FedCV: A Federated Learning Framework for Diverse Computer Vision Tasks

Chaoyang He¹, Alay Dilipbhai Shah¹, Zhenheng Tang², Di Fan¹
Adarshan Naiynar Sivashunmugam¹, Keerti Bhogaraju¹, Mita Shimpî¹, Li Shen³, Xiaowen Chu²,
Mahdi Soltanolkotabi¹, Salman Avestimehr¹
University of Southern California¹, Hong Kong Baptist University², Tencent AI Lab³
{chaoyang.he, avestime}@usc.edu

Abstract

Federated Learning (FL) is a distributed learning paradigm that can learn a global or personalized model from decentralized datasets on edge devices. However, in the computer vision domain, model performance in FL is far behind centralized training due to the lack of exploration in diverse tasks with a unified FL framework. FL has rarely been demonstrated effectively in advanced computer vision tasks such as object detection and image segmentation. To bridge the gap and facilitate the development of FL for computer vision tasks, in this work, we propose a federated learning library and benchmarking framework, named FedCV, to evaluate FL on the three most representative computer vision tasks: image classification, image segmentation, and object detection. We provide non-I.I.D. benchmarking datasets, models, and various reference FL algorithms. Our benchmark study suggests that there are multiple challenges that deserve future exploration: centralized training tricks may not be directly applied to FL; the non-I.I.D. dataset actually downgrades the model accuracy to some degree in different tasks; improving the system efficiency of federated training is challenging given the huge number of parameters and the per-client memory cost. We believe that such a library and benchmark, along with comparable evaluation settings, is necessary to make meaningful progress in FL on computer vision tasks. FedCV is publicly available: https://github.com/FedML-AI/FedCV.

1. Introduction

FL has the potential to rescue many interesting computer vision (CV) applications which centralized training cannot handle due to various issues such as privacy concerns (e.g. in medical settings), data transfer and maintenance costs (most notably in video analytic) [88, 89], or sensitivity of proprietary data (e.g. facial recognition) [37]. In essence, FL is an art of trade-offs among many optimization objectives [74, 14], including improving model accuracy and personal-
Second, CV model training normally requires large-scale computing research in a distributed computing environment, but current FL algorithms are mostly published as standalone simulations, which further enlarges the research gap (e.g., the recently released FedVision library [49] only contains object detection and single GPU training).

Third, the efficacy of proposed FL algorithms on diverse CV tasks is still vague. When combined with multiple optimization objectives such as privacy, security, and fairness, the problem becomes even more challenging. Currently, only image classification in small-scale datasets and models has been evaluated in these algorithms (see Table 16 in the Appendix). Researchers may attempt to solve a specific problem in realistic CV tasks by designing new algorithms, but the current research community lacks such a library to connect diverse CV tasks with algorithmic exploration.

Due to these obstacles, there is an urgent need to bridge the gap between pure algorithmic research and CV application-driven research. Our philosophy to do so can be illustrated in Figure 1. Specifically, we design a unified federated learning library, named FedCV, to connect various FL algorithms with multiple important CV tasks, including image segmentation and object detection. Under this framework, the benchmark suite is provided to assist further research exploration and fair comparison. To approach a realistic federated dataset, we provide methods to partition the dataset into non-identical and independent distribution. Model-wise, we believe the best solution for CV is to improve pre-training for SOTA models with efficient federated learning methods, which requires us to design efficient and effective task-specific models with pre-trained models as the backbone in the FL setting. To reduce the learning curve and engineering burden for CV researchers, we provide various representative FL algorithms as one line, easy-to-use APIs. Most importantly, these APIs provide distributed computing paradigm, which is essential to accelerating the federated training of most CV models. Moreover, we also make the framework flexible in exploring algorithms with new protocols of distributed computing, such as customizing the exchange information among clients and defining specialized training procedures.

To demonstrate the ability of our framework and provide benchmarking experimental results, we run experiments in three computer visions: image classification, image segmentation, and object detection. Our benchmark study suggests that there are multiple challenges that deserve future exploration: many deep learning training tricks may not be directly applied to FL; the non-I.I.D dataset actually degrades the model accuracy to some degree in different tasks; improving the system efficiency of federated training is challenging given the huge number of parameters and the per-client memory cost. We hope FedCV will serve as an easy-to-use platform for researchers to explore diverse research topics at the intersection of computer vision and federated learning, such as improving models, systems, or federated optimization methods.

2. Related Works

[33] is the first work that applies federated learning to a real-world image dataset, Google Landmark [78], which has now become the standard image dataset for federated learning research. [6, 41, 43] apply federated learning on medical image segmentation tasks, which aims at solving the issue in which the training data may not be available at a single medical institution due to data privacy regulations. In the object detection task, [84] proposes a KL divergence method to mitigate model accuracy loss due to non-I.I.D. Our work is closely related to FedVision [49], a federated learning framework for computer vision. It supports object detection in the smart city scenario using models including FastRCNN and YOLOv3. However, FedVision only supports the FedAvg algorithm and single-GPU training. Our FedCV platform provides diverse computer tasks and various FL algorithms. For federated learning in other application domains, we refer to the comprehensive vision paper [37].

3. Preliminary and Challenges

Federated learning (FL) is a distributed learning paradigm that can leverage a scattered and isolated dataset to train a global or personalized model for each client (participant) while achieving privacy preservation, compliance with regulatory requirements, and savings on communication and storage costs for such large edge data. The most straightforward formulation is to assume all clients need to collaboratively train a global model. Formally, the objective function is as follows:

\[
\min_W F(W) \overset{\text{def}}{=} \min \sum_{k=1}^N f^{(k)}(W) = \frac{1}{N} \sum_{k=1}^N \ell(W; X, y)
\]

In computer vision, \(W\) can be any CNN or Transformer model (e.g., ViT). \(f^{(k)}(W)\) is the \(k\)th client’s local objective function that measures the local empirical risk over the heterogeneous dataset \(D^k\). \(\ell\) is the loss function of the global CNN model. For the image classification task, \(\ell\) is the cross-entropy loss.

To solve this federated optimization problem, FedAvg is the first federated optimization algorithm to propose the concept of FL. To better understand the challenges of FL on CV, we rewrite its optimization process in Algorithm 1 with annotations. As we can see, there are several clear characteristics that distinguish FL from conventional distributed training in a sealed data center:
Algorithm 1 FedAvg Algorithm: A Challenge Perspective

1: **Initialization:** there is a number of clients in a network: the client $k$ has local dataset $D^k$; each client’s local model is initialized as $W_0$;

2: 

3: **Server Executes:**

4: for each round $t = 0, 1, 2, \ldots$ do

5: $S_t \leftarrow$ (sample a random set of clients)

6: for each client $k \in S_t$ in parallel do

7: $W_{t+1}^k \leftarrow \text{ClientUpdate}(k, W_t)$

8: end for

9: $W_{t+1} \leftarrow \frac{1}{n} \sum_{k=1}^{K} W_{t+1}^k$

10: end for

11:

12: **Client Training ($k$, $W$):** // Run on client $k$

13: $B \leftarrow$ (split $D^k$ into batches)

14: for each local epoch $i$ with $i = 1, 2, \ldots$ do

15: for batch $b \in B$ do

16: $W \leftarrow W - \eta \nabla_W F(W; b)$

17: end for

18: end for

19: return $W$ to server

1. Data heterogeneity and label deficiency at the edge. In conventional distributed training, centralized datasets are evenly distributed into multiple compute nodes for parallel computing, while in FL, the data is generated at the edge in a property of non-identical and independent distribution (non-I.I.D.). For example, in the CV scenario, smartphone users generate images or videos with distinct resolutions, qualities, and contents due to differences in their hardware and user behaviors. In addition, incentivizing users to label their private image and video data is challenging due to privacy concerns.

