DeMed: A Novel and Efficient Decentralized Learning Framework for Medical Images Classification on Blockchain

Garima Aggarwal[1⋆], Chun-Yin Huang[1⋆], Di Fan[2], Xiaoxiao Li[1], and Zehua Wang[1]

[1] University of British Columbia, Vancouver, Canada
[2] University of South California, Los Angeles, California, USA
{xiaoxiao.li, zwang}@ece.ubc.ca

Abstract. Training predictive models with decentralized medical data can boost the healthcare research and is important for healthcare applications. Although the federated learning (FL) was proposed to build the predictive models, how to improve the security and robustness of a learning system to resist the accidental or malicious modification of data records are still the open questions. In this paper, we describe `DeMed`, a privacy-preserving decentralized medical image analysis framework empowered by blockchain technology. While blockchain is limited in serial computing, the decentralized data interaction in blockchain is very desired to preserve the data privacy when training models. To adapt blockchain in medical image analysis, our framework consists of the self-supervised learning part running on users' local devices and the smart contract part running on blockchain. The prior is to obtain the provable linearly separable low-dimensional representations of local medical images and the latter is to obtain the classifier by synthetically absorbing users' self-supervised learning results. The proposed `DeMed` is validated on two independent medical image classification tasks on pathological data and chest X-ray. Our work provides an open platform and arena for FL, where everyone can deploy a smart contract to attract contributors for medical image classification in a secure and decentralized manner while preserving the privacy in medical images.

Keywords: Blockchain · Federated Learning · Self-supervised Learning.

1 Introduction

Machine learning (ML) models have shown their advantage in many different tasks in healthcare filed. The medical image analysis is one of the most important applications. To effectively train a high-quality deep learning model, the aggregation of a significant amount of patient information is required. Multi-institutional healthcare predictive model can accelerate research and facilitate

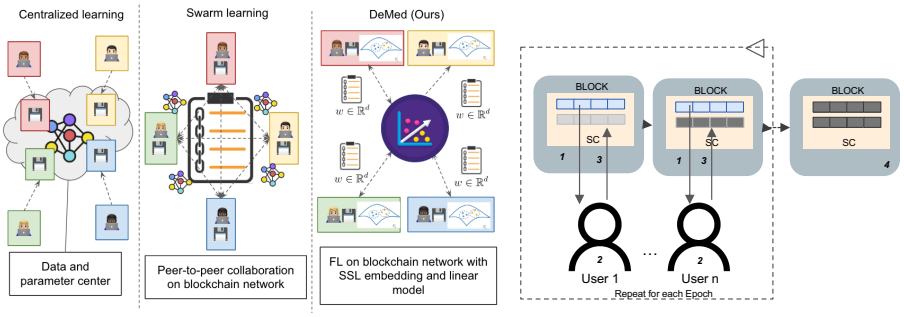
⋆ These authors contributed equally to this work

quality improvement on patient-care by leveraging different data sources and learning a model from data originated from the other institutions. However, improper data disclosure could place sensitive personal health information at risk. In addition, regulations such as GDPR [22] and HIPPA [8] strictly require protecting user information and granting transparent authorizations for the use of healthcare data.

Although federated learning [19] (FL) can be a solution to training ML models in a multi-party setting without data sharing, the users in FL still must share other forms of sensitive information (*e.g.*, model gradients or weights) to a *centralized service*. Such sharing is problematic when the central third party is not trustworthy, as prior work has demonstrated that adversaries can attack the model or data by the poisoning attack [26] and inversion attack [28] through observation of the target's shared model updates in the central server.

