FedOCR: Efficient and Secure Federated Learning for Scene Text Recognition

Wenqing Zhang[1], Yang Qiu[1], Song Bai[1], Rui Zhang[2], Xiaolin Wei[2], Xiang Bai[1]

[1]Huazhong University of Science and Technology, [2]Meituan
{wenqingzhang, yqiu, xbai}@hust.edu.cn, songbai.site@gmail.com, {zhangrui36, weixiaolin02}@meituan.com

Abstract

While scene text recognition techniques have been widely used in commercial applications, data privacy has rarely been taken into account by this research community. Most existing algorithms have assumed a set of shared or centralized training data. However, in practice, data may be distributed on different local devices that can not be centralized to share due to privacy restrictions. In this paper, we study how to make use of decentralized datasets for training a robust scene text recognizer while keeping them stay on local devices. To the best of our knowledge, we propose the first framework leveraging federated learning for scene text recognition, which is trained with decentralized datasets collaboratively. Hence we name it FedOCR. To make FedCOR fairly suitable to be deployed on end devices, we make two improvements including using lightweight models and hashing techniques. We argue that both are crucial for FedOCR in terms of communication efficiency and security for federated learning. The simulations on decentralized datasets show that the proposed FedOCR achieves competitive results to the models that are trained with centralized data, with fewer communication costs and higher-level privacy-preserving.

Introduction

Text in scene images contains valuable semantic information for text reading and has become one of the most popular research topics in academia and industry for a long time (Goel et al. 2013; Almazán et al. 2014; Su and Lu 2014; Luo, Jin, and Sun 2019; Li et al. 2019; Zhang et al. 2020; Yu et al. 2020). In practice, scene text recognition has been applied to various real-world scenarios, such as autonomous navigation, photo transcription, and scene understanding. With the development of deep learning and the emergence of public text datasets, significant progress on scene text recognition has been made in recent years.

However, most of the existing scene text recognition algorithms assume that a large scale set of training images is easily accessible. As shown in Fig. 1(a), in real conditions, algorithms may achieve sub-optimal performance and be unable to model the data variations or diversity owing to the lack of sufficient images. To remedy this, some works (Bartz et al. 2019; Hu et al. 2020) merge different public datasets to build a more robust text recognizer, as illustrated in Fig. 1(b). However, centralizing data in this way is simply problematic in **many real-world scenarios**. For example, many laws and regulations strengthening the data privacy constrain the use of data stored on local devices, such as General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche 2017). Besides, centralizing tremendous image data from different local devices incurs heavy communication loads. That means it is simply intractable to centralize large amounts of data for scene text recognition training in practice. Our solution, which works within the framework of federated learning, is illustrated in Fig. 1(c).

Federated Learning (FL), a new concept first proposed by McMahan *et al.* (McMahan et al. 2016), allows data owners to train a shared model collaboratively while keeping data stored on different local devices. However, directly applying FL to scene text recognition faces two inevitable difficulties. First, in most scene text recognition algorithms, a heavyweight backbone model is usually adopted for the sake of better performance. Hence, it results in heavy burdens of the parameter transmission while doing federated learning. Second, there is an extra computational cost from a privacy-preserving module to handle privacy leakage due to the honest-but-curious global server in general federated learning frameworks.

In this paper, to the best of our knowledge, we propose the first federated learning framework for scene text recognition, which we name FedOCR. In our FedOCR (a schematic is given in Fig. 2), all participants train a shared model collaboratively without centralizing the training images. In this manner, datasets on different local devices have an indirect influence on the training of the global model, which leads to a competitive performance to the model trained with a centralized set of data. To improve the communication efficiency between the global server and local clients, we argue two important aspects in FedOCR, *i.e.*, lightweight models and hashing techniques. Moreover, benefited from the hashing technique, we can avoid privacy leakage to the global server by a specific hashing function and the random seeds, which saves an extra computational cost for a privacy-preserving module. As a consequence, the proposed FedOCR is readily to be deployed in practical applications for scene text recognition.

Compared with existing scene text recognition meth-

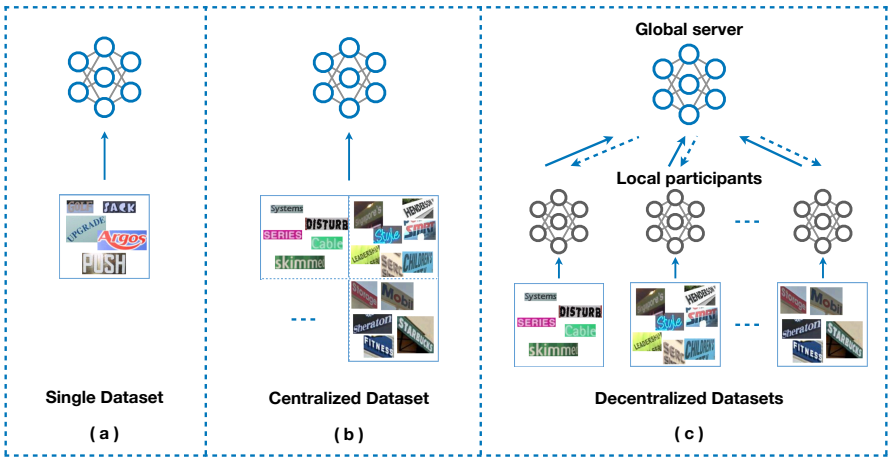

Figure 1: An illustration of training scene text recognizers with (a) a single dataset, (b) a centralized dataset from different devices, and (c) decentralized datasets distributed on different local devices.