2. System constraints and heterogeneity. Training large DNN models at the edge is extremely challenging even when using the most powerful edge devices. In terms of memory, edge training requires significantly more memory than the edge inference requires. The bandwidth for edge devices is smaller than that of distributed training in the data center environment (InfiniBand can be used); the edge devices normally do not have GPU accelerators. What is even worse is that these system abilities are heterogeneous due to diverse hardware configurations.

3. Robustness and Privacy. Since federated training is not in a sealed data center environment, as is the traditional distributed training, it is easier to manipulate the data and model poisoning attacks. Therefore, making the training algorithm robust against attacks is also an important research direction in FL. In addition, although privacy preservation is one of the main goals, researchers also demonstrate that the exchanged gradient between the client and the server may, to some degree, lead to privacy leaks. More privacy-preserving techniques must be evaluated on various computer vision applications [81].

4. FedCV Design

To solve these challenges in diverse CV tasks, we need a flexible and efficient distributed training framework with easy-to-use APIs, benchmark datasets and models, and reference implementations for various FL algorithms.

---

**Figure 2. Overview of FedCV System Architecture Design**

To bridge the gap between CV and FL research, we have designed an open-source federated learning system for computer vision, named FedCV. FedCV is built based on the FedML research library [27], which is a widely used FL library that only support image classification, ResNet and simple CNN models. The system architecture of FedCV is illustrated in Figure 2. To distinguish FedCV from FedML, we color-code the modules specific to FedCV.

**Benchmark Suite for Diverse CV Tasks:** FedCV supports three computer vision tasks: image classification, image segmentation, and object detection. Related datasets and data loaders are provided. Users can either reuse our data distribution or manipulate the non-I.I.D. by setting hyperparameters. Models are curated for benchmark evaluation. More details of the benchmark suite are given in Section 5.

**Reference Implementation for Representative FL Algorithms:** Currently, FedCV includes the standard implementations of multiple state of the art FL algorithms: Federated Averaging (FedAvg) [52], FedOpt (server Adam) [58], FedNova (client optimizer) [75], FedProx [62], FedMA [73], as well as some novel algorithms that have diverse training paradigms and network typologies, including FedGKT.
(efficient edge training) [24], Decentralized FL [29], Vertical Federated Learning (VFL) [82], Split Learning [20, 72], Federated Neural Architecture Search (FedNAS) [23], and Turbo-Aggregate [65]. These algorithms support multi-GPU distributed training, which enables training to be completed in a reasonable amount of time. Note that most published FL optimization algorithms are based on standalone simulations, which lead to a extremely long training time. In this paper, we bridge this gap and make the CV-based FL research computationally affordable.

**Easy-to-use APIs for Algorithm Customization:** With the help of the FedML API design, FedCV enables diverse networks, flexible information exchange among workers/clients, and various training procedures. We can easily implement new FL algorithms in a distributed computing environment. We defer API design details to the Appendix.

**Other Functionality:** We support several development tools to simplify the research exploration. Specifically, researchers can load multiple clients into a single GPU, which scales up the client number with fewer GPUs, although there may be GPU contention among processes; in the lowest layer, FedCV reuses FedML-core APIs but further supports tensor-aware RPC (remote procedure call), which enables the communication between servers located at different data centers (e.g., different medical institutes); enhanced security and privacy primitive modules are added to support techniques such as secure aggregation in upper layers.

5. **FedCV Benchmark Suite: Datasets, Models, and Algorithms**

| Task             | Dataset      | Model                  |
|------------------|--------------|------------------------|
| Image Classification | CIFAR-100    | EfficientNet[67]       |
|                  |              | MobileNet[32]          |
| Image Classification | GLD-23K[78]  | ViT[12]                |
| Image Segmentation | PASCAL VOC[22] | DeeplabV3+             |
|                  |              | UNet[55]               |
| Object Detection  | COCO[48]     | YOLOv5[36]             |

We summarize the benchmark suite in FedCV in Table 1, and introduce such a curated list task-by-task as follows:

**Image Classification.** The curated datasets are Google Landmarks Dataset 23k (GLD-23K) [78] and CIFAR-100 dataset [39] with non-I.I.D partition. GLD-23K dataset is suggested by Tensorflow Federated [19], a natural federated dataset from smartphone users. For the model, we suggest EfficientNet [67] and MobileNet-V3 [32], which are two lightweight CNNs. Since the attention-based Transformer model has become a trending model in CV, we suggest Vision Transformer (ViT) [12] (ViT-B/16) to conduct experiments. As the research progresses, we may be able to support more efficient Transformers.

**Image Segmentation.** We use the augmented PASCAL VOC dataset with annotations from 11355 images [22]. These images are taken from the original PASCAL VOC 2011 dataset which contains 20 foreground object classes and one background class. For models, DeepLabV3+ [?] and U-Net [55] are supported since they are representative image segmentation models in centralized training. In our experiments, we utilize ResNet-101 and MobileNet-V2 as two backbones of DeepLabV3+ and U-Net.

**Object Detection.** We use the COCO [48] dataset since it contains realistic images that include detecting and segmenting objects found in everyday life through extensive use of Amazon Mechanical Turk. We then use YOLOv5 [36], an optimized version of YOLOv4 [3] as the baseline model. It outperforms all the previous versions and approaches EfficientDet Average Precision(AP) with higher frames per second (FPS). In YOLOv5, four network models (YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x) with different network depths and widths are provided to cater to various applications. We use these four models as the pretrained models.

![Non-I.I.D. data distribution on augmented PASCAL VOC dataset with different α values.](image)

**Non-I.I.D. Preprocessing.** Our non-I.I.D. partition method is Latent Dirichlet Allocation (LDA) [71], which is a common practice in FL research to obtain synthetic federated datasets. As an example, we visualize the non-I.I.D. in Figure 3. Further details of the dataset partition can be found in the Appendix. For all datasets, we provide data downloading scripts and data loaders to simplify the data preprocessing. Note that we will update regularly to support new datasets and models.

The supported FL algorithms include FedAvg, FedOpt and many other representative algorithms. We provide a full list and description in the appendix.
6. Experiments

In this section, we present the experimental results on image classification, image segmentation, and objective detection tasks on varying deep learning models and datasets.

