



Distributed Learning based on Asynchronized Discriminator GAN for remote sensing image segmentation

Mingkang Yuan

Key Laboratory of Space Utilization,
Technology and Engineering Center
for Space Utilization, Chinese
Academy of Sciences, Beijing, China

Ye Li*

Key Laboratory of Space Utilization,
Technology and Engineering Center
for Space Utilization, Chinese
Academy of Sciences, Beijing, China

Jiaxi Sun

Institute of Automation, Chinese
Academy of Sciences, Beijing, China

Baokun Shi

Beijing University of Technology,
Beijing, China

Jinzhong Xu

Key Laboratory of Space Utilization,
Technology and Engineering Center
for Space Utilization, Chinese
Academy of Sciences, Beijing, China

Lele Xu

Key Laboratory of Space Utilization,
Technology and Engineering Center
for Space Utilization, Chinese
Academy of Sciences, Beijing, China

Yisu Wang

Key Laboratory of Space Utilization,
Technology and Engineering Center
for Space Utilization, Chinese
Academy of Sciences, Beijing, China

ABSTRACT

Remote sensing images are usually distributed in different departments and contain private information, so they normally cannot be available publicly. However, it is a trend to jointly use remote sensing images from different departments, because it normally enables the model to capture more information and remote sensing image analysis based on deep learning generally requires lots of training data. To address the above problem, in this paper, we apply a distributed asynchronized discriminator GAN framework (DGAN) to jointly learn remote sensing images from different client nodes. The DGAN is composed of multiple distributed discriminators and a central generator, and only the synthetic remote sensing images generated by the DGAN are used to train a semantic segmentation model. Based on DGAN, we establish an experimental platform composed of multiple different hosts, which adopts socket and multi-process technology to realize asynchronous communication between hosts, and visualize the training and testing process. During DGAN training, instead of original remote sensing images or convolutional network model information, only synthetic images, losses and labeled images are exchanged between nodes. Therefore, the DGAN well protects the privacy and security of the original remote sensing images. We verify the performance of the DGAN on three remote sensing image datasets (City-OSM, WHU and Kaggle

Ship). In the experiments, we take different distributions of remote sensing images in client nodes into consideration. The experiments show that the DGAN has a great capacity for distributed remote sensing image learning without sharing the original remote sensing images or the convolutional network model. Moreover, compared with a centralized GAN trained on all remote sensing images collected from all client nodes, the DGAN can achieve almost the same performance in semantic segmentation tasks for remote sensing images.

CCS CONCEPTS

• Security and privacy; • Human and societal aspects of security and privacy; • Privacy protections;

KEYWORDS

DGAN, Remote sensing images, Privacy

ACM Reference Format:

Mingkang Yuan, Ye Li, Jiaxi Sun, Baokun Shi, Jinzhong Xu, Lele Xu, and Yisu Wang. 2022. Distributed Learning based on Asynchronized Discriminator GAN for remote sensing image segmentation. In *2022 the 8th International Conference on Communication and Information Processing (ICCIP 2022)*, November 03–05, 2022, Beijing, China. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3571662.3571668>

*Corresponding author: liye@csu.ac.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICCIP 2022, November 03–05, 2022, Beijing, China

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9710-0/22/11...\$15.00

<https://doi.org/10.1145/3571662.3571668>

1 INTRODUCTION

With the rapid development of remote sensing technology, remote sensing images have been widely used in the exploitation of earth resources, territorial control and environmental monitoring. Deep learning has been applied to remote sensing image analyses, such as target detection [1], semantic segmentation [2], etc. It's widely known that a large amount of data is necessary for training a successful deep learning model. However, remote sensing images

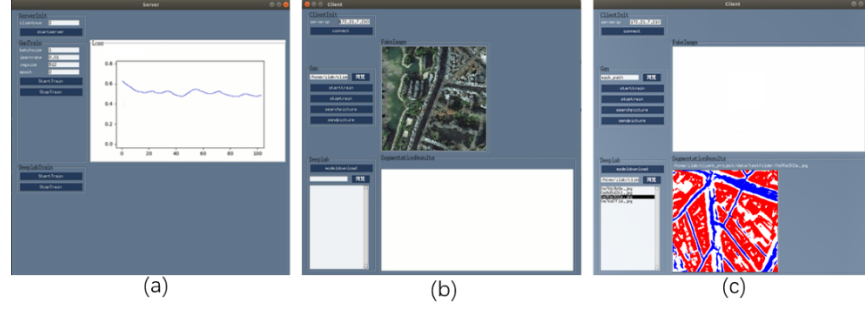


Figure 1: Schematic diagram of DGAN experimental platform. (a) UI interface of the server. (b) UI interface of the client. (c) UI interface of clients for semantic segmentation tasks