The blockchain [20] has emerged as a more appropriate system to facilitate private, verifiable, crowd-sourced decentralized computation, which is based on peer-to-peer networking and coordination while maintaining confidentiality without the need for a central coordinator, thereby going beyond FL. In a blockchain system, the data records are not saved in a centralized data server but maintained by network peers with consecutive data blocks. Further, the blockchain system provides an open platform and arena for FL, which enables sharing ML models among all parties without an intermediary. With blockchain and smart contact (SC), it is not the privilege of the big institutes to propose and train the learning models, but everyone can deploy a SC to attract contributors for medical image classification in a decentralized manner. However, there are inevitable obstacles to launching deep learning (DL)-based FL on the blockchain. First, latency and capacity are two fundamental elements that limit the throughput on blockchain. For example, on the Ethereum blockchain, the cost necessary to perform a transaction on the network is known as 'gas cost'. Transmission of DL models with hundreds of thousands of parameters hampers their practical utility. Second, without a central controller, if something goes wrong in a model training, *i.e.,* receiving weights from malicious users, we don't have a clear idea of how to identify the problem and correct it.

To overcome the aforementioned limitations, we propose `DeMed`, which is a framework for decentralized medical image analysis. It can reduce the input dimension of medical data to the point where the features are provably separable using a simple linear classifier. To this end, we first leverage the state-of-the-art reconstruction-based self-supervised learning (SSL) method, MAE [14], for low-dimensional representation learning. We then propose a *lightweight yet reliable* metric to select high quality users. Furthermore, we write a SC [9] using Solidity [6] for model parameter transmission. We tested the system on microscopic and X-Ray image classification tasks [21, 23], and achieve comparable performance with Swarm Learning [24] and Centralized Learning, while protecting the model from users that may degrade the model. The comparison between the learning strategies are given in Fig. 1(a).



(a) Comparison between learning strategies. (b) Blockchain module in `DeMed`

Fig. 1: (a) Comparison with different learning strategy. For Centralized Learning, a center collects data and be in charge of training the model. For Swarm Learning, users under SC keep their own data and train the model in peer-to-peer communications. For `DeMed`, users keep their data, train their own local model, and upload the weights to blockchain. The strategy is similar to Federated Learning, but the weights are aggregated and protected inside SC. (b) Four steps in the blockchain module of `DeMed`: i) every global epoch, users download same weights from the SC (blue). ii) Each user trains these weights locally with their respective data and iii) uploads them to the SC (grey). iv) Weights are aggregated after the epoch and original weights are updated to the new aggregated weights, to be used in the next epoch.

2 Preliminary

2.1 Blockchain

A blockchain system [1] is a decentralized data processing and maintaining system built on top of the peer-to-peer computer networks. Each peer in blockchain saves the data in the bundles (*i.e.*, blocks) which are chained up in chronological order. All the data records in the chain of blocks, so called *blockchain*, are maintained by each peer individually. Having one peer with its local data copy tampered does not affect the global data records, which makes the blockchain system be resistance to tampering. Another attractive feature in blockchain is no single point of failure, when comparing with the traditional database system. Every peer in the peer-to-peer network can provide the data access service to the public. Besides, the evolution history of the data records are fully traceable. Indeed, the data records in the chain of blocks are no more than the state transition events which are called *transactions* in blockchain [2, 3].

There are several works utilizing blockchain for FL. For example, [15] uses SVM over blockchain based federated learning which enables different operators to train intelligent driving models without sharing data. [18] investigates blockchain assisted FL that punishes malicious users by the reward system, and ensures robustness in FL training. [10] leverages Private Blockchain and Public Blockchain to attain accountability, privacy, and robustness, and propose an off-chain trojan detection for malicious users. Most of the related works focuses on

privacy and robustness concerns in FL. However, to the best of our knowledge, we are the first one that utilizes SSL to facilitate blockchain based training on large Deep Learning models.

2.2 Self-supervised learning (SSL)

SSL solves auxiliary prediction tasks (known as pretext tasks) without requiring labeled data to learn useful semantic representations. These pretext tasks are created solely using the input features, such as predicting a missing image patch, recovering the color channels of an image from context, predicting missing words in text, or forcing the similarity the different views of images, *etc.* [11, 13, 14, 27]. They improve the effectiveness of learning representations for downstream prediction tasks. Studies have shown that simple machine learning model, such as linear classifier, can achieve superiors performance by taking the embedded feature learned by SSL. Empirical and theoretical results have shown the features learned by proper SSL strategy are linearly separable using simple classifiers [16].