ods (Luo, Jin, and Sun 2019; Li et al. 2019; Bartz et al. 2019; Zhan and Lu 2019; Yue et al. 2020; Bhunia et al. 2021) without federated learning, the proposed framework has the following intriguing merits. First, FedOCR can make use of more abundant image data from different local devices. Particularly, there are billions of end devices with tremendous text images benefiting scene text recognition. Therefore, our framework may have great potential in real-world applications of scene text reading. Second, by design, our framework has a superior trade-off between parameter transmission efficiency and performance. The proposed text recognizer has much fewer parameters than existing scene text recognition algorithms but encouragingly reaches a comparable performance. Last, it can encrypt and decrypt with the hashing technique, which provides higher-level privacy-preserving without an extra computational cost.

In summary, the main contributions of this paper are three-fold.

- We reveal the problem of data privacy in scene text recognition, which is somehow overlooked by the existing methods.
- We propose the first federated scene text recognition framework called FedOCR for training a recognizer with decentralized datasets distributed on different local devices.
- FedOCR is a highly communication-efficient as well as privacy-preserving framework by incorporating lightweight backbones and hashing techniques, which makes it suitable to be deployed in real privacy-sensitive applications and edge devices.

Related Work

Scene text recognition has attracted great interest for a long time. According to Long *et al.* (Long, He, and Yao 2018), representative methods can be roughly divided into two mainstreams, *i.e.*, Connectionist Temporal Classification (CTC) based and attention-based methods. Generally, the CTC-based methods model scene text recognition as a sequence recognition task. For example, Shi *et al.* (Shi, Bai, and Yao 2016) combine the convolutional neural network (CNN) with the recurrent neural network (RNN) to extract sequence features from input images, and decode the features with a CTC layer. Different from Shi *et al.* (Shi, Bai, and Yao 2016), Gao *et al.* (Gao et al. 2019) use stacked convolutional layers to extract contextual information from inputs without RNN, and show advantages with low computational costs. Zhang *et al.* (Zhang, Gupta, and Zisserman 2020) turn text recognition into a visual matching problem by exploiting the repetition of glyphs in language, and build this similarity between units into the proposed architecture. Meanwhile, attention-based methods extract features more effectively via the attention mechanism. For instance, Liu *et al.* (Liu et al. 2018) propose a binary convolutional encoder-decoder network to provide real-time scene text recognition. Liu *et al.* (Liu, Chen, and Wong 2018) propose a character-aware neural network with a hierarchical attention mechanism, which adopts a local transformation to rectify characters individually. Unlike other attention-based algorithms, Bai *et al.* (Bai et al. 2018) propose Edit Probability (EP) to handle the misalignment between the output sequence of probability distribution and the ground-truth sequence. Nguyen *et al.* (Nguyen et al. 2021) incorporate a dictionary in both the training and inference stage to make a more robust scene text recognition system.

Undoubtedly, large amounts of real-world data are needed in practical applications of those scene text recognition methods. However, tremendous image datasets are distributed on different companies, communities or local devices, and can not be centralized to share. To handle this problem, McMahan *et al.* (McMahan et al. 2016) first propose the concept of Federated Learning (FL) to train deep networks from decentralized data collaboratively. Following McMahan *et al.* (McMahan et al. 2016), many researchers

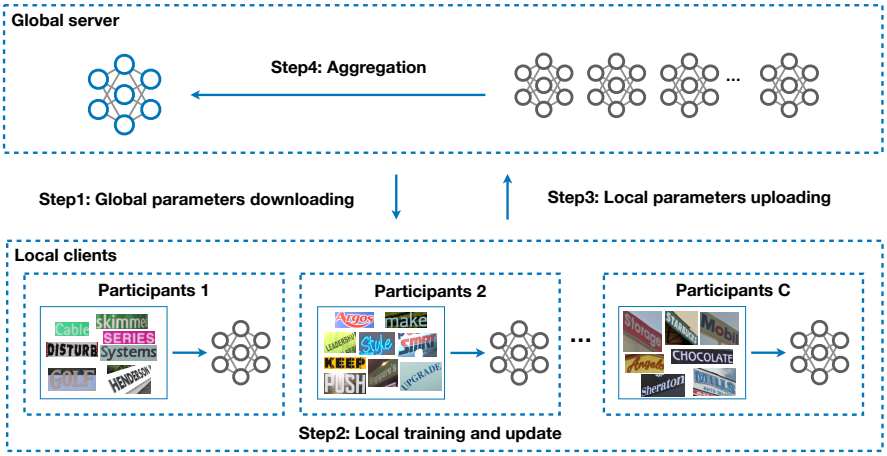

Figure 2: The pipeline of our federated scene text recognition framework.

are working on improving the federated learning with more efficient parameter transmission and higher-level privacy-preserving. To improve privacy security, Wei *et al.* (Wei et al. 2019) propose a federated learning framework based on differential privacy, in which artificial noises are added to the local parameters of participants before the model aggregation. Sun *et al.* (Sun et al. 2021) develop an effective defense called Soteria by perturbing data representations to against model inversion attack in FL. To improve communication efficiency, Reisizadeh *et al.* (Reisizadeh et al. 2019) propose a communication-efficient federated learning method with periodic averaging and quantization. Gao *et al.* (Gao, Xu, and Huang 2021) propose the error-compensated double compression mechanism to significantly reduce the communication cost. **Especially very recently, the computer vision community starts to pay attention to federated learning, thus arising several pioneering works.** For example, Luo *et al.* (Luo et al. 2019) implement object detection algorithms with federated learning and release a reliable benchmark framework. In the medical field, Zhu *et al.* (Zhu et al. 2019) implement a privacy-preserving federated learning system with the differential privacy for brain tumor segmentation. Li *et al.* (Li, He, and Song 2021) conduct contrastive learning in model-level to correct the local training of individual parties to achieve high performance in image classification tasks. To the best of our knowledge, we propose the first federated scene text recognition framework, which is more efficient in communication and provides higher-level privacy-preserving.