6.1. Image Classification

6.1.1 Implementation Details

For image classification, the client number per round is 10. All experiments are conducted on a computing cluster with GTX 2080Ti. Each client has one GPU and the communication bandwidth is 10 Gbps. We conduct extensive experiments with EfficientNet [67], MobileNet V3 [32] and ViT [12] on CIFAR-100 [39], and CLD-23K [78],[19] datasets. The hyper-parameter settings are listed in the Appendix.

6.1.2 Experimental Results

The main experimental results are presented in table 2. Below, we provide detailed comparisons of the implemented classification models on the proposed FedCV platform.  

GLD-23k Non-IID vs. IID. Figure 4(a) shows that the test accuracy of centralized training with EfficientNet and MobileNet outperforms FedAvg training by ten percent. And for the ViT, the accuracy of centralized training is similar with FedAvg. 

Impacts of different degrees of Non-IID. Figure 4(b) and Figure 15 (in Appendix) show the influence of different degrees of Non-IID on the training performance of EfficientNet and MobileNetV3. Experimental results align with the results of LDA [71]. A higher $\alpha$ (i.e., lower degree of Non-IID) causes the test accuracy to increase. 

Fine-tuning vs. training from scratch. Figure 4(b), Figure 15 in appendix, and Figure 4(c) show that the performance of fine-tuning is more effective than training from scratch. For the convergence speed, fine-tuning can achieve a test accuracy of 60%, nearly 20 times faster than training from scratch. After training is completed, fine-tuning outperforms training from scratch by about 20 percent.

Momentum SGD vs SGD. Figure 4 (d), and Figures 14(a)-(b) (in appendix) show that SGD with momentum cannot guarantee better performance than vanilla SGD. When using EfficientNet on CIFAR-100 dataset of $\alpha = 0.5$, momentum SGD has similar performance to SGD with fine tuning, but with a much higher test accuracy than SGD training from scratch. With $\alpha = 0.1$, the performance of momentum SGD is not significantly influenced by fine-tuning, whereas vanilla SGD can see significant improvement.

Learning rate scheduler. Figure 5, and Figure 14(c)-(d) (in appendix) show an interesting result in which the linear learning rate decay may not improve the performance, and
Figure 5. Test accuracy on CIFAR-100 with EfficientNet trained with momentum SGD, with/without fine-tuning and learning rate scheduler. Hyper-parameters are set as Table 11 in appendix. Here, Sched. means using learning rate scheduler with step size of 0.99.

Efficiency analysis. We summarize the system performance of three models in Table 3, which demonstrate that if we train a large deep learning model such as ViT in the federated setting, there exists a huge communication overhead compared with small models. Furthermore, in the real federated environment, the communication bandwidth could be even worse.

### 6.2. Image Segmentation

#### 6.2.1 Implementation Details

For the image segmentation, we train DeeplabV3+ and U-Net within the FedCV platform, in which the number of clients involved in each round of image segmentation are either 4 or 8. These studies are carried out on a computing cluster with a Quadro RTX 5000 graphics card. Each client has one GPU, with a 15.754 GB/s communication bandwidth.

Table 4 comprises a list of models and hyper-parameters we explored for evaluating performance of segmentation tasks in the federated setting. Note that we use following abbreviation throughout our analysis: TT: Training Type for Backbone. There are three strategies that we use for training the backbone. (i) Fine-Tuning (FT): We start with a ImageNet-pretrained backbone and fine-tune it for our task. (ii) Freezed Backbone (FZ): Similarly to FT, we start with ImageNet pre-trained backbone but do not train or fine-tune the backbone at all to save on computational complexity. (iii) Scratch Federated Learning (SFL): Training the entire architecture end-to-end starting from scratch.

### Table 4. Dataset, models and hyper-parameters choices for federated image segmentation task.

| Model          | Backbone (TT) | DD     | C | mIOU  |
|----------------|---------------|--------|---|-------|
| DeepLabV3+     | ResNet-101 (FT) | IID    | 4 | 77.9% |
| DeepLabV3+     | ResNet-101 (FT) | N-IID  | 4 | 76.47%|
| DeepLabV3+     | ResNet-101 (FT) | N-IID  | 8 | 75.69%|
| DeepLabV3+     | ResNet-101 (SFL) | N-IID | 4 | 75.44%|
| DeepLabV3+     | ResNet-101 (FZ) | N-IID | 4 | 68.24%|
| DeepLabV3+     | MobileNetV2 (FT) | N-IID | 4 | 69.31%|
| U-Net          | ResNet-101 (FT) | IID    | 4 | 75.14%|
| U-Net          | ResNet-101 (FT) | N-IID  | 4 | 74.34%|
| U-Net          | ResNet-101 (FT) | N-IID  | 8 | 73.65%|
| U-Net          | ResNet-101 (SFL) | N-IID | 4 | 74.2% |
| U-Net          | ResNet-101 (FZ) | N-IID | 4 | 51.19%|
| U-Net          | MobileNetV2 (FF) | N-IID | 4 | 66.14%|

### Table 5. Summary of test results on Pascal VOC dataset for federated image segmentation task.

**Table 5** shows that pre-trained backbones coupled with fine-tuning results in only a slightly better performance (less than 2%) compared to training from scratch, which indicates that while pre-trained backbones aid in federated image segmentation accuracy, they are not necessary. This finding opens the door to advanced tasks such as medical imaging, where pre-
trained backbones may not be useful and end-to-end training from scratch is the only viable alternative.

**Batch Size vs. Memory Trade-Off.** Figure 6(b) and Table 6 show that a smaller batch size, such as 4 instead of 10, reduces memory by roughly a factor of two while sacrificing nearly 2% accuracy. This is an important trade off to make because edge devices in a federated learning setup may have constrained memory.

| Model          | Backbone    | BS | Memory   | mIoU   |
|----------------|-------------|----|----------|--------|
| DeeplabV3+     | ResNet-101  | 4  | 6119M    | 72.38% |
| DeeplabV3+     | ResNet-101  | 6  | 8009M    | 73.28% |
| DeeplabV3+     | ResNet-101  | 8  | 10545M   | 74.89% |
| DeeplabV3+     | ResNet-101  | 10 | 13084M   | 75.5%  |
| UNet           | ResNet-101  | 4  | 6032M    | 71.54% |
| UNet           | ResNet-101  | 6  | 8456M    | 71.89% |
| UNet           | ResNet-101  | 8  | 10056M   | 72.4%  |
| UNet           | ResNet-101  | 10 | 12219M   | 73.55% |

Table 6. Performance and memory analysis for various batch size of segmentation models on Pascal VOC Dataset. BS: Batch Size

**Data distribution impact analysis.** For various partition values α, Figure 3 depicts the distribution of classes among clients. Even when the partition factor changes from totally homogeneous to extremely heterogeneous, as shown in Figure 7 (a), the accuracy only degrades by about 2%. This further demonstrates that federated segmentation learning can instill enough generalization capability in local clients to allow them to perform well on unknown data, obviating the need for centralized or widely distributed data.