3 Method

3.1 Overview of the framework

We aim to train a medical image classifier on a blockchain via SC ⋆. Using the `DeMed` pipeline, the input dimensions of the medical data are reduced to the point where the features are separable by a linear classifier, thus also reducing the number of parameters that need to be stored in the SC. This makes our system viable even without integration of decentralized storage infrastructure. We collect publicly available *in-domain* data to pre-train MAE and distribute the MAE encoder as feature extractor to all users.⋆ The users can use the extractor to obtain the features of their own data and only the weights of the linear classifier will be trained and uploaded to the SC where the aggregation is done automatically. In this paper, we implement two different aggregation methods. The blockchain module of `DeMed` pipeline is shown in Fig. 1(b). Note `DeMed` is different from two existing learning framework for mult-user data learning: Centralized Learning and Swarm Learning (shown in Fig. 1(a)). Centralized Learning aggregates all weights from the users which requires a server center, while Swarm Learning requires all users to train at the same time at the blockchain side and directly write the whole deep model to SC.

We consider there is a hospital that wants to train a medical medical image classifier but doesn't have enough data. The hospital initializes a `DeMed` system for the task and is in charge of collecting *in-domain* unlabeled data, training a SSL representation extractor, and distributing the extractor to the users. The users will contribute their data by uploading the weights of locally trained linear

⋆ https://github.com/ubc-tea/DeMed-DeCaF22/blob/main/contracts/decentraldl.sol

⋆ An alternative way is to pre-train MAE using ImageNet and finetune on the collected data afterwards, if the number of the collected data is low.

classifier. `DeMed` is a learning framework that launches FL on blockchain. We aim to do an in depth privacy analysis in future work to investigate the privacy preserving attribute of `DeMed`.

3.2 Launch efficient deep learning training on blockchain

Self-supervised learning embedding space Motivated by [16], a well-trained SSL backbone can project the data onto a linearly separable space under proper assumptions. We utilize a state-of-the-art reconstruction-based SSL framework, Masked AutoEncoder(MAE) [14], as our feature extractor. MAE utilizes state-of-the-art image classification framework, Vision Transformer (ViT) [12], as the encoder for semantic feature extraction, and uses a lighter version of ViT as decoder. It divides an input image into patches, randomly blocks a certain percentage of image patches, and feeds them into the autoencoder architecture. By blocking out a large amount of image patches, the model is forced to learn a more complete representation. With the aim of positional embedding and transformer architecture, MAE is able to generalize the relationship between each image patch and obtain the semantic information among the whole image, which achieves the state-of-the-art performance in self-supervised image representation training. The pre-trained MAE encoder is then distributed to the users in SC.

Training federated linear model on blockchain We deploy the SC in Ethereum [25] blockchain to facilitate privacy-preserving FL. Ethereum can be seen as a transaction-based state machine, and a transaction is a cryptographically signed instruction constructed by an actor. Ethereum blockchain provides a mechanism to facilitate transactions between two consenting parties, which is called the SC. [9] SC is a piece of code, residing on a blockchain based platform, that executes an agreement or a logic. The code itself is replicated on multiple nodes of the blockchain, hence demarking the permanence, security and immutability of agreed upon logic. When the code is executed, a new block is added to the blockchain. The code is executed only on acceptance of all the parameters for the called functions.

In our pipeline, the communication exists between a hospital and the users of the system through the Ethereum and smart contracts. The transactions in our pipeline include storing weights in the SC, downloading the weights from the SC, and aggregating these weights. The only trained weights are from the classifier, which we use a linear layer. To begin with, the hospital will initialize the weights in the blockchain 3.2. For every epoch, the users download the weights from the blockchain, update the weights on their data, and upload the updated weights to the blockchain. Weights are gathered from all the users in the SC for aggregation.