Methodology

In this section, we first introduce the pipeline of our federated scene text recognition framework. Then, we describe the details of local training and global aggregation, which are the two main steps in federated learning. Finally, we elaborate on how to improve communication efficiency and preserve data privacy in our framework.

Pipeline of FedOCR

According to Yang *et al.* (Yang et al. 2019), our framework is a kind of horizontal federated learning, where datasets of different participants share the same feature space but differ in samples. Suppose we have C data owners, which have different sets of training images $\{D_1, ..., D_C\}$. We denote the accuracy of the text recognizer trained with decentralized datasets $\{D_1, ..., D_C\}$ as Acc_{FED}. Note that these decentralized datasets are not shared or transferred to other participants during training. We denote the accuracy of the text recognizer trained with a centralized dataset $D = D_1 \cup ... \cup D_C$ as Acc_{SUM}. Basically, the objective of FedOCR is to minimize the difference between Acc_{FED} and Acc_{SUM}. A smaller difference between Acc_{FED} and Acc_{SUM} means a better performance of our federated learning for scene text recognition.

Fig. 2 illustrates the pipeline of our federated scene text recognition framework. There are C participants, each of which has a set of data containing cropped text word images and transcriptions, and a global server for local model parameter aggregation. We assume all participants agree in advance on the same network architecture and the same training objective but do not share their datasets. The whole learning process can be decomposed into four steps:

(1) Before each round of local training, all participants start with the same parameters, which are initialized randomly in the first round and downloaded from the global server in the next rounds.

(2) Each participant trains the model with its dataset for E_l epochs individually.

(3) All participants calculate parameter increments compared to the original parameters in a round, and all parameter increments are sent to the global server.

(4) The global server aggregates all parameter increments by average, and updates a set of global parameters. Before the next local training, the global parameters are downloaded for local model updating.

Algorithm 1: Local Training

Input: Latest global parameters W_t^{global} in round t; Local training learning rate $\eta^i, i \in [0, C-1]$

for each $i \in [0, C-1]$ **do**
    Overwrite local weight vectors: $W_t^i = W_t^{global}$
end for
for all local participant $i \in \{0, 1, ..., C-1\}$) **do**
    for $e \in [0, E_l - 1]$ **do**
        for $s \in [0, step_{max}]$ **do**
            Sample a minibatch B_s
            Compute gradients: $g_t^i = \bigtriangledown L(B_s; W_t^i)$
            Update local parameters: $W_t^i = W_t^i - \eta^i \cdot g_t^i$
        end for
    end for
    Compute local parameter increments:
    $\Delta W_t^i = W_t^i - W_t^{global}$
    Send ΔW_t^i and data size S_i to the global server
end for

Following this pipeline, our federated training continues until convergence.

Local Training. In our FedOCR, each participant i and the global server maintain a set of local model parameters W^i and W^{global}, respectively. Algorithm 1 describes the local training process of our framework. As shown, all participants first download the latest global parameters from the global server and overwrite their local parameters. Then, participants train local models with their datasets independently for E_l epochs and send parameter increments to the global server. During local training, all participants do not share any image data with others. To update the global parameters efficiently, all participants should train their models enough before parameter transmission. McMahan *et al.* (McMahan et al. 2016) demonstrate that sufficient epochs of local training can bring a dramatic increase in parameter update efficiency. Detailed experiment settings of our FedOCR are provided in the next section.

Global Aggregation. To aggregate parameter increments from different local participants, McMahan *et al.* (McMahan et al. 2016) propose a straightforward approach to aggregate all local participants' parameters by average. Following steps in Algorithm 2, we adapt the federated average method (McMahan et al. 2016) to our federated scene text recognition framework. In the global aggregation step of our FedOCR, we average all parameter increments and update former global parameters, which are available for all participants' downloading.

Communication Efficiency

Communication efficiency is an essential property in federated learning. For instance, if the size of one participant's model is one hundred megabytes, tens of gigabytes will be required to transmit in a round, when hundreds of clients participate in a federated learning framework. Under such a circumstance, plenty of parameters result in huge communication costs, which lead to a training bottleneck. To reduce

Algorithm 2: Global Aggregation

Input: All local parameter increments $\{\Delta W_t^i | i \in [0, C-1]\}$ in round t; Local data size $\{S_i | i \in [0, C-1]\}$ Global parameters W_t^{global};

Compute global parameter increments:
$\Delta W_t^{global} = (\ \sum_{i=0}^{C-1} S_i \Delta W_t^i\ )\ /\ \sum_{i=0}^{C-1} S_i$
Update global parameters:
$W_{t+1}^{global} = W_t^{global} + \Delta W_t^{global}$
Send W_{t+1}^{global} to all participants

