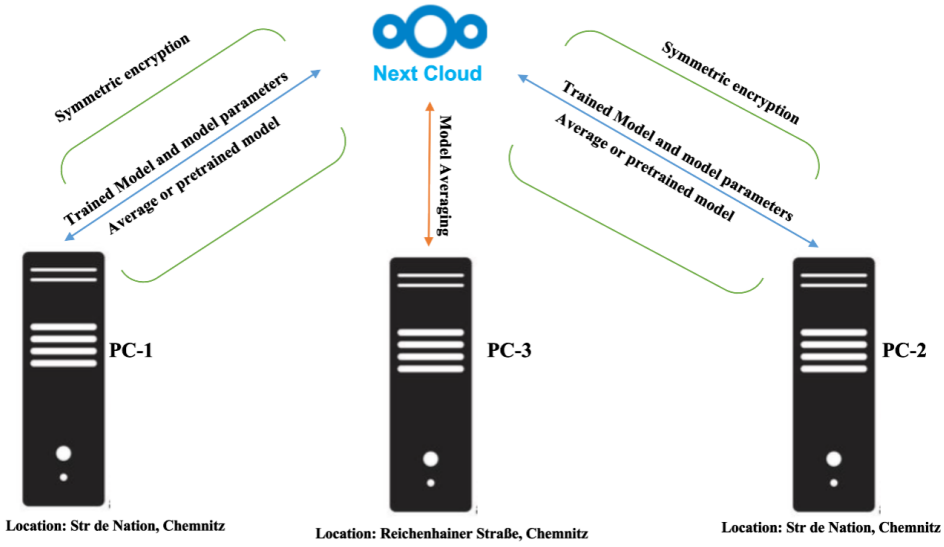

Fig. 2: Illustration of our proposed FL process, the client systems denoted PC-1 & PC-2 are from the same location and server system-3 is from a different location

Experimental ID	Description
Exp-1	Trained model from PC-1 was used as a pretrained model to train the model in PC-2
Exp-2	Trained model from PC-2 was used as a pretrained model to train the model in PC-1
Exp-3	Trained models from PC-1 and PC-2 were sent to the server to perform the model averaging

Table 7: FL experimental design

Figure 4 describes the graphical user interface (GUI), which was designed using HTML, CSS, and Javascript and comprises 3 modules. The first column comprises the model configuration where the hyperparameters are set, the hyperparameter values are predefined, but the users can modify the them also manually. The second column is framework selection, which comprises 3 dropdown menus, namely machine learning framework (Pytorch, Tensorflow, Sklearn), fusion method (weighted average, simple average), and dataset (oct images, fundus images, text mining). In the dataset dropdown menu, fundus images and text mining are just placeholders for other models that might require FL in the future of