3.3 Secure training on blockchain with user selection

One essential step in `DeMed` is model aggregation. Considering the communication cost in writing model weights to SC, we select a portion of users in each

global round. In this section, we describe a naïve weights aggregation method and a more advanced aggregation strategies that is robust to malicious users. The logic to choose the users based on any of the following two aggregation methods lies within the hospital. In case of user selection, the users add their norms and cosines to the SC, which help the hospital make a decision on user selection for the secure aggregation. To reduce the gas consumption for blockchain transactions, we could adopt Layer-2 solutions [4] such as the Optimistic Rollup [5] or Zero Knowledge Proof Rollup [7] technologies. They bundle up transactions and submit a summary of the changes required to represent all the transactions in a batch rather than sending each transaction individually.

Naïve weights aggregation To ensure the model sees all the users' data, we divide the users into small sets (batches) where each set has B users. During training, we iteratively feed in B users' data until all users' data are "seen" by the model. For example, the users from i-th set will download the global weights after the weights of users from $(i-1)$-th set are aggregated.

User selection weights aggregation Although naïve weights aggregation makes use of all users' weights to contribute to the global model, it may lead to unstable convergence and is prone to be attacked by malicious users. Malicious users are those who tries to drag down model training by uploading poisoned weights. Therefore, we propose User Selection (US) weights aggregation that selects users that contribute better weights would allow more efficient training and avoids malicious users. To address this problem, we use the weight drifts (denoted by d) and cosine similarity (denoted by cos) for user selection, which are defined as follows:

$$d = ||W_0 - w_k||_2, \tag{1}$$

and

$$cos = \frac{V \cdot W_k}{max(||V||_2 \cdot ||W_k||_2, \epsilon)}, \tag{2}$$

with $V = W - W_0, W_k = w_k - W_0, \epsilon = e^{-8}$, where V is the direction of the gradient, W is the naïve aggregation of the epoch, w_k and W_k are the local model weights and gradient direction of the k-th user after training for that epoch, and W_0 is initial weights used to train for the model for the particular epoch. Note that this is similar to [17] but we calculate the V based on all the gradients of the *current* run. Furthermore, instead of using a single criteria, we leverage both weight drifts and cosine similarity in user selection, which is detailed in Section 4.1. Weight drift and cosine similarity aim to pick users who have weights closest to the other weights in distance and direction, respectively.

4 Experiment

4.1 Experiment Setup and Datasets

Setup We evaluate DeMed on 2 medical datasets: PCam [21], a microscopic dataset for identifying metastatic tissue in histopathologic scans of lymph node

sections and COVIDx [23], a chest X-Ray dataset for COVIDx classification. In our experiments, we divide a dataset into 3 disjoint sets:

- **Public Train Set**: randomly sampled large amount of data from the datasets. This resembles the public available *in-domain* data and is used to pre-trained the SSL representation extractor.
- **Validation Set**: Randomly sampled data points for testing. This simulates the testing set that is kept in the smart contract to examine the weights uploaded by the users.
- **User Train Sets**: Randomly sampled 100 data points for 16 users. This resembles the data that each user has.

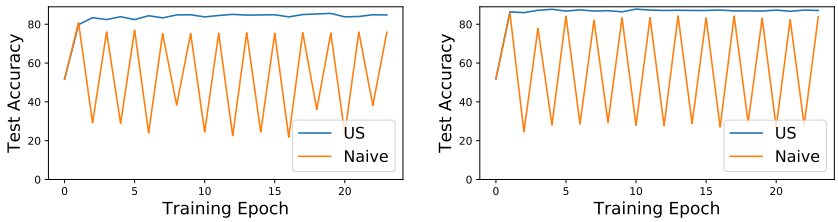
For training MAE, the experiments are run on NVIDIA DeForce RTX 3090 Graphic card with PyTorch. We follow the training strategy in [14]. However, due to the hardware limitation, we fix the batch size to 256 and adjust the training epoch accordingly. For DeMed learning, we use the extracted representations to train a linear layer that maps the embedding dimensions into predictions. Here, the embedding dimension of MAE is 1024 and number of classes is 2, so a fully connected (1024×1) layer and BCELoss is applied. Note that although we only simulate 16 users in **User Train Sets**, the system is scalable to more users. We test the performances of scenarios that there are only 2, 4, 8 users are allowed to to join per transaction, and found that the accuracies are similar. In the following experiments, we will only show the results for 8 users per transaction (please refer to Table. 2).