Algorithm 3: Hashing Technique

Input: Compression ratio γ; Hashing seeds $\{seed^l | l \in [0, L-1]\}$, where L is the number of network layers;
Output: A compressed network;

for each layer l in the entire network **do**
    Assume the total size of weight matrix W^l is T^l
    Generate a real weight vector R^l with a size $T^l * \gamma$
    Generate a random sort RS^l of numbers from 0 to $T^l - 1$ with a hashing function and a seed $seed^l$
    Generate an index vector I^l: $[\lfloor e \cdot \gamma \rfloor$, for e in $RS^l]$
    Reshape I^l as the shape of W^l
    Generate a virtual weight matrix: $V^l = R^l[I^l]$
end for
Initialize our network with $R^l, l \in [0, L-1]$, and the total parameter size is compressed to $\gamma \cdot \sum_{l=0}^{L-1} T^l$

communication burdens, we replace the heavyweight backbone, such as ResNet (He et al. 2016), for feature extraction in text recognizers with a lightweight neural network. To further decrease the parameter size, we extend a hashing technique (Chen et al. 2015) to compress the parameters of CNN and RNN, which makes it applicable for any text recognizer. In this way, the text recognizer in our FedOCR has much fewer parameters compared with existing text recognition algorithms, which shows great potential in practical federated learning deployment.

Hashing Technique. In fact, any well-designed scene text recognition model can be applied in our federated learning framework. However, considering the communication efficiency, the network with fewer parameters is more appropriate and practical. Therefore, we propose to compress model parameters by a hashing technique. Specifically, we compress network parameters in a weight sharing manner that a random subset of parameters in a layer share the same parameter. Following Algorithm 3, we can compress the parameters in a scene text recognition network with a hyperparameter γ to control the compression ratio, and it can reduce the parameter size to a large extent. It should be noted that $\lfloor e \cdot \gamma \rfloor$ means the largest integer that is smaller than $e \cdot \gamma$ in Algorithm 3. Notably, the specific hashing function and the random seeds are shared among all local participants to keep the same relationship between real weight vectors and virtual weight matrices of all local models.

Text Recognizer. Following the above methods, we can improve any existing text recognition algorithms to construct a lightweight text recognizer. Specifically, in our experiments, we optimize a classical text recognizer, ASTER (Shi et al. 2018). We replace the encoder in ASTER with ShuffleNetV2 (Ma et al. 2018) and apply the hashing technique to the entire model parameters. Benefited from hashing techniques and lightweight networks, we successfully decrease communication costs to a large extent in our federated learning framework.

Moreover, we keep the network structure and experiment settings the same with ASTER as much as possible. Similar to ASTER, the text recognizer in our experiments consists of a rectification network, a lightweight convolutional encoder, and an attentional sequence-to-sequence model. We briefly introduce the method of scene text recognition as follows: Firstly, an input image is rectified by a rectification network before being sent into a recognition network. The rectification network based on the Spatial Transformer Network (STN) aims to rectify perspective or curved texts. Secondly, we use a lightweight neural network as the encoder to extract the feature sequence from the rectified image. Lastly, we use an attentional sequence-to-sequence model as the decoder to translate the feature sequence. During inference, we use beam searching by holding five candidates with the highest accumulative scores at every step.

Network Training. After neural network initialization, the mapping relationship between real weight vectors and virtual weight matrices is fixed, which is defined in Algorithm 3. In the forward computation, it is the virtual weight matrices that participate in calculation with input features. In the backward propagation, the gradients of all parameters in real weight vectors are calculated based on virtual weight matrices' parameter gradients.

Let $V^l_{i,j}$ denote the i-th row and j-th column element of a virtual weight matrix at layer l, and let R^l_k denote the k-th element in the corresponding real weight vector. Assuming that

$$\frac{\partial \mathcal{L}}{\partial V^l_{i,j}} = g^l_{i,j}, \tag{1}$$

where $g^l_{i,j}$ is computed from the loss. Moreover,

$$\frac{\partial V^l_{i,j}}{\partial R^l_k} = \mathbb{I}(I^l[i,j],k), \text{ and } \mathbb{I}(a,b) = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Based on the above equations, we can obtain any parameter's gradient in the real weight vector as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial R^l_k} &= \sum_i \sum_j \frac{\partial L}{\partial V^l_{i,j}} \cdot \frac{\partial V^l_{i,j}}{\partial R^l_k} \\ &= \sum_i \sum_j g^l_{i,j} \cdot \mathbb{I}(I^l[i,j],k). \end{aligned} \tag{3}$$

Privacy Preserving

Federated learning can provide training procedures at a high level of security, but the global server still has a chance to compromise data privacy, such as model inversion (Fredrikson, Jha, and Ristenpart 2015) and GAN-based attacks (Hitaj, Ateniese, and Perez-Cruz 2017). Usually, local network parameters or their increments are sent to the global server in each communication round, which gives the honest-but-curious server a chance to spy on local sets of data. In recent works, Geiping *et al.* (Geiping et al. 2020) and Phong *et al.* (Phong et al. 2018) show that the gradients may reveal information of training samples and apply an additively homomorphic encryption scheme to their federated framework. Shokri *et al.* (Shokri and Shmatikov 2015) propose to upload partially gradients added with noise to avoid information leakage, and apply differential privacy to parameter updates for a higher level of security. However, the above methods bring more computational costs or a dramatic decrease in accuracy because of the privacy-preserving module.