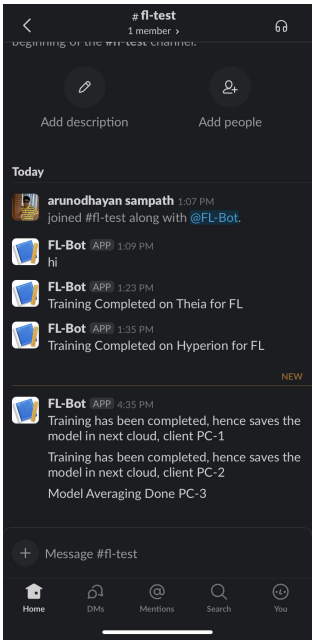

Fig. 3: Slack FL-Bot interaction

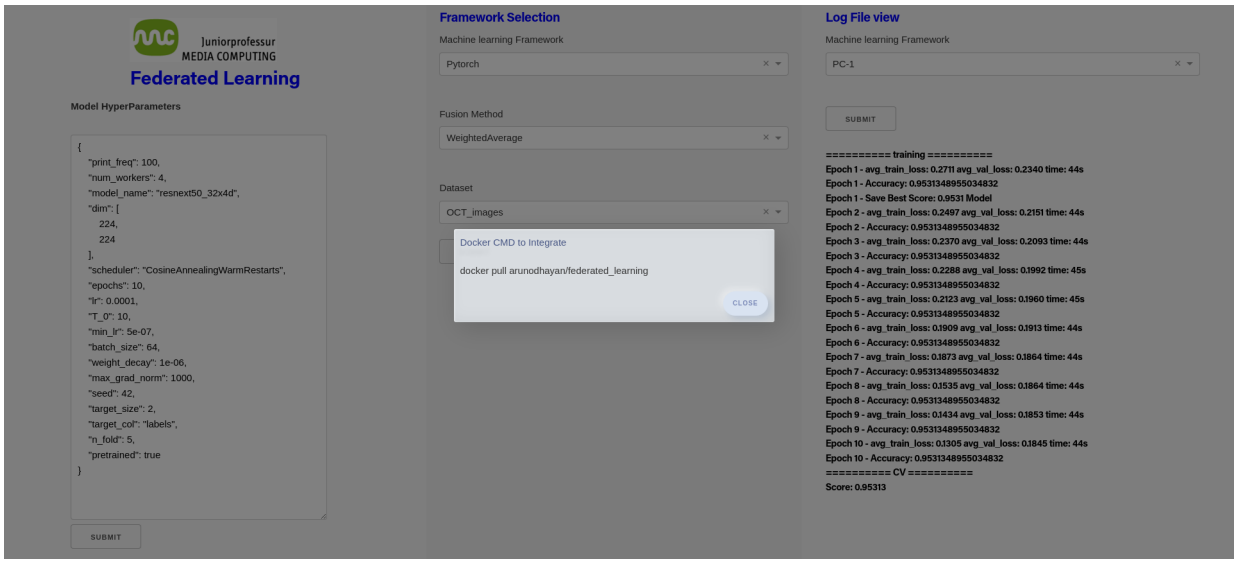

Fig. 4: The UI interface with a docker pull pop-up after completing the framework selection column

our ophthalmic research. The selection of respective dropdown menus leads to a popup message for docker pull, as displayed in Figure 4. The final column comprises log details from the user system.

4 Results

Table 8 summarizes the results. Mixnet has 5 million parameters, a less deep model architecture compared to other architectures containing more than 11 million parameters. The less deep mixnet model with our FL experimental setup from Table 7 achieves an F1 score using the EXP-1 setup closer to the centralized approach results.

Architectures	**Centralized slice-wise F1-score**	**FL slice-wise F1-score**		
		EXP-1	**EXP-2**	**EXP-3**
EfficientNet-B0	59.4 ± 1.1	**58.6**	57.2	58.1
EfficientNet-B4	60.9 ± 1.1	**60.8**	60.1	59.2
EfficientNet-B5	58.6 ± 0.5	**58.4**	57.4	58.0
EfficientNet-v2s	60.3 ± 0.4	**59.7**	59.0	58.7
Seresnext50_32x4d	65.0 ± 1.0	**64.9**	62.3	62.4
Tf_Mixnet_s	73.6 ± 0.9	**72.5**	71.8	70.2

Table 8: Comparison of F1-scores from centralized data training vs. FL-based results (EXP-1, EXP-2 and EXP-3, as documented in Table 7, respectively) based on 6 different architectures.

5 Conclusion and future work

In this paper, a federated learning framework was proposed to classify 16 different OCT biomarkers from a scientific community dataset. The best prediction results obtained from the FL approach with a macro average F1-score of 72.5% are close to our centralized data processing approach with 73.6%. The obtained model weights and other respective files from client systems are encrypted before transmitting them to the server system to perform model averaging. FL plays a crucial role in reducing data privacy risks when handling medical data allowing to train AI system without data having to leave the clinical site. Automated biomarker classification is supposed to assist doctors in analyzing patient OCT data. However, what classification accuracy is necessary in clinical routine and whether a higher accuracy will lead to more acceptance on the physician side for the use of AI is the subject of future research.