normally cannot be available publicly, because they are usually distributed in different departments and contain privacy information. If these remote sensing images are centralized, it may lead to the leakage of data privacy. To solve the problem of privacy protection, we apply federated learning (FL) to remote sensing image analyses. Instead of directly exposing users' data, FL often communicates the model gradients [3]. However, it has been proved in recent years that the model gradients leak data information [8].

To solve the above problems, we apply a distributed asynchronous discriminator GAN framework (DGAN) to the distributed learning of remote sensing images. The DGAN refers to the method proposed by Qi Chang et al. [10], which is used in brain tumor images and cell nucleus images. In DGAN, instead of communicating the gradient information, only the synthetic images, losses and labeled images are exchanged between nodes. Finally, the synthetic images generated by the DGAN are used to train a semantic segmentation model.

To summarize, this paper makes the following contributions:

- Compared with the work proposed by Qi Chang et al. [10], we achieve the distributed learning of remote sensing images instead of learning medical images. The remote sensing images are normally protected privately and have more complex background than the medical images, and therefore it is necessary to further verify the DGAN's performance on remote sensing datasets.
- Based on the DGAN, we establish an experimental platform, and visualize the training and testing process. A schematic diagram of the DGAN experimental platform is shown in Figure 1.
- In our experiments, we deploy DGAN on three independent hosts for the distributed learning, and socket and multi-process technology are used to realize asynchronous communication between the server and multiple clients. In the work of Qi Chang et al. [10], they did not deploy their method in distributed hosts. In addition, we take different distributions of remote sensing images in client nodes into consideration for further verification of the DGAN.

2 RELATED WORK

FL usually realizes joint learning of distributed datasets by sharing model information [4-9]. FedAvg [5] deployed the same deep

learning model at all nodes, and then averaged parameters of local models with weights proportional to sizes of the client datasets for achieving distributed learning. FedProx [6] shared the global model at all nodes, and adds a proximal term to the client cost functions for limiting the impact of local updates. But it is undeniable that the model information (parameters, gradients) sharing exposes FL to a new risk. For instance, poisoning attacks [18] can manipulate a client node and add tampered data weight to the global model, which undermine the accuracy of the global model. Backdoor attacks [11] change the model's behavior by inserting a backdoored component directly into the model, which may affect the classification results. S.Truxex et al. [12] presented a generalized framework for the development of a membership inference attack model, in which a member can use the global model to infer the training data information of other users. Therefore, sharing model information in FL easily causes privacy disclosure of original image data.

To protect data privacy, Qi Chang et al [10] proposed the distributed GAN, which is used in brain tumor images and nucleus images. The distributed GAN is composed of multiple distributed discriminators and a central generator, and protects the privacy of the original data by only communicating the labeled images, losses and synthetic images. However, they did not deploy the distributed GAN on multiple hosts to verify the DGAN in their experiment. In this paper, we apply the DGAN on remote sensing images which have more complex background than medical images. And we set up an experimental platform to deploy DGAN on different hosts for further verifying the DGAN's performance. Finally, the synthetic images generated by the generator are used as training data to train the deep learning model.

3 METHOD

3.1 DGAN architecture

The DGAN is composed of multiple distributed discriminators and a central generator. The central generator, denoted as G , takes the labeled images as input and generates synthetic images to fool the discriminators. The local discriminators, denote as D_1 to D_n , distinguish between real and fake images. In the framework, only the synthetic images, losses, and labeled images are exchanged between the central generator and local discriminators. In this way, the original images of local discriminators and model information are not shared, and therefore the privacy of the original images is