In our FedOCR, we adopt the hashing technique to compress the entire model parameters with a hashing function and random seeds. They are equivalent to an encryption-decryption module and the keys, but it can save much computational costs without the encryption-decryption step. For the parameter aggregation in the global server, we only upload increments of the parameters in real weight vectors, which can not be used to reconstruct the complete network without the specific hashing function and the random seeds. As for all local participants, they share the same hashing function and random seeds, so the average operation in the global aggregation can be directly applied to these parameter increments. Therefore, the global server can not compromise the private data, while it can finish its global aggregation task. Moreover, the other attackers also can not do anything with the compressed parameter increments, because they mean nothing without hashing functions and seeds. In this way, we enhance the privacy-preserving in our FedOCR without introducing an extra computational cost.

Experiments

Datasets

Two synthetic datasets (Jaderberg et al. 2014; Gupta, Vedaldi, and Zisserman 2016) and six public real-world datasets are used to train local models, and our models are evaluated on seven general datasets. In our federated settings, we construct different local datasets with the public real-world datasets. These datasets are briefly introduced as follows:

Synth90k (Jaderberg et al. 2014) contains 9 million images generated from a set of 90k English words. Words are rendered onto natural images with random transformations and effects.

SynthText (Gupta, Vedaldi, and Zisserman 2016) contains 0.8 million images for end-to-end text detection and recognition tasks. Therefore, we crop word images using the ground-truth word bounding boxes.

ICDAR 2003 (IC03) (Lucas et al. 2005) contains 860 cropped word images for evaluation after discarding images that contain non-alphanumeric characters or have fewer than

three characters, which follows (Mishra, Alahari, and Jawahar 2012). For training, we use 1150 cropped images after filtering.

ICDAR 2013 (IC13) (Karatzas et al. 2013), which inherits most images from IC03 and extends it with new images, contains 1015 cropped word images for evaluation after filtering. For training, we use 848 cropped images after filtering.

ICDAR 2015 (IC15) (Karatzas et al. 2015) contains images captured by a pair of Google Glasses casually, and many images are severely distorted or blurred. For a fair comparison, we evaluate models on 1811 cropped word images after filtering. For training, we use 4426 cropped images after filtering.

IIIT5K-Words (IIIT5K) (Mishra, Alahari, and Jawahar 2012) contains 3000 word images collected for evaluation and 2000 word images for training, which are mostly horizontal text images.

Street View Text (SVT) (Wang, Babenko, and Belongie 2011) is collected from the Google Street View, and it contains 647 images of cropped words, many of which are severely corrupted by noise, blur, or low resolution.

Street View Text Perspective (SVTP) (Quy Phan et al. 2013), which is collected from Google StreetView and contains many distorted images, contains 645 word images for evaluation.

CUTE80 (CUTE) (Risnumawan et al. 2014) contains 80 real-world curved text images with high quality. For evaluation, we crop 288 word images according to its ground-truth.

ArT (Chng et al. 2019) is a combination of Total-Text, SCUT-CTW1500, and Baidu Curved Scene Text, which contains images with arbitrary-shaped texts. For training, we use 30271 word images after discarding images that contain non-alphanumeric characters and vertical texts.

COCO-Text (Veit et al. 2016) is based on the MS COCO dataset, which contains images of complex everyday scenes. For training, we use 31943 cropped images after discarding images that contain non-alphanumeric characters and vertical texts.

Experiment settings

Decentralized Datasets for Federated Learning Different local datasets are constructed by public real-world datasets in our experiment settings. We use the training images from IC03 (Lucas et al. 2005), IC13 (Karatzas et al. 2013), IC15 (Karatzas et al. 2015), IIIT5K (Mishra, Alahari, and Jawahar 2012), ArT (Chng et al. 2019), and COCO-Text (Veit et al. 2016). As a sequence, we have 70638 real-word text images in total. To simulate the decentralized datasets distributed on local devices in federated learning, we, as an honest server, should not known the data distribution and whether there is a data bias. Hence, we randomly split all training images into different sets of image data for C participants, It should be mentioned that these different sets of image data should not be shared or transferred to other participants during the training procedures.

Federated Settings. Some hyper-parameters should be noted in our federated settings: C, the number of participants in our federated scene text recognition framework; γ, the compression ratio of the hashing technique; E_l, the number of epochs that each local participant trains the model with its dataset before communication with the global server; B, the batch size in local training. In our experiments, we set $C = 5$, $E_l = 3$, $B = 512$ and $\gamma \in \{1/2, 1/4, 1/8\}$.

Baseline and FedOCR-Hash. In our experiments, we adopt ASTER[1] (Shi et al. 2018) as the text recognition baseline in our FedOCR, which is denoted as ASTER-FL. Then, we replace the encoder in ASTER-FL with ShuffleNetV2 (Ma et al. 2018), and this variant of ASTER-FL in our FedOCR is denoted as FedOCR-Hash$_1$. To further reduce the parameter size, we apply the hashing technique to compress FedOCR-Hash$_1$ with different ratios $\gamma \in \{1/2, 1/4, 1/8\}$, and these models are denoted as FedOCR-Hash$_\gamma$ in the following paper.

Implementation Details. Following the federated settings, we construct $C = 5$ participants in our FedOCR for experiments. In each local training, all models are locally trained via Adadelta (Zeiler 2012) with an initialization learning rate of 1.0, and each participant trains the scene text recognition model with its dataset individually for $E_l = 3$ epochs in each round. All word images are trained directly without data argumentation. As for the complete federated training process of our FedOCR, each participant trains its model with the two synthetic datasets for 4 rounds, then trains on its real-world dataset for 40 rounds.