This first successful feasibility study encourages us to extend our current approach to OCTs at different clinical sites in the future to study the image classification capability and text mining with daily routine data. Blockchain could be implemented to communicate with the client and server systems to ensure better encryption of data and hence create the most secure communication. However, the FL method and technology must also lead to acceptance by data privacy officers and ethics committees when it comes to data protection, data security and legal certainty.

Acknowledgment

This research was funded by the Federal Ministry of Education and Research, namely the *Medical Informatics Hub in Saxony (MiHUBx)* with the grant number 01ZZ2101C.

References

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, page 439–450, New York, NY, USA, 2000. Association for Computing Machinery.

[CBPA14] Sourabh Chandra, Siddhartha Bhattacharyya, Smita Paira, and Sk Alam. A study and analysis on symmetric cryptography. 11 2014.

[CMOB19] Mingqing Chen, Rajiv Mathews, Tom Ouyang, and Françoise Beaufays. Federated learning of out-of-vocabulary words. *CoRR*, abs/1903.10635, 2019.

[EL01] A. Eskicioglu and L. Litwin. Cryptography. *IEEE Potentials*, 20(1):36–38, 2001.

[GDG+17] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

[GSvD+20] Filip Granqvist, Matt Seigel, Rogier van Dalen, Áine Cahill, Stephen Shum, and Matthias Paulik. Improving on-device speaker verification using federated learning with privacy, 2020.

[HSS17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks, 2017.

[HYF+18] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. Loadaboost: Loss-based adaboost federated machine learning on medical data. *CoRR*, abs/1811.12629, 2018.

[KMA+19] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *CoRR*, abs/1912.04977, 2019.

[LB20] Stefan Lenz and Harald Binder. Deep generative models in DataSHIELD, 2020. Publisher: arXiv Version Number: 1.

[MKY+22] Mohit Prabhushankar, Kiran Kokilepersaud, Yash-Yee Logan, Stephanie Trejo Corona, Ghassan AlRegib, and Charles Wykoff. Olives dataset: Ophthalmic labels for investigating visual eye semantics, 2022.

[MMRyA16] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs/1602.05629, 2016.

[NSSK16] Satish Balasaheb Nimse, Mukesh Digambar Sonawane, Keum-Soo Song, and Taisun Kim. Biomarker detection technologies and future directions. *Analyst*, 141(3):740–755, 2016.

[PKL+22] Mohit Prabhushankar, Kiran Kokilepersaud, Yash-yee Logan, Stephanie Trejo Corona, Ghassan AlRegib, and Charles Wykoff. OLIVES Dataset: Ophthalmic Labels for Investigating Visual Eye Semantics, June 2022. arXiv:2209.11195 [cs, eess].

[RMRB19] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *CoRR*, abs/1906.04329, 2019.

[Sam22] Arunodhayan Sampathkumar. Federated learning docker hub, January 2022.

[SFM+16] Virginia Smith, Simone Forte, Chenxin Ma, Martin Takác, Michael I. Jordan, and Martin Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *CoRR*, abs/1611.02189, 2016.

[SOS+19] Philipp Seeböck, José Ignacio Orlando, Thomas Schlegl, Sebastian M. Waldstein, Hrvoje Bogunović, Sophie Klimscha, Georg Langs, and Ursula Schmidt-Erfurth. Exploiting Epistemic Uncertainty of Anatomy Segmentation for Anomaly Detection in Retinal OCT. *IEEE Transactions on Medical Imaging*, 39(1):87–98, 2019.

[SS15] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1310–1321, New York, NY, USA, 2015. Association for Computing Machinery.

[TL19a] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019.

[TL19b] Mingxing Tan and Quoc V. Le. Mixconv: Mixed depthwise convolutional kernels, 2019.

[VYJ08] Jaideep Vaidya, Hwanjo Yu, and Xiaoqian Jiang. Privacy-preserving svm classification. *Knowl. Inf. Syst.*, 14(2):161–178, jan 2008.

[YAE+18] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *CoRR*, abs/1812.02903, 2018.