Due to the lightweight of `DeMed`, we could launch the blockchain module on CPU only. We used Ganache as a local blockchain for our experiment. The SC for the weights of linear layer was written in Solidity programming language. For training local linear classifier, the experiments run on 8-Core Intel Core i9 processor. Each user will download the global weights, train for 3 epochs locally, and then upload the new weights to the SC. Learning rate is set to $5e^{-3}$, and Adam optimizer is selected.

The Naïve aggregation of weights does not filter out malicious users from the system. Hence, we used model weights drift d (Eq. (1)) and cosine similarity cos (Eq. (2)) to filter out users from our system that would lead to a decline in the accuracy. We first request calculating the d for all users, and band weight submission for those whose d are too large/small (we remove 10 users from this step). Second, we request calculating the cos for the rest 10 users, and pick the 2,4,8 number of users with the largest cosine similarity. Finally, we aggregate the weights of the selected users as the final weights for the respective epoch.

4.2 Comparison between aggregation methods

We evaluate training results of the two aggregation methods. We first show that Naïve aggregation and US aggregation result in similar performance. Then we show adding one malicious user will degrade Naïve aggregation's performance while US aggregation is not influenced by the malicious user.



(a) Covidx (b) PCam

Fig. 2: Comparison of testing accuracy over training epochs for two weights aggregation methods: naïve vs user selection (US). One user is malicious. The zigzag curve for naïve aggregation and worse testing results indicate that it is prone to be attacked by malicious users. We show the results for selecting 8 users per transaction.

Table 1: Testing accuracy for `DeMed`(2, 4, 8 users) using Naïve Weights Aggregation and User Selection(US) Weights Aggregation.

# users/round	2		4		8	
Aggregation method	Naive	US	Naive	US	Naive	US
CovidX	84.1	84.1	84.6	84.1	84.4	85.2
PCam	86.2	86.5	87.4	87.3	87.2	87.3

Testing Performance We train the classification model for two datasets on `DeMed` (2, 4, 8 users cases) using Naïve Weights Aggregation and User Selection(US) Weights Aggregation as shown in Table 1. One can observe that using 8 users per aggregation gives the best results. The user selection method has slightly better accuracy as the best contributing users are selected for weight aggregation, while for Naïve method every user contributes their weights evenly.

Training with Malicious Users We simulate a malicious user attack by manipulating a user's weight into $W_{poisoned} = -10 \times W_{original}$. The accuracy curve is shown in Fig. 2. One can observe that the curve for naïve aggregation is zigzag. This is because for Naïve user aggregation, the malicious user also contributes it's weights, thus leading to declined accuracy whenever the model *sees* the malicious data. On the other hand, in case of user selection, the malicious user is screened out and accuracy does not decline.

4.3 Comparison between Learning Strategies

We train classification models for the 2 datasets on `DeMed` (2, 4, 8 users cases), Swarm Learning, and Centralized Learning, and the testing accuracy are shown in Table 2. One can observe that using 8 users in `DeMed` results in the best classification performance. Also, we would like to emphasize that `DeMed` can achieve comparable results while having better flexibility than Swarm Learning and being more privacy preserving than Centralized Learning.

Table 2: Testing accuracy for DeMed, Swarm Learning, and Centralized Learning.

Method	Centralized Learning	Swarm Learning	DeMed		
User selection	-	-	2	4	8
CovidX	84.8	84.8	84.1	84.1	85.2
PCam	87.8	87.9	86.5	87.3	87.4

5 Conclusion

We propose DeMed, an efficient decentralized learning framework that utilizes pre-trained SSL feature extractor to realize blockchain-based training on SC. By training classifier on the extracted features, we leverage a linear model on SC in a FL fashion. We also design user selection mechanism similar to [17] but with slight difference in finding the most representative users in each aggregation to detect malicious users. Overall, DeMed shows comparable model performance to Centralized Learning and Swarm Learning, while preserving security and flexibility. We believe that DeMed can facilitate privacy-preserving decentralized learning for medical image analysis.