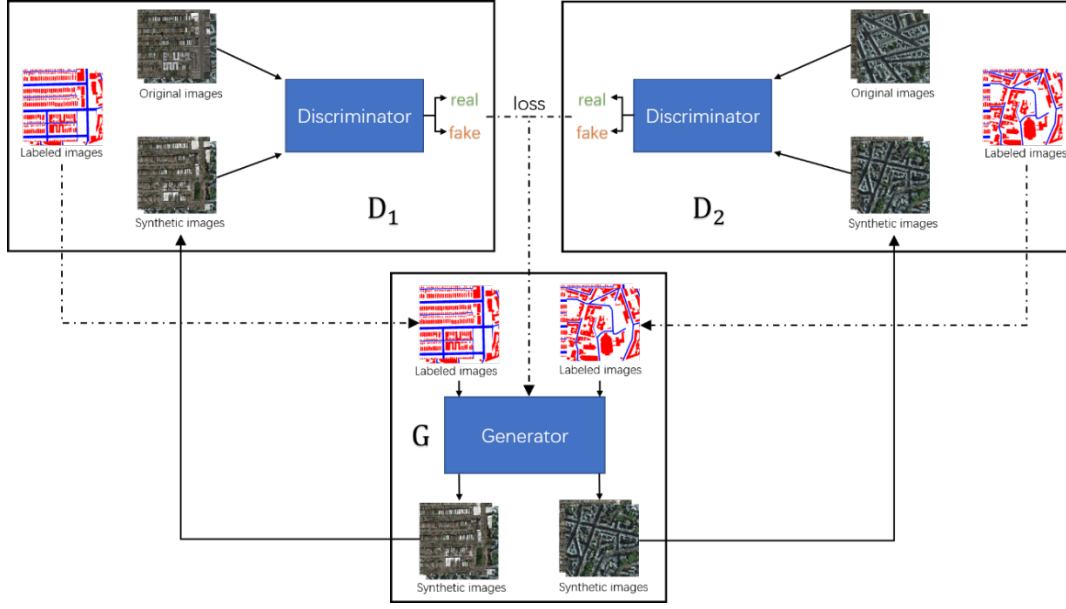


Figure 2: The overall structure of the DGAN. It contains a central generator G and multiple distributed discriminators $D_1, \dots, D_n$. This paper adopts two discriminators D_1 and D_2 . G receives the labeled images from D_1 and D_2 and generates synthetic images. D_1 and D_2 learn to distinguish between real and fake images and send losses.

protected. An overview of the proposed architecture is shown in Figure 2.

The central generator consists of three convolutional layers, nine residual blocks and two transposed convolutions. All non-residual convolutional layers are followed by batch normalization and the ReLU activation. Each discriminator has the same structure as that in PatchGAN. The discriminator individually quantifies the fake or real value of different small patches in the image.

3.2 Optimization process

The optimization process is divided into two parts: D-update and G-update. In each iteration, the sample minibatch contains k images, and k is fine-tuned on different datasets, roughly ranging from 2 to 5. Here, θ_G and θ_{D_j} are the parameters of the generator G and j -th discriminator D_j ; r denotes the original images owned by the discriminators; l is the labeled images corresponding to the original images; $\hat{r}$ is the synthetic images generated by the generator; r_i^j denotes the i -th original image of the minibatch of the j -th client node; similarly to r_i^j , l_i^j is the labeled image; $\hat{r}_i^j$ is the synthetic image. The update steps are as follows:

D-update: Calculating the loss for j -th discriminator D_j and updating D_j , $j = 1, 2, \dots, N$.

- Send the k labeled images l^j of D_j to the central generator, $l^j = \{l_1^j, \dots, l_k^j\}$.
- The central generator G takes l^j as input, and generates k synthetic images $\hat{r}^j$, and then sends $\hat{r}^j$ to the D_j , $\hat{r}^j = \{\hat{r}_1^j, \dots, \hat{r}_k^j\}$.

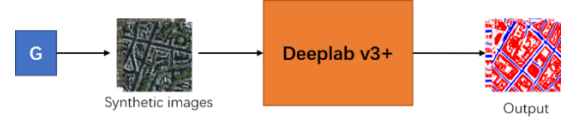


Figure 3: Using the trained G as a data provider to train the DeepLab v3+ model.

- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_{D_j}} \frac{1}{k} \sum_{i=1}^k \left[\log D_j(r_i^j) + \log(1 - D_j(G(l_i^j))) \right]$$

- Repeat the above steps until all client nodes are updated.

G-update: After updating all discriminators, G will be updated using the adversarial loss.