**Resiliency in the face of increasing clients** The number of rounds needed for the model to converge increases as the number of clients increases (see figure 7(b)). When compared to smaller client sizes, which are theoretically expected to perform better since each local client has more data points to train on, it has little effect on final accuracy after a sufficient number of rounds.

### 6.2.3 System Performance Analysis

ResNet is one of the most widely used backbones for encoder-decoder architecture in image segmentation tasks; however, it has a high computing cost that many edge devices might not be able to bear. There are two obvious ways to trim the cost down: (i) Freezing the pre-trained backbone; (ii) Plugging computationally efficient backbone (Eg. MobileNetV2). Figure 8 depicts the performance variance when one of the two described strategies is applied for backbones in DeeplabV3+ and UNet architectures for federated image segmentation. When compared to every other mix, the accuracy of ResNet-101 backbone is demonstrably higher. On the other hand, as shown in Table 7, the alternatives are extremely efficient at the cost of performance degradation.

### 6.3. Object Detection

#### 6.3.1 Implementation Details

For object detection, we use pre-trained YOLOV5 for federated learning experiments with the FedAvg algorithm. The client number we used include 4 and 8 for performance comparison. Each client was run at one GPU (NVIDIA V100). The metric in our experiments is mAP@0.5 (mean average precision with a threshold of 0.5 for IOU).

#### 6.3.2 Experimental Results

**Learning rate.** In the federated setting, different learning rates are evaluated. While keeping the other hyperparameters (e.g., client number is set to 4), we notice that \( lr=0.01 \) can have a better result compared to the other choices from Figure 9 (a).

**Non-I.I.D. evaluation.** For the client numbers 4 and 8, we use the partition method introduced in the appendix to obtain synthetic federated datasets. We found that when using YOLOv5, it is difficult for FedAvg to reach the same results as that of centralized for Non-I.I.D. dataset. Figure 9 shows there is a large gap between centralized training and FedAvg.
Figure 7. Performance evaluation of segmentation task on Pascal VOC dataset. Figure (a): Evaluating performance of DeeplabV3+ and UNet models with ResNet101 as a backbone on various partition factors. Figure (b): Evaluation performance of DeeplabV3+ and UNet models with ResNet-101 as backbone on varying number of clients.

Table 7. System performance chart of segmentation network architectures we considered. TT: Training Type. BS: Batch Size

| Model          | Backbone (TT) | Dataset       | Params   | FLOPS   | Memory (BS) | Total Time |
|----------------|---------------|---------------|----------|---------|-------------|------------|
| DeeplabV3+     | ResNet-101 (FT) | PASCAL VOC    | 59.34M   | 88.85G  | 7541M (10)  | 13084M     | 14.16h     |
| DeeplabV3+     | ResNet-101 (FZ) | PASCAL VOC    | 16.84M   | 88.85G  | 12104M (16) | 13084M     | 23.59h     |
| DeeplabV3+     | MobileNetV2 (FT) | PASCAL VOC    | 5.81M    | 26.56G  | 7687M (16)  | 13084M     | 20.5h      |
| UNet           | ResNet-101 (FT) | PASCAL VOC    | 51.51M   | 62.22G  | 7687M (16)  | 13084M     | 51.11h     |
| UNet           | ResNet-101 (FZ) | PASCAL VOC    | 9.01M    | 62.22G  | 7687M (16)  | 13084M     | 22.03h     |
| UNet           | MobileNetV2 (FT) | PASCAL VOC    | 7.91M    | 14.24G  | 7687M (16)  | 13084M     | 22.03h     |

Figure 8. Performance comparison of DeeplabV3+ and UNet with ResNet101 and MobileNetV2 as backbones. DeeplabV3+ (ResNet101) reaches a better accuracy compared to other alternatives. FT: Fine-Tuning Backbone. FZ: Freezed Backbone

Figure 9. Experiments on detection tasks on varying learning rates and number of clients. Figure (a): Non-IID data comparison with different learning rate. Figure (b): Non-IID data comparison with different number of clients.

Table 8. System performance of YOLOv5

| Model   | Layers | Parameters | FLOPS | Total Time |
|---------|--------|------------|-------|------------|
| YOLOv5s | 283    | 7.27M      | 17.1G | 25.1h      |
| YOLOv5m | 391    | 21.4M      | 51.4G | 49.3h      |
| YOLOv5l | 499    | 47.1M      | 115.6G| 73.5h      |
| YOLOv5x | 607    | 87.8M      | 219.0G| 92.4h      |

system performance of four different model variants. We can see that as the network structure depth and width increased among the four models, the model performed well with a better mAP.
7. Conclusion

In this work, we propose an easy-to-use federated learning framework for diverse computer vision tasks, including image classification, image segmentation, and object detection, dubbed FedCV. We provide several non-IID benchmarking datasets, models, and various reference FL algorithms. We hope that FedCV can open doors for researchers to develop new federated algorithms for various computer vision tasks.

FedML Ecosystem [27] aims to provide a one-stop scientific research platform through FedML Ecosystem and finally realize trustworthy ML/AI, which is more secure, scalable, efficient, and ubiquitous. FedCV serves as one of the key components of FedML ecosystem. The other important applications include FedNLP [47], FedGraphNN [25], and FedIoT [88].

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In International Conference on Artificial Intelligence and Statistics, pages 2938–2948, 2020.

[2] Arjun Nitin Bhalgaji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In International Conference on Machine Learning, pages 634–643, 2019.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.

[4] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. arXiv preprint arXiv:1611.04482, 2016.

[5] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv e-prints, page arXiv:1602.05629, Feb. 2016.

[6] Qi Chang, Hui Qu, Yikai Zhang, Mert Sabuncu, Chao Chen, Tong Zhang, and Dimitris Metaxas. Synthetic Learning: Learn From Distributed Asynchronized Discriminator GAN Without Sharing Medical Image Data. arXiv e-prints, page arXiv:2006.00080, May 2020.

[7] Chien-Lun Chen, Leana Golubchik, and Marco Paolieri. Backdoor attacks on federated meta-learning. arXiv preprint arXiv:2006.07026, 2020.

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. arXiv preprint arXiv:2011.10566, 2020.

[9] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. 2019.

[10] Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. arXiv preprint arXiv:2006.08848, 2020.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In International Conference on Learning Representations, 2021.

[13] David Enthoven and Zaid Al-Ars. An overview of federated deep learning privacy attacks and defensive strategies. arXiv preprint arXiv:2004.04676, 2020.

[14] Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and Salman Avastimehr. Fairfed: Enabling group fairness in federated learning. ICML 2021 - International Workshop on Federated Learning for User Privacy and Data Confidentiality, 2021.

[15] Afneza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. arXiv preprint arXiv:2002.07948, 2020.

[16] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. arXiv preprint arXiv:1808.04866, 2018.

[17] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. arXiv preprint arXiv:1712.07557, 2017.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS’14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[19] Google. Tensorflow federated datasets. https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets.

[20] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. Journal of Network and Computer Applications, 116:1–8, 2018.