The learning rate is decayed to 0.1 and 0.01 at the 5-th round and the 30-th round, respectively. Following Algorithm 2, in the global aggregation step, the global server aggregates the parameter increments from all participants by average. To simply simulate the communication procedure of federated learning, we replace the parameter transmission between participants and the global server with saving and restoring checkpoints on the hard-disk.

Evaluation Metric. In our experiments, we use the case-insensitive word accuracy for evaluation. If the word prediction and the ground-truth are the same in the lower case, the prediction is correct. The recognition accuracy is the percentage of the correct number of total. Furthermore, the objective of FedOCR is to minimize the difference between the accuracy of the text recognizer trained with decentralized datasets and trained with a centralized dataset. A smaller difference means a better performance of our FedOCR.

Experiments on FedOCR

In this subsection, we first compare the parameter reduction and the accuracy decrease of different models in our FedOCR. Then, we analyze the performance of our FedOCR compared with the other two training manners and show that

[1]https://github.com/ayumiymk/aster.pytorch

Models	Backbone	γ	Param. (M)	Model (MB)	Accuracy (%)
ASTER-FL	ResNet	-	20.99	80.52	91.94
FedOCR-Hash$_1$	ShuffleNetV2	-	13.34 ($\downarrow$ 36.45%)	51.37 ($\downarrow$ 36.20%)	89.08 ($\downarrow$ 3.11%)
FedOCR-Hash$_{1/2}$	ShuffleNetV2	1/2	6.70 ($\downarrow$ 68.08%)	26.05 ($\downarrow$ 67.65%)	86.65 ($\downarrow$ 5.75%)
FedOCR-Hash$_{1/4}$	ShuffleNetV2	1/4	3.38 ($\downarrow$ 83.90%)	13.38 ($\downarrow$ 83.38%)	85.39 ($\downarrow$ 7.12%)
FedOCR-Hash$_{1/8}$	ShuffleNetV2	1/8	1.72 ($\downarrow$ 91.81%)	7.05 ($\downarrow$ 91.24%)	82.58 ($\downarrow$ 10.18%)

Table 1: Parameter size and accuracy comparison between different models in our FedOCR. The accuracy is the average result of all testing datasets. The models size refers to the storage occupied on the hard-disk. γ is the compression ratio of the hashing technique, and "$\gamma = -$" means that we do not apply the hashing technique to the model. The reduction percentages of parameter size, model size, and accuracy compared with ASTER-FL are shown in parentheses respectively.

Models	Training	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE
ASTER-FL	single	93.7	89.0	93.7	93.8	80.6	82.3	85.4
	centralized	95.0	91.7	95.3	94.6	82.2	83.3	91.7
	federated	95.0	90.7	94.8	94.0	82.0	82.3	91.0
FedOCR-Hash$_1$	single	90.8	83.0	90.9	89.4	77.3	77.5	82.6
	centralized	93.1	86.4	92.5	92.2	79.7	80.6	86.8
	federated	92.9	86.9	92.0	91.7	79.4	80.8	86.5
FedOCR-Hash$_{1/2}$	single	89.2	83.0	90.2	88.5	75.3	73.8	77.8
	centralized	91.6	83.6	91.0	90.3	77.9	75.5	82.3
	federated	91.2	84.2	91.6	90.7	77.5	76.0	82.6
FedOCR-Hash$_{1/4}$	single	87.2	79.1	87.1	86.1	73.4	71.5	77.4
	centralized	89.4	81.6	89.5	88.8	75.9	74.3	81.6
	federated	89.0	81.8	89.3	89.2	76.3	75.2	81.6
FedOCR-Hash$_{1/8}$	single	83.5	74.8	84.8	81.4	70.2	71.2	73.3
	centralized	86.7	78.8	86.7	86.0	72.3	71.6	79.5
	federated	86.6	80.1	87.1	85.4	72.4	71.6	79.9

Table 2: Recognition accuracy in different training manners. "single": The model is trained only with one participant's dataset; "centralized": The model is trained with a centralized set of image data; "federated": The global model is trained with decentralized sets of image data in a federated manner. The detailed structures of different FedOCR-Hash are shown in Tab. 1.

our FedOCR achieves the objective of federated learning. Finally, we evaluate the two improvements in communication efficiency of our FedOCR.

Comparison of Parameter Size and Accuracy. Tab. 1 shows the parameter size and model size of different models in our FedOCR. The accuracy is the average result of all testing datasets. The models size refers to the storage occupied on the hard-disk. Compared with ASTER-FL, FedOCR-Hash$_1$ reduce 36.45% parameter size, but there is only a 3.11% accuracy decrease. As for different FedOCR-Hash$_\gamma$ in our experiments, FedOCR-Hash$_{1/4}$ with an appropriate compression ratio γ achieves a 83.90% reduction in parameter size and drops only 7.12% in accuracy. Improved by the lightweight backbone and the hashing technique, the model size of the scene text recognizers in our FedOCR reduces to a large extent, and these lightweight text recognizers encouragingly reach a comparable performance.

Federated Learning for Scene Text Recognition. Tab. 2 shows the detailed results on all testing datasets of ASTER-FL and different FedOCR-Hash$_\gamma$ in three manners of training. First, "single" training means that the model is trained only with one participant's dataset. Second, "centralized" training means that the model is trained with a centralized set of image data. Third, "federated" training means that the model is trained with decentralized sets of image data in a federated manner. As shown in Tab. 2, "federated" and "centralized" training results of all models are similar to each other and better than "single" training results. In the "single" training manner, scene text recognition faces the problem in practice that the image data for training is limited, which causes poor performance in scene text recognition. However, we succeed in training a shared model with decentralized sets of image data collaboratively in the "federated" training manner, and we do not exchange or expose any image data to other participants. Expectantly, our FedOCR achieves comparable results, which are very close to the results of the "centralized" training manner. Therefore, our FedOCR is effective to train a more robust model without centralizing datasets on different local devices.