Acknowledgement

This work is supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada (RGPIN-2021-02970, DGECR-2021-00187), NVIDIA Hardware Award, and Public Safety Canada (NS-5001-22170).

References

1. Blockchain. https://www.investopedia.com/terms/b/blockchain.asp, accessed: 2022-07-30
2. Blockchain transactions. https://onezero.medium.com/how-does-the-blockchain-work-98c8cd01d2ae, accessed: 2022-07-30
3. Etehreum transactions. https://ethereum.org/en/developers/docs/transactions/, accessed: 2022-07-30
4. Optimistic rollups. https://ethereum.org/en/developers/docs/scaling/, accessed: 2022-07-30
5. Optimistic rollups. https://ethereum.org/en/developers/docs/scaling/optimistic-rollups/, accessed: 2022-07-30
6. Solidity. https://docs.soliditylang.org/en/v0.8.15/, accessed: 2022-07-30
7. Zero-knowledge rollups. https://ethereum.org/en/developers/docs/scaling/zk-rollups/, accessed: 2022-07-30
8. Act, A.: Health insurance portability and accountability act of 1996. Public law **104**, 191 (1996)
9. Buterin, V.: Ethereum white paper: A next generation smart contract & decentralized application platform (2013), https://github.com/ethereum/wiki/wiki/White-Paper

10. Desai, H.B., Ozdayi, M.S., Kantarcioglu, M.: Blockfla: Accountable federated learning via hybrid blockchain architecture. In: Proceedings of the eleventh ACM conference on data and application security and privacy. pp. 101–112 (2021)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
13. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems **33**, 21271–21284 (2020)
14. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)
15. Hua, G., Zhu, L., Wu, J., Shen, C., Zhou, L., Lin, Q.: Blockchain-based federated learning for intelligent control in heavy haul railway. IEEE Access **8**, 176830–176839 (2020)
16. Lee, J.D., Lei, Q., Saunshi, N., Zhuo, J.: Predicting what you already know helps: Provable self-supervised learning. Advances in Neural Information Processing Systems **34** (2021)
17. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems **2**, 429–450 (2020)
18. Ma, C., Li, J., Ding, M., Shi, L., Wang, T., Han, Z., Poor, H.V.: When federated learning meets blockchain: A new distributed learning paradigm. arXiv preprint arXiv:2009.09338 (2020)
19. 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. pp. 1273–1282. PMLR (2017)
20. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review p. 21260 (2008)
21. Veeling, B.S., Linmans, J., Winkens, J., Cohen, T., Welling, M.: Rotation equivariant cnns for digital pathology. In: International Conference on Medical image computing and computer-assisted intervention. pp. 210–218. Springer (2018)
22. Voigt, P., Von dem Bussche, A.: The EU general data protection regulation (GDPR). Intersoft consulting (2018)
23. Wang, L., Lin, Z.Q., Wong, A.: Covid-net: a tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Scientific Reports **10**(1), 19549 (Nov 2020). https://doi.org/10.1038/s41598-020-76550-z, `https://doi.org/10.1038/s41598-020-76550-z`
24. Warnat-Herresthal, S., Schultze, H., Shastry, K.L., Manamohan, S., Mukherjee, S., Garg, V., Sarveswara, R., Händler, K., Pickkers, P., Aziz, N.A., et al.: Swarm learning for decentralized and confidential clinical machine learning. Nature **594**(7862), 265–270 (2021)
25. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger
26. Xie, C., Koyejo, S., Gupta, I.: Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In: International Conference on Machine Learning. pp. 6893–6901. PMLR (2019)

27. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision. pp. 649–666. Springer (2016)
28. Zhu, L., Han, S.: Deep leakage from gradients. In: Federated learning, pp. 17–31. Springer (2020)