[21] Filip Hanzely, Boxin Zhao, and Mladen Kolar. Personalized federated learning: A unified framework and universal optimization techniques. arXiv preprint arXiv:2102.09743, 2021.

[22] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In International Conference on Computer Vision (ICCV), 2011.

[23] Chaoyang He, Murali Annavaram, and Salman Avastimehr. Fednas: Federated deep learning via neural architecture search. CVPR 2020 Workshop on Neural Architecture Search and Beyond for Representation Learning, pages arXiv–2004, 2020.

[24] Chaoyang He, Murali Annavaram, and Salman Avastimehr. Group knowledge transfer: Federated learning of large cnns...
at the edge. NeurIPS 2020 (Advances in Neural Information-Processing Systems 2020), 2020.

[25] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. Fedgraphm: A federated learning system and benchmark for graph neural networks. DPML@ICLR 2021 and GNNSys@MLSys 2021, 2021.

[26] Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annavaram, and Salman Avetismehr. Spreadgm: Serverless multi-task federated learning for graph neural networks. International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2021 (FL-ICML’21) and Deep Learning on Graphs: Method and Applications with KDD 2021 (DLG-KDD’21), 2021.

[27] Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Han Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avetismehr. Fedm1: A research library and benchmark for federated machine learning. NeurIPS 2020 (Advances in Neural Information Processing Systems 2020) Federated Learning Workshop Best Paper Award, abs/2007.13518, 2020.

[28] Chaoyang He, Shen Li, Mahdi Soltanolkotabi, and Salman Avetismehr. Pipetraformer: Automated elastic pipelining for distributed training of large-scale models. In International Conference on Machine Learning, pages 4150–4159. PMLR, 2021.

[29] Chaoyang He, Conghui Tan, Hanlin Tang, Shuang Qiu, and Ji Liu. Central server free federated learning over single-sided trust social networks. NeurIPS 2020 (Advances in Neural Information Processing Systems 2020) - Federated Learning Workshop, 2020.

[30] Chaoyang He, Zhengyu Yang, Erum Mushtaq, Sunwoo Lee, Mahdi Soltanolkotabi, and Salman Avetismehr. Ssfl: Tackling label deficiency in federated learning via personalized self-supervision. arXiv preprint arXiv:2110.02470, 2021.

[31] Chaoyang He, Haishan Ye, Li Shen, and Tong Zhang. Milenas: Efficient neural architecture search via mixed-level reformulation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11990–11999, 2020.

[32] A. Howard, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le. Searching for mobilenetv3. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 1314–1324, 2019.

[33] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Federated Visual Classification with Real-World Data Distribution. arXiv e-prints, page arXiv:2003.08082, Mar. 2020.

[34] Han Huang, Li Shen, Chaoyang He, Weisheng Dong, Haozhi Huang, and Guangming Shi. Lightweight image super-resolution with hierarchical and differentiable neural architecture search. arXiv preprint arXiv:2105.03939, 2021.

[35] Yutao Huang, Lingyang Chu, Zirui Zhan, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. AAAI, 2021.

[36] Glenn Jocher. YOLov5, 2020.

[37] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. Foundations and Trends in Machine Learning, Vol 14, Issue 1–2, 2021.

[38] Sai Praneeth Karimireddy, Satyen Kale, Mehray Mohri, Sashank J Reddi, Sebastian U Stich, and Anand Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. arXiv preprint arXiv:1910.06378, 2019.

[39] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Master’s thesis, Department of Computer Science, University of Toronto, 2009.

[40] Sunwoo Lee, Tuo Zhang, Chaoyang He, and Salman Avetismehr. Layer-wise adaptive model aggregation for scalable federated learning. arXiv preprint arXiv:2110.10302, 2021.

[41] Daiqing Li, Amlan Kar, Nishant Ravikumar, Alejandro F Frangi, and Sanja Fidler. Fed-Sim: Federated Simulation for Medical Imaging. arXiv e-prints, page arXiv:2009.00668, Sept. 2020.

[42] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. arXiv preprint arXiv:1905.10497, 2019.

[43] Wenqi Li, Fausto Milletari, Daguang Xu, Nicola Riecke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M. Jorge Cardoso, and Andrew Feng. Privacy-preserving Federated Brain Tumour Segmentation. arXiv e-prints, page arXiv:1910.00962, Oct. 2019.

[44] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. arXiv preprint arXiv:1907.02189, 2019.

[45] Zhize Li, Dmitry Kovalev, Xin Qian, and Peter Richtárik. Acceleration for compressed gradient descent in distributed and federated optimization. arXiv preprint arXiv:2002.13164, 2020.

[46] Jiacheng Liang, Songze Li, Wensi Jiang, Bochuang Cao, and Chaoyang He. Omnilytics: A blockchain-based secure data market for decentralized machine learning. ICML 2021 - International Workshop on Federated Learning for User Privacy and Data Confidentiality, 2021.

[47] Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avetismehr. Fednlp: A research platform for federated learning in natural language processing. arXiv preprint arXiv:2104.08815, 2021.

[48] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.

[49] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning. In AAAI, pages 13172–13179, 2020.

[50] Grigory Malinovsky, Dmitry Kovalev, Elmir Gasanov, Laurent Condat, and Peter Richtarik. From local sgd to local fixed point methods for federated learning. arXiv preprint arXiv:2004.01442, 2020.
Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Thomas Brox Olaf Ronneberger, Philipp Fischer. U-net: Yishay Mansour, Mehryar Mohri, Jae Ro, and Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, San-
Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Saurav Prakash and A. Salman Avestimehr. Mitigating byzan-
Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-effi cient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, pages 1273–1282, 2017.
Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In 2019 IEEE Symposium on Security and Privacy (SP), pages 691–706. IEEE, 2019.
Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. arXiv preprint arXiv:1902.00146, 2019.
Thomas Brox Olaf Ronneberger, Philipp Fischer. U-net: Convolutional networks for biomedical image segmentation, 2015.
Triphuwanesh Orekondy, Seong Joon Oh, Yang Zhang, Bernt Schiele, and Mario Fritz. Gradient-leaks: Understanding and controlling deanonymization in federated learning. arXiv preprint arXiv:1805.05838, 2018.
Saurav Prakash and A. Salman Avetisian. Mitigating byzan-
tine attacks in federated learning. arXiv preprint arxiv:2010.07541, 2020.
Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. arXiv preprint arxiv:2003.00295, 2020.
Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. arXiv preprint arXiv:2007.07682, page 12, 2020.
Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhifeng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3):211–252, 2015.
Theo Ryffiel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. arXiv preprint arXiv:1811.04017, 2018.
Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. ArXiv, abs/1812.06127, 2018.
Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, Keith Rush, and Sushant Prakash. Federated recon-
struction: Partially local federated learning. arXiv preprint arXiv:2102.03448, 2021.
Jinhyun So, Basak Guler, and A. Salman Avetisianh. Byzantine-resilient secure federated learning. IEEE Journal on Selected Areas in Communication, Series on Machine Learning for Communications and Networks, 2020.
Jinhyun So, Basak Guler, and A Salman Avetisianh. Turbo-
aggregate: Breaking the quadratic aggregation barrier in se-
cure federated learning. arXiv preprint arXiv:2002.04156, 2020.
Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? arXiv preprint arXiv:1911.07963, 2019.
Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 6105–6114. PMLR, 09–15 Jun 2019.
Aleksiej Triastcyn and Boi Faltungs. Federated learning with bayesian differential privacy. In 2019 IEEE International Conference on Big Data (Big Data), pages 2587–2596. IEEE, 2019.
Aleksiej Triastcyn and Boi Faltungs. Federated generative privacy. IEEE Intelligent Systems, 2020.
Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In Pro-
cceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pages 1–11, 2019.
Matthew Brown Tzu-Ming Harry Hsu, Hang Qi. Measuring the effects of non-identical data distribution for federated visual classification, 2019.
Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564, 2020.
Hongyi Wang, Mikhail Yurockin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. arXiv preprint arXiv:2002.06440, 2020.
Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avetisian, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. arXiv preprint: arXiv:2107.06917, 2021.
Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. arXiv preprint arXiv:2007.07481, 2020.
Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representa-
tives: User-level privacy leakage from federated learning. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications, pages 2512–2520. IEEE, 2019.
Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. A framework for evaluating gradient leakage attacks in federated learning. arXiv preprint arXiv:2007.07481, 2020.
T. Weyand, A. Araujo, B. Cao, and J. Sim. Google land-
smarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2572–2581, 2020.
Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dbda: Distributed backdoor attacks against federated learning. In International Conference on Learning Representations, 2019.
[80] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. Hybridalpha: An efficient approach for privacy-preserving federated learning. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pages 13–23, 2019.