- The G collects all losses.
- Update G , and the loss function is as follows:

$$\nabla_{\theta_G} \frac{1}{Nk} \sum_{j=1}^N \sum_{i=1}^k \log(1 - D_j(G(l_i^j)))$$

3.3 Semantic segmentation

After finishing the training of the DGAN, we collected the synthetic images from the DGAN as the training set for different tasks. In this paper, we use the trained G as a data provider to train DeepLab v3+ [14] model for a semantic segmentation task on remote sensing images, which is shown in Figure 3.

3.4 Platform establishing

We set up an experimental platform to deploy DGAN on different hosts, and visualize the training and testing process. Figure 1 shows a schematic diagram of the platform. Socket and multi-process technology are adopted to realize asynchronous communication between the server and clients. The following experiments are achieved based on the platform to learn the distributed remote sensing data from multiple hosts.

4 EXPERIMENTS

In this section, we perform experiments on remote sensing datasets to illustrate how DGAN learns the data distributions from different subsets, and then apply synthetic datasets to train a semantic segmentation model. Three different remote sensing datasets are used: City-OSM [15], WHU building dataset [16], and Kaggle Ship [17]. We use a centralized GAN (CGAN) for comparison, and CGAN’s generator and discriminator network is the same as DGAN. Unlike DGAN’s distributed subsets, CGAN centralizes all subsets as a training set.

4.1 Environment configuration

We use the central generator as a server with two Intel Xeon Gold 6244 CPUs, 128GB RAM and two NVIDIA Tesla A100 GPUs. Two distributed discriminators act as clients, each with an Intel Xeon Gold 6230 CPU, 128GB RAM, and an NVIDIA GeForce RTX 3090 GPU.

4.2 Evaluation metrics

We adopt the following metrics to evaluate the segmentation performance of remote sensing datasets: Dice score (Dice), Pixel Accuracy score (Pa), and Mean Intersection over Union score (Miou).

$$\text{Dice}(R, P) = \frac{2|R \cap P|}{R + P}$$

$$\text{Pa}(R, P) = \frac{|R \cap P|}{R}$$

$$\text{Miou}(R, P) = \frac{|R \cap P|}{R \cup P}$$

Where R represents the ground-truth mask and P represents the segmentation result.

4.3 Experiments on different datasets

In this subsection, we compare the quality of distributed learning in 3 settings: (1) Seg_Real: The original remote sensing images are used to train the semantic segmentation model; (2) Seg_CGAN: The remote sensing images generated by CGAN are used to train the semantic segmentation model; (3) Seg_DGAN: The remote sensing images generated by DGAN are used to train the semantic segmentation model. In all experiments, the test set remains the same for fair comparison.

4.3.1 City-OSM. The City-OSM dataset is city remote sensing images of Chicago, Paris, and Berlin, including 1632 images with an image resolution of 3328×3072 , and the annotated categories include buildings and roads. We removed some images with a large background, and selected 640 images as the training set and 160 images as the test set. There are two distributed discriminators

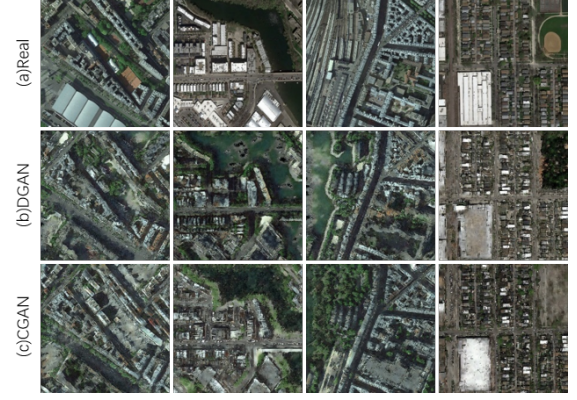


Figure 4: Examples of synthetic images on the City-OSM dataset. (a) Original remote sensing images. (b) Synthetic images generated by DGAN. (c) Synthetic images generated by CGAN.