Communication Efficiency Improvement. In Tab. 2, FedOCR-Hash$_1$ shows comparable accuracy with ASTER-FL in the "federated" training manner. Owing to the lightweight backbone in FedOCR-Hash$_1$, it has fewer parameters than ASTER-FL, which benefits communication efficiency in federated learning. As shown in Fig. 3, FedOCR-Hash$_1$ has a higher accuracy than ASTER-FL when little communication bytes are uploaded.

Fig. 3 illustrates the accuracy curves of different models on IIIT5k versus uploaded bytes in federated training procedures. FedOCR-Hash$_\gamma$ with a smaller compression ratio γ achieves higher accuracy when limited communication

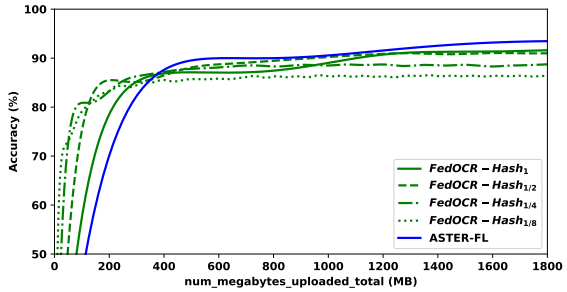

Figure 3: Accuracy on IIIT5k versus number of uploaded megabytes of different models with limited transmitted bytes in federated learning.

bytes are uploaded, and it shows greater advantages in communication efficiency. The advantage of our FedOCR-Hash$_\gamma$ will be more distinctive when more local clients participate in our FedOCR. Considering both Tab. 1 and 2, FedOCR-Hash$_{1/4}$ with an appropriate compression ratio γ shows a significant overall performance in communication efficiency and accuracy of federated learning. Only 13.38 megabytes are required to be transmitted by each participant, which results in a faster parameter transmission with the same communication bandwidth.

Benefited from lightweight models and hashing techniques, our federated scene text recognition framework shows a comparable performance and advantages in communication efficiency. Considering plenty of participants and the unstable data transmission network in the real world, our FedOCR has great potential in practical application deployment.

Conclusion and Future Work

In this paper, we reveal the problem of data privacy in scene text recognition and address the difficulty in utilizing decentralized datasets distributed on local devices with federated learning. To the best of our knowledge, we propose the first federated scene text recognition framework named FedOCR. In our FedOCR, we succeed in training a shared text recognizer collaboratively with decentralized datasets and avoid violating rules of data privacy. Benefited from lightweight models and hashing techniques, we reduce communication costs to a large extent and provide higher-level privacy-preserving against the honest-but-curious global server. In terms of taking advantage of tremendous decentralized real-world data in practice, our communication-efficient federated learning framework for scene text recognition shows intriguing merits.

Recently, the domain shift in scene text recognition has attracted great interest in academia, and some methods are proposed, such as GA-DAN (Zhan, Xue, and Lu 2019) and SSDAN (Zhang et al. 2019). Notably, the domain shift occurs in federated learning for scene text recognition as well, which leads to a deterioration on the global accuracy. Hence, we are working on the domain adaptation of decentralized datasets within our framework.

References

Almazán, J.; Gordo, A.; Fornés, A.; and Valveny, E. 2014. Word spotting and recognition with embedded attributes. *TPAMI*, 36(12): 2552–2566.

Bai, F.; Cheng, Z.; Niu, Y.; Pu, S.; and Zhou, S. 2018. Edit probability for scene text recognition. In *CVPR*.

Bartz, C.; Bethge, J.; Yang, H.; and Meinel, C. 2019. KISS: Keeping It Simple for Scene Text Recognition. *arXiv preprint arXiv:1911.08400*.

Bhunia, A. K.; Sain, A.; Kumar, A.; Ghose, S.; Chowdhury, P. N.; and Song, Y.-Z. 2021. Joint Visual Semantic Reasoning: Multi-Stage Decoder for Text Recognition. In *ICCV*, 14940–14949.

Chen, W.; Wilson, J.; Tyree, S.; Weinberger, K.; and Chen, Y. 2015. Compressing neural networks with the hashing trick. In *ICML*.

Chng, C.-K.; Liu, Y.; Sun, Y.; Ng, C. C.; Luo, C.; Ni, Z.; Fang, C.; Zhang, S.; Han, J.; Ding, E.; et al. 2019. Icdar2019 robust reading challenge on arbitrary-shaped text (rrc-art). *arXiv preprint arXiv:1909.07145*.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*.

Gao, H.; Xu, A.; and Huang, H. 2021. On the Convergence of Communication-Efficient Local SGD for Federated Learning. In *AAAI*.

Gao, Y.; Chen, Y.; Wang, J.; Tang, M.; and Lu, H. 2019. Reading scene text with fully convolutional sequence modeling. *Neurocomputing*, 339: 161–170.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting Gradients–How easy is it to break privacy in federated learning? *arXiv preprint arXiv:2003.14053*.