[81] Chien-Sheng Yang, Jinhyun So, Chaoyang He, Songze Li, Qian Yu, and Salman Avestimehr. Lightsecagg: Rethinking secure aggregation in federated learning. arXiv preprint arXiv:2109.14236, 2021.

[82] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol., 10(2), Jan. 2019.

[83] Felix X Yu, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Federated learning with only positive labels. arXiv preprint arXiv:2004.10342, 2020.

[84] Peihua Yu and Yunfeng Liu. Federated object detection: Optimizing object detection model with federated learning. In Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, pages 1–6, 2019.

[85] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. arXiv preprint arXiv:2002.04758, 2020.

[86] Geng Yuan, Xiaolong Ma, Wei Niu, Zhengang Li, Zhenglun Kong, Ning Liu, Yifan Gong, Zheng Zhan, Chaoyang He, Qing Jin, et al. Mest: Accurate and fast memory-economic sparse training framework on the edge. In Thirty-Fifth Conference on Neural Information Processing Systems, 2021.

[87] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. arXiv preprint arXiv:1905.12022, 2019.

[88] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and Salman Avestimehr. Federated learning for internet of things: Applications, challenges, and opportunities. arXiv preprint arXiv:2111.07494, 2021.

[89] Tuo Zhang, Chaoyang He, Tianhao Ma, Lei Gao, Mark Ma, and Amir Salman Avestimehr. Federated learning for internet of things. Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems, 2021.
Appendix

In the appendix, we provide more details of the benchmark suite and experiments.

7.1. Benchmark Suite

7.1.1 Dataset

CIFAR-100. The CIFAR-100 dataset [39] has 100 classes, of which each contains 600 images with size $32 \times 32$.

Google Landmarks Dataset 23k (GLD-23K) is a subset of Google Landmark Dataset 160k [78]. This GLD-23K dataset includes 203 classes, 233 clients, and 23080 images. We follow the setting of GLD-23K from Tensorflow federated [19]. We also provide the data loader for GLD-160K in our source code.

PASCAL VOC - Augmented. We use the augmented PASCAL VOC dataset with annotations from 11355 images [22]. These images are taken from the original PASCAL VOC 2011 dataset, which contains 20 foreground object classes and one background class.

COCO [48] is a dataset for detecting and segmenting objects found in everyday life through extensive use of Amazon Mechanical Turk.

7.1.2 Non-I.I.D. Partition and Distribution Visualization

For GLD-23K, we follow the setting of GLD-23K from Tensorflow federated [19], which means that the number of total clients is 233 for GLD-23K.

We make use of Latent Dirichlet Allocation (LDA) [71] method to partition CIFAR-100 and PASCAL VOC into non-I.I.D. dataset. The settings of our non-I.I.D. partition can be referred to table 9.

Comparing to the classification and segmentation task with one picture has one label using the LDA partition method, the object detection task always has several labels on one image. In this case, we take a different partition method. First, we calculate the frequency of each object in all images (one object is only counted one time even it occurs more than one time on one image). Second, we sort the objects’ frequency and put the object with the highest frequency into one client. Finally, we sort the remaining pictures and repeat the first step until all the images have been assigned to the clients. Figure 10 shows the COCO non-IID data distribution on 8 clients. Different color represents different clients, and we could see every categories label data are non-IID distributed to different clients.

We can see that the visualization matrix is sparse. Some clients have many samples of some labels, but few samples of some other labels. This makes training become much harder.

7.1.3 Models

EfficientNet [67] and MobileNet-V3 [32] are two light weighted convolutional neural networks. They achieves the goal of improving accuracy and greatly reducing the amount of model parameters and calculations. In this paper, we use EfficientNet-b0 and MobileNet-V3 Large to conduct experiments.

Vision Transformer (ViT) [12] is a novel neural net-
work exploiting transformer into Computer Vision and attain excellent results compared to state-of-the-art convolutional networks. We use ViT-B/16 to conduct experiments.

MobileNetV2 is a lightweight Convolutional Neural Network. It is primarily designed to support running neural networks in mobile and edge devices that have severe memory constraints. It implements Inverted Residuals concept where the residual connections are used between bottleneck layers. It also applies Linear BottleNecks and Depthwise Separable Convolutions concept. In our implementation, we use a network that was pre-trained on the ImageNet dataset.

DeepLabV3+ is a neural network that employs two main principles Atrous Convolutions, Depthwise Separable Convolutions, and Encoder-Decoder architecture (as shown in Figure 12 which attain great performance on Image Segmentation. In our experiments, we exploit MobileNet-V2 and ResNet-101 as two kinds of backbones of DeepLabV3+ to conduct our experiments.

U-Net [55] is a Convolutional Neural Network that follows an Encoder-Decoder architecture pattern. As shown in Figure 13, U-Net does not need a backbone network to perform segmentation. However, we have experimented with Resnet-101 and MobileNetV2 pre-trained backbones during experimentation to improve the segmentation output further.

YOLOv5 [36] is an optimized version of YOLOv4 [3]. It outperforms all the previous versions and gets near to EfficientDet Average Precision(AP) with higher frames per second (FPS). Four network models (Yolov5s, Yolov5m, Yolov5l, Yolov5x) with different network depths and widths cater to various applications. Here we use these four models as the pre-trained models.