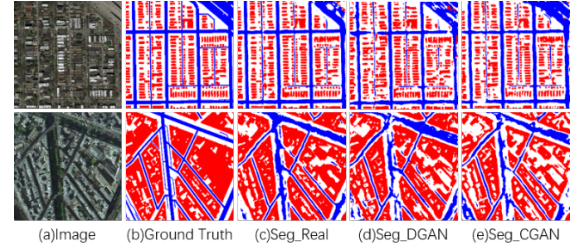


Figure 5: Visualization results on City-OSM dataset for the semantic segmentation task. (a) Test images. (b) Ground-truth. (c)~(e) are the segmentation results of three models which are trained by using original city remote sensing images, synthetic remote sensing images of DGAN, and synthetic remote sensing images of CGAN, respectively.

in this paper, each with 320 original city remote sensing images. We conduct the following segmentation experiments: (1) Seg_Real: 640 original remote sensing images are used to train the semantic segmentation model; (2) Seg_CGAN: 640 synthetic remote sensing images generated by CGAN are used to train the semantic segmentation model; (3) Seg_DGAN: 640 synthetic remote sensing images generated by DGAN are used to train the semantic segmentation model. Figure 4 and Figure 5 show the synthetic remote sensing images and semantic segmentation results. Table 1 shows the comparison of semantic segmentation indexes of city-OSM city remote sensing images.

4.3.2 WHU building dataset. The WHU City Remote Sensing Dataset contains two subsets: The East Asian subset and the Global Cities subset. The East Asian remote sensing subset consists of 6 adjacent satellite images, covering 860 square kilometers in East Asia, with a ground resolution of 0.45m. The entire image is seamlessly cropped into 17,388 512×512 tiles, the annotated category only includes buildings. For convenient training and testing, we removed

Table 1: Semantic segmentation results on the City-OSM dataset Images

Method	Dice	Pa	Miou
Seg_Real	0.7937	0.8036	0.6594
Seg_CGAN	0.7384	0.7513	0.5866
Seg_DGAN	0.7282	0.7445	0.5751

Table 2: Semantic segmentation results on the WHU East Asia dataset Images

Method	Dice	Pa	Miou
Seg_Real	0.9014	0.9815	0.8326
Seg_CGAN	0.8065	0.9685	0.7138
Seg_DGAN	0.8457	0.9706	0.7584

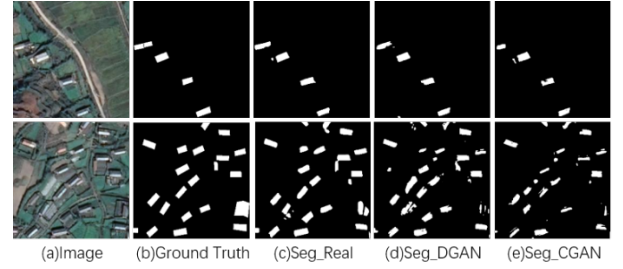
Table 3: Semantic segmentation results on the WHU Global City dataset Images

Method	Dice	Pa	Miou
Seg_Real	0.8382	0.8797	0.7302
Seg_CGAN	0.7840	0.8427	0.6595
Seg_DGAN	0.7927	0.8444	0.6695

**Figure 6: Examples of synthetic images on the WHU East Asian dataset. (a) Original remote sensing images. (b) Synthetic images generated by DGAN. (c) Synthetic images generated by CGAN.**

remote sensing images that only contain the background (field, forest). Finally, there are 3130 remote sensing images in the training set and 903 remote sensing images in the test set. Each discriminator has 1565 remote sensing images. Similar to City-OSM, we conduct the segmentation experiments. Figure 6 and Figure 7 show the synthetic remote sensing images and semantic segmentation results. Table 2 shows the comparison of semantic segmentation indexes of WHU East Asia remote sensing images.

The WHU Global City subset is collected from various remote sensing resources around the world. It contains 204 remote sensing images of 10 cities in total. The ground resolution ranges from 0.3m to 2.5m, and the image resolution is 512×512 . We take 150 remote sensing images as the training set and 54 remote sensing images as

**Figure 7: Visualization results on WHU East Asia dataset for the semantic segmentation task. (a) Test images. (b) Ground truth. (c)~(e) are the segmentation results of three models which are trained by using original city remote sensing images, synthetic remote sensing images of DGAN, and synthetic remote sensing images of CGAN, respectively.**

the test set. Considering the small amount of data, we augmented the training set to 450 by flipping and rotating. Each discriminator has 225 remote sensing images. Similar to City-OSM, we conduct the segmentation experiments. Figure 8 and Figure 9 show the synthetic remote sensing images and semantic segmentation results. Table 3 shows the comparison of semantic segmentation indexes of WHU Global City remote sensing images.