Goel, V.; Mishra, A.; Alahari, K.; and Jawahar, C. 2013. Whole is greater than sum of parts: Recognizing scene text words. In *ICDAR*.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hitaj, B.; Ateniese, G.; and Perez-Cruz, F. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *CCS*.

Hu, W.; Cai, X.; Hou, J.; Yi, S.; and Lin, Z. 2020. GTC: Guided Training of CTC towards Efficient and Accurate Scene Text Recognition. In *AAAI*.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; et al. 2015. ICDAR 2015 competition on robust reading. In *ICDAR*.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and De Las Heras, L. P. 2013. ICDAR 2013 robust reading competition. In *ICDAR*.

Li, H.; Wang, P.; Shen, C.; and Zhang, G. 2019. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*.

Li, Q.; He, B.; and Song, D. 2021. Model-Contrastive Federated Learning. In *CVPR*, 10713–10722.

Liu, W.; Chen, C.; and Wong, K.-Y. K. 2018. Char-net: A character-aware neural network for distorted scene text recognition. In *AAAI*.

Liu, Z.; Li, Y.; Ren, F.; Goh, W. L.; and Yu, H. 2018. Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network. In *AAAI*.

Long, S.; He, X.; and Yao, C. 2018. Scene text detection and recognition: The deep learning era. *arXiv preprint arXiv:1811.04256*.

Lucas, S. M.; Panaretos, A.; Sosa, L.; Tang, A.; Wong, S.; Young, R.; Ashida, K.; Nagai, H.; Okamoto, M.; Yamamoto, H.; et al. 2005. ICDAR 2003 robust reading competitions: entries, results, and future directions. *IJDAR*, 7(2-3): 105–122.

Luo, C.; Jin, L.; and Sun, Z. 2019. Moran: A multi-object rectified attention network for scene text recognition. *PR*, 90: 109–118.

Luo, J.; Wu, X.; Luo, Y.; Huang, A.; Huang, Y.; Liu, Y.; and Yang, Q. 2019. Real-world image datasets for federated learning. *arXiv preprint arXiv:1910.11089*.

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 116–131.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012. Top-down and bottom-up cues for scene text recognition. In *CVPR*.

Nguyen, N.; Nguyen, T.; Tran, V.; Tran, M.-T.; Ngo, T. D.; Nguyen, T. H.; and Hoai, M. 2021. Dictionary-Guided Scene Text Recognition. In *CVPR*, 7383–7392.

Phong, L. T.; Aono, Y.; Hayashi, T.; Wang, L.; and Moriai, S. 2018. Privacy-preserving deep learning via additively homomorphic encryption. *TIFS*, 13(5): 1333–1345.

Quy Phan, T.; Shivakumara, P.; Tian, S.; and Lim Tan, C. 2013. Recognizing text with perspective distortion in natural scenes. In *ICCV*.

Reisizadeh, A.; Mokhtari, A.; Hassani, H.; Jadbabaie, A.; and Pedarsani, R. 2019. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. *arXiv preprint arXiv:1909.13014*.

Risnumawan, A.; Shivakumara, P.; Chan, C. S.; and Tan, C. L. 2014. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18): 8027–8048.

Shi, B.; Bai, X.; and Yao, C. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 39(11): 2298–2304.

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2018. Aster: An attentional scene text recognizer with flexible rectification. *TPAMI*.

Shokri, R.; and Shmatikov, V. 2015. Privacy-preserving deep learning. In *CCS*.

Su, B.; and Lu, S. 2014. Accurate scene text recognition based on recurrent neural network. In *ACCV*.

Sun, J.; Li, A.; Wang, B.; Yang, H.; Li, H.; and Chen, Y. 2021. Soteria: Provable Defense Against Privacy Leakage in Federated Learning From Representation Perspective. In *CVPR*, 9311–9319.

Veit, A.; Matera, T.; Neumann, L.; Matas, J.; and Belongie, S. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.

Voigt, P.; and Von dem Bussche, A. 2017. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *ICCV*.

Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H. H.; Farhad, F.; Jin, S.; Quek, T. Q.; and Poor, H. V. 2019. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. *arXiv preprint arXiv:1911.00222*.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *TIST*, 10(2): 12.

Yu, D.; Li, X.; Zhang, C.; Liu, T.; Han, J.; Liu, J.; and Ding, E. 2020. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, 12113–12122.

Yue, X.; Kuang, Z.; Lin, C.; Sun, H.; and Zhang, W. 2020. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV*, 135–151.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhan, F.; and Lu, S. 2019. Esir: End-to-end scene text recognition via iterative image rectification. In *CVPR*.

Zhan, F.; Xue, C.; and Lu, S. 2019. GA-DAN: Geometry-aware domain adaptation network for scene text detection and recognition. In *ICCV*.

Zhang, C.; Gupta, A.; and Zisserman, A. 2020. Adaptive text recognition through visual matching. In *ECCV*, 51–67.

Zhang, H.; Yao, Q.; Yang, M.; Xu, Y.; and Bai, X. 2020. AutoSTR: Efficient Backbone Search for Scene Text Recognition. In *ECCV*.

Zhang, Y.; Nie, S.; Liu, W.; Xu, X.; Zhang, D.; and Shen, H. T. 2019. Sequence-to-sequence domain adaptation network for robust text image recognition. In *CVPR*.

Zhu, W.; Baust, M.; Cheng, Y.; Ourselin, S.; Cardoso, M. J.; and Feng, A. 2019. Privacy-Preserving Federated Brain Tumour Segmentation. In *Machine Learning in Medical Imaging: 10th International Workshop*.