7.1.4 FL Algorithms

Optimizer. Unlike vanilla SGD in FedAvg [5], here we use Momentum SGD to optimize models. For centralized training, our Momentum SGD is the same as the traditional Momentum SGD. However, for FedAvg, the accumulated gradient will be cleared when clients receive the global model from the server. The momentum coefficient is 0.9 for all experiments.

Learning Rate Scheduler. For centralized training on all tasks, we exploit linear learning rate decay each epoch. For FedAvg, we do not use a learning rate scheduler on all tasks. However, we try it on image classification in order to evaluate its effect. Note that the learning rate decay is based on the communication round, not the local epochs as the traditional training process.

FedAvg Training. We use the FedAvg [5] algorithm to conduct federated learning. We list our FedAvg training settings in table 10, in which $C$ means the number of clients.
| Dataset                | clients | Partition                  | Num of labels |
|-----------------------|---------|----------------------------|---------------|
| CIFAR-100             | 100     | LDA with $\alpha$ in [0.1, 0.5, 100] | 100           |
| GLD-23K               | 233     | Default                    | 203           |
| augmented PASCAL VOC  | 4, 8    | LDA with $\alpha$ in [0.1, 0.5, 100] | 21            |
| COCO                  | 8       | Our partition              | 80            |

Table 9. Summary of Dataset partition.

participating in calculation per round, and $E$ means the number of local epochs per round.

| Task                | C | E |
|---------------------|---|---|
| Image Classification| 10| 1 |
| Image Segmentation  | 4, 8| 1, 2 |
| Object Detection    | 4, 8| 1 |

Table 10. Summary of FedAvg settings.

**Image Transform.** For federated learning, it can not be assumed that clients know data distribution of other clients. So we use the average RGB value as $[0, 5, 0.5, 0.5]$ and standard deviation as $[0, 5, 0.5, 0.5]$, instead of them of all images.

**Layer-wise Learning** Both the models for image segmentation task have encoder-decoder architecture. For encoder layers, we employ ImageNet-1K pre-trained backbones, and hence they are to be just fine-tuned. The decoder layers are trained from scratch. To enforce this set-up, we modify our learning rates layer-wise so that the decoder layers get a learning rate 10 times more than the encoder layers.

7.2. More Experimental Results and Hyper-parameters

For image classification, we list all experiment results and the corresponding hyper-parameters in table 11, 12, 13.

As shown in figure 15, on CIFAR-100 dataset, the higher $\alpha$ makes training more difficult. The fine-tuning can increase the performance a lot.

Figure 4(d) (In main paper) and 14(a)(b) compare the effect of SGD and Momentum SGD. On CIFAR-100 dataset, when $\alpha = 0.5$ and with fine-tuning, Momentum SGD has similar performance with SGD. However, on the GLD-23K dataset, the performance of Momentum SGD is worse than SGD on both fine-tuning and without fine-tuning. One potential reason is that the accumulated gradients make local clients go too far in its local direction, which means that the diversity of the model parameters between different clients is increased by Momentum SGD. Here the local gradient accumulation is much different from the centralized training, in which the accumulated gradient has information of many data samples. But in federated learning, each client only has accumulated gradients with information of its own data samples. Maybe this could be addressed if we can develop some protocols to preserve the gradient information of other clients.

| Data     | Opt.   | Sched. | LR   | SFL. Acc | FT. Acc |
|----------|--------|--------|------|----------|---------|
| Cent.    | M SGD  | Linear | 0.003| 0.6011   | 0.593   |
|          |        |        | 0.01 | 0.6058   |         |
|          |        |        | 0.03 | 0.6055   |         |
|          |        |        | 0.1  |         | 0.215   |
| $a=0.1$  | M SGD  | No     | 0.003| 0.2239   | 0.4295  |
|          |        |        | 0.01 | 0.2306   | 0.2651  |
|          |        |        | 0.03 | 0.03025  | 0.37    |
| $a=0.5$  | M SGD  | Linear | 0.01 | 0.2384   | 0.3104  |
|          |        |        | 0.03 | 0.2469   | 0.2676  |
|          |        |        | 0.01 | 0.1358   | 0.1451  |
|          | SGD    | Linear | 0.03 | 0.1779   | 0.3203  |
| $a=100$  | M SGD  | No     | 0.003| 0.3045   | 0.529   |
|          |        |        | 0.01 | 0.1847   | 0.4479  |
|          |        |        | 0.03 | 0.2861   | 0.5366  |

Table 11. Summary of test accuracy on CIFAR-100 with EfficientNet-b0. In this table, Opt. represents optimizer, Sched. means learning rate scheduler, SFL. means scratch federated learning, i.e. not loading a pretrained model, and FT. means fine-tuning, i.e. loading a pretrained model. In other tables, we also use these abbreviations.

| Data     | Opt.   | Sched. | LR   | SFL. Acc | FT. Acc |
|----------|--------|--------|------|----------|---------|
| Cent.    | M SGD  | Linear | 0.003| 0.3619   |         |
|          |        |        | 0.01 | 0.5785   |         |
|          |        |        | 0.03 | 0.4375   |         |
|          |        |        | 0.1  | 0.2678   |         |
| $a=0.1$  | M SGD  | No     | 0.003| 0.2629   | 0.4276  |
|          |        |        | 0.01 | 0.2847   | 0.2959  |
|          |        |        | 0.03 | 0.2754   | 0.04848 |
| $a=0.5$  | M SGD  | No     | 0.003| 0.3411   | 0.3714  |
|          |        |        | 0.01 | 0.426    | 0.4691  |
|          |        |        | 0.03 | 0.4418   | 0.4412  |
| $a=100$  | M SGD  | No     | 0.003| 0.3594   | 0.5203  |
|          |        |        | 0.01 | 0.4507   | 0.4046  |
|          |        |        | 0.03 | 0.4764   | 0.5193  |

Table 12. Summary of test accuracy on CIFAR-100 with MobileNetV3.

Figure 5 (In main paper) and 14(c)(d) compare the effect of linear learning rate decay scheduler. On the GLD-23K dataset, when using fine-tuning, the performance between using the scheduler and not using is similar. However, when not using fine-tuning, using a scheduler cannot improve performance. But on CIFAR-100 dataset, when $\alpha = 0.1$
and not using fine-tuning, we can see the benefit from the scheduler. It would be a non-trivial problem of how to use the learning rate scheduler in FedAvg. And another interesting result of FedAvg is that the test accuracy may drop after some training iterations. The reason for this may be the diverse optimal model parameter of different clients. Maybe some new learning rate scheduler could solve this problem well.

7.2.1 Key Takeaways

**Fine Tuning.** From experiment results, it is obvious that fine-tuning can improve the performance of FedAvg a lot.

**Optimizer.** The experiment results show that in FedAvg, Momentum SGD cannot guarantee better performance than SGD. Because each client has non-I.I.D. dataset, the direction of gradients have high diversity. The accumulated gradients may be in the wrong direction of the current global model, making the model parameters deviate from the right way. So in federated learning, those optimizers that need an accumulation of gradients, like momentum SGD and RM-Sprop, may not be suitable in federated learning. Maybe we can develop more new optimizers for FedAvg.