4.3.3 Kaggle Ship. The Kaggle Ship dataset comes from the 2018 Kaggle Competition, with a total of 192,556 768×768 images, and the annotated category only includes ships. For convenient training and testing, we removed the images without ships, and selected 3950 remote sensing images as the training set and 1660 remote sensing images as the test set. Two distributed discriminators are used, each with 1975 remote sensing images. Similar to City-OSM, we conduct the segmentation experiments. Figure 10 and Figure 11 show the

Table 4: Semantic segmentation results on the Kaggle Ship dataset Images

Method	Dice	Pa	Miou
Seg_Real	0.9768	0.9982	0.9555
Seg_CGAN	0.8528	0.9899	0.7706
Seg_DGAN	0.8809	0.9917	0.8063



Figure 8: Examples of synthetic images on the WHU Global City dataset. (a) Original remote sensing images. (b) Synthetic images generated by DGAN. (c) Synthetic images generated by CGAN.

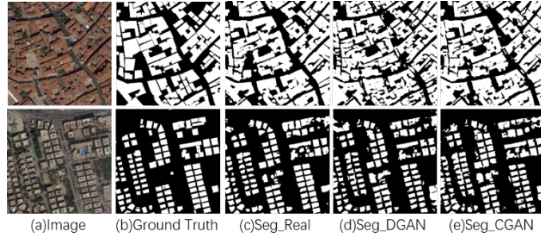


Figure 9: Visualization results on WHU Global City dataset for the semantic segmentation task. (a) Test images. (b) Ground-truth. (c)~(e) are the segmentation results of three models which are trained by using original city remote sensing images, synthetic remote sensing images of DGAN, and synthetic remote sensing images of CGAN, respectively.

synthetic remote sensing images and semantic segmentation results. Table 3 shows the comparison of semantic segmentation indexes of Ship remote sensing images.

4.3.4 Results. As shown in Tables 1~4, compared with the model directly trained on the original images, the semantic segmentation accuracies of the models trained on the CGAN’s and DGAN’s synthetic images decrease. Compared with the CGAN, the DGAN has similar semantic segmentation performance, and does not transmit the original data among the client nodes during training. Therefore, compared with CGAN, DGAN can better protect the privacy of original remote sensing image data while ensuring the performance of semantic segmentation. Figures 6, 8, and 10 show that the DGAN



Figure 10: Examples of synthetic images on the Kaggle Ship dataset. (a) Original remote sensing images. (b) Synthetic images generated by DGAN. (c) Synthetic images generated by CGAN.

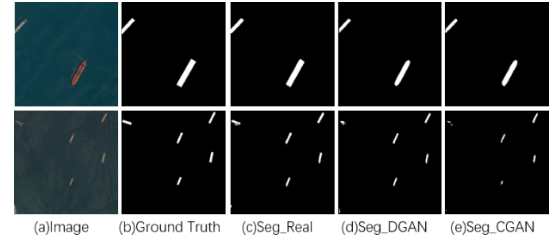


Figure 11: Visualization results on Kaggle Ship dataset for the semantic segmentation task. (a) Test images. (b) Ground-truth. (c)~(e) are the segmentation results of three models which are trained by using original city remote sensing images, synthetic remote sensing images of DGAN, and synthetic remote sensing images of CGAN, respectively.

can commendably learn the data distributions, and Figures 7, 9, and 11 prove that the synthetic images can be commendably used in the semantic segmentation task.

4.4 Reconsider the data distribution

In the above experiments, remote sensing images of different cities are mixed. However, the reality is that different client nodes may have remote sensing images of different cities/regions. To get closer to reality, we divide the images according to different cities in this section.

4.4.1 City-OSM. In this subsection, we use the remote sensing images of Chicago and Paris from the City-OSM. Due to the inconsistency in the number of images in Chicago and Paris, we culled the images so that the number of images in Chicago and Paris is

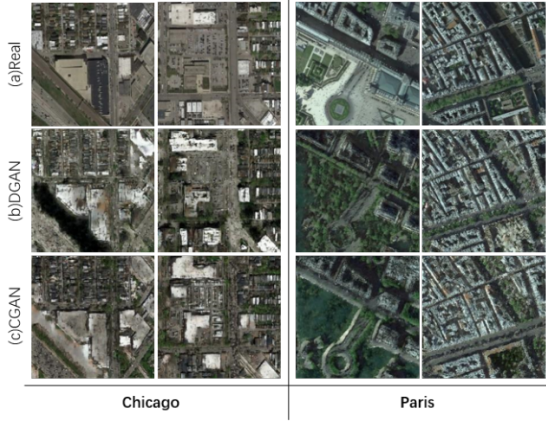


Figure 12: Examples of synthetic images on the City-OSM dataset. (a) Original remote sensing images. (b) Synthetic images generated by DGAN. (c) Synthetic images generated by CGAN.