**Learning Rate Scheduler.** Experiment results show that the performance of learning rate decay is various. FedAvg has much more global epochs than traditional training, which makes the learning rate decay a lot in late training stages. Thus, it is necessary to consider a suitable learning rate scheduler for federated learning.

**Image Transform.** For federated learning, it can not be assumed that clients know the data distribution of other clients. In such a case, how to decide on a suitable Image Transform is not a trivial problem. Moreover, this is not limited to computer vision but also happens in other areas.
under federated learning settings. Because it is hard to get
the global information of the data, we cannot directly do
normalization of them.

**Data Augmentation.** In recent years, there is a lot of
advanced data augmentation techniques in computer vision
proposed like AutoAugment [9], and GAN [18], and are
exploited in many advanced models like EfficientNet [67].
However, in federated learning, the whole data distribution
cannot be attained by the server or any client. In this case,
AutoAugment and GAN cannot be directly used in federated
learning. This may lead to a performance drop.

![Graph showing test accuracy of FedAvg with MobileNet on CIFAR-100 with different Non-IID degree.](image)

Figure 15. Test accuracy of FedAvg with MobileNet on CIFAR-100 with different Non-IID degree. Hyper-parameters of this figure can be found in table 12
| Conference   | Paper Title                                                                 | dataset                  | partition method   | model                                      | worker/device number |
|--------------|------------------------------------------------------------------------------|--------------------------|--------------------|--------------------------------------------|----------------------|
| ICML 2019    | Analyzing Federated Learning through an Adversarial Lens [2]                 | Fashion-MNIST            | natural non-IID    | 3 layer CNNs                               | 10                   |
|              |                                                                               | UCI Adult Census dataset | -                  | fully connected neural network             | 10                   |
| ICML 2019    | Agnostic Federated Learning [54]                                              | UCI Adult Census dataset | -                  | logistic regression                        | 10                   |
|              |                                                                               | Fashion-MNIST            | -                  | logistic regression                        | 10                   |
|              |                                                                               | Cornell movie dataset    | -                  | two-layer LSTM mode                        | 10                   |
|              |                                                                               | Penn TreeBank (PTB) data set | -                  | two-layer LSTM mode                        | 10                   |
| ICML 2019    | Bayesian Nonparametric Federated Learning of Neural Networks [87]             | MNIST                    | Dir(0.5)           | 1 hidden layer neural networks             | 10                   |
|              |                                                                               | CIFAR10                  | Dir(0.5)           | 1 hidden layer neural networks             | 10                   |
| ICML 2020    | Adaptive Federated Optimization [58]                                          | CIFAR-100                | Pachinko Allocation Method | ResNet-18                                     | 10                   |
|              |                                                                               | FEMNIST                  | natural non-IID    | CNN (2xconv)                              | 10                   |
|              |                                                                               | FEMNIST                  | natural non-IID    | Auto Encoder                              | 10                   |
|              |                                                                               | Shakespeare              | natural non-IID    | RNN                                       | 10                   |
|              |                                                                               | StackOverflow            | natural non-IID    | logistic regression                       | 10                   |
|              |                                                                               | StackOverflow            | natural non-IID    | 1 RNN LSTM                                | 10                   |
| ICML 2020    | FetchSGD: Communication-Efficient Federated Learning with Sketching [59]     | CIFAR-10/100             | 1 class / 1 client | ResNet-9                                   | -                    |
|              |                                                                               | FEMNIST                  | natural non-IID    | ResNet-101                                | -                    |
|              |                                                                               | PersonaChat              | natural non-IID    | GPT2-small                                | -                    |
| ICML 2020    | Federated Learning with Only Positive Labels [83]                            | CIFAR-10                 | 1 class / client  | ResNet-8/32                                | -                    |
|              |                                                                               | CIFAR-100                | 1 class / client  | ResNet-56                                  | -                    |
|              |                                                                               | AmazonCAT                | 1 class / client  | Fully Connected Nets                       | -                    |
|              |                                                                               | WikiLSHTC               | 1 class / client  | -                                         | -                    |
|              |                                                                               | Amazon670K               | 1 class / client  | -                                         | -                    |
| ICML 2020    | SCAFFOLD: Stochastic Controlled Averaging for Federated Learning [38]        | EMNIST                   | 1 class / 1 client | Fully connected network                    | -                    |
| ICML 2020    | From Local SGD to Local Fixed-Point Methods for Federated Learning [50]      | a9a(LIBSVM)              | -                  | Logistic Regression                        | -                    |
|              |                                                                               | a9a(LIBSVM)              | -                  | Logistic Regression                        | -                    |
| ICML 2020    | Acceleration for Compressed Gradient Descent in Distributed and Federated Optimization [15] | a5a                       | -                  | logistic regression                        | -                    |
|              |                                                                               | mushrooms               | -                  | logistic regression                        | -                    |
|              |                                                                               | a9a                      | -                  | logistic regression                        | -                    |
|              |                                                                               | w6a LIBSVM               | -                  | logistic regression                        | -                    |
| ICLR 2020    | Federated Learning with Matched Averaging [73]                               | CIFAR-10                 | -                  | VGG-9                                      | 16                   |
|              |                                                                               | Shakespeare              | sampling 66 clients | 1-layer LSTM                               | 66                   |
| ICLR 2020    | Fair Resource Allocation in Federated Learning [12]                          | Synthetic dataset use LR | natural non-IID    | multinomial logistic regression            | 10                   |
|              |                                                                               | Vehicle                  | natural non-IID    | SVM for binary classification              | 10                   |
|              |                                                                               | Shakespeare              | natural non-IID    | RNN                                        | 10                   |
|              |                                                                               | Sent140                  | natural non-IID    | RNN                                        | 10                   |
| ICLR 2020    | On the Convergence of FedAvg on Non-IID Data [44]                            | MNIST                    | natural non-IID    | logistic regression                        | 10                   |
|              |                                                                               | Synthetic dataset use LR | natural non-IID    | logistic regression                        | 10                   |
| ICLR 2020    | DBA: Distributed Backdoor Attacks against Federated Learning [79]             | Lending Club Loan Data   | -                  | 3 FC                                       | 10                   |
|              |                                                                               | MNIST                    | -                  | 2 conv and 2 fc                            | 10                   |
|              |                                                                               | CIFAR-10                 | -                  | lightweight Resnet-18                      | 10                   |
|              |                                                                               | Tiny-Imagenet            | -                  | Resnet-18                                  | 10                   |
| ML Sys 2020  | Federated Optimization in Heterogeneous Networks [62]                        | MNIST                    | natural non-IID    | multinomial logistic regression            | 10                   |
|              |                                                                               | FEMNIST                  | natural non-IID    | multinomial logistic regression            | 10                   |
|              |                                                                               | Shakespeare              | natural non-IID    | RNN                                        | 10                   |
|              |                                                                               | Sent140                  | natural non-IID    | RNN                                        | 10                   |

*Note: we will update this list once new publications are released.*