the same. Finally, a total of 500 remote sensing images are used as the training set while the discriminators have remote sensing images of different cities. Because of the different street layouts, greening degrees and architectural styles in different cities, the data distribution gap becomes larger. We found that if the discriminators update every iteration, it will cause the model to be difficult to converge. We think it's because the generator does not have enough training to learn the data distribution owned by the current communication node, so the discriminator is easy to judge the real and fake images. With model training, the discriminator loss decreases rapidly and approaches zero. Through experiments, we found that adjusting the update frequency to one epoch would make the model convergence easier than updating the model every iteration. Figure 12 and Figure 13 show the synthetic remote sensing images and semantic segmentation results. Table 5 demonstrates the comparison of semantic segmentation indexes of City-OSM City remote sensing images.

4.4.2 WHU building dataset. The WHU Global Cities Dataset contains remote sensing images of 10 cities: Wuhan, Taiwan, Los Angeles, Ottawa, Cairo, Milan, Santiago, Cordova, Venice, and New York. We divided the 10 cities into two subsets by their architectural styles. The D_1 discriminator has remote sensing images of 5 cities of Wuhan, Cairo, Milan, Cordova and Venice, the D_2 discriminator has remote sensing images of 5 cities of Taiwan, Los Angeles, Ottawa, San Diego, and New York. We performed the same data augmentation operation as above. Each discriminator has 225 remote sensing images. And we found that adjusting the update



Figure 13: Visualization results on City-OSM dataset for the semantic segmentation task. (a) Test images. (b) Ground-truth. (c)~(e) are the segmentation results of three models which are trained by using original city remote sensing images, synthetic remote sensing images of DGAN, and synthetic remote sensing images of CGAN, respectively.

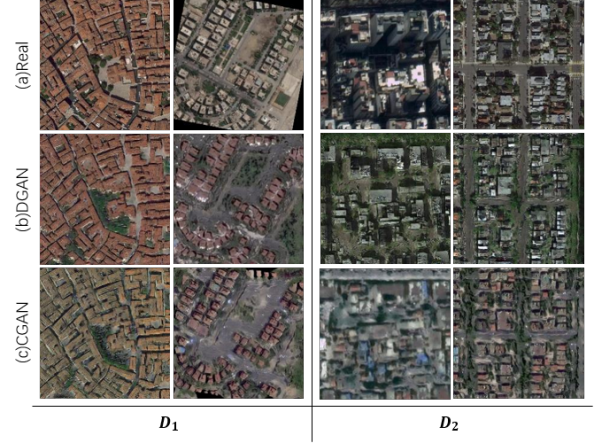


Figure 14: Examples of synthetic images on the WHU Global City dataset. (a) Original remote sensing images. (b) Synthetic images generated by DGAN. (c) Synthetic images generated by CGAN.

frequency of the discriminators to 4 epochs makes the model easier to converge. Figure 14 and Figure 15 show the synthetic remote sensing images and semantic segmentation results. Table 6 shows the comparison of semantic segmentation indexes of WHU Global City remote sensing images.

4.4.3 Results. After reconsidering data distribution (different client nodes have remote sensing images of different cities), we show the semantic segmentation results in Tables 5, 6. The results demonstrate that the change mentioned above in data distribution does not impact the performance of the DGAN. By adjusting the

Table 5: Semantic segmentation results on the City-OSM dataset Images

Method	Dice	Pa	Miou
Seg_Real	0.8076	0.8161	0.6792
Seg_CGAN	0.7366	0.7466	0.5850
Seg_DGAN	0.7398	0.7543	0.5895

Table 6: Semantic segmentation results on the WHU Global City dataset Images

Method	Dice	Pa	Miou
Seg_Real	0.8376	0.8777	0.7290
Seg_CGAN	0.7815	0.8426	0.6569
Seg_DGAN	0.8041	0.8541	0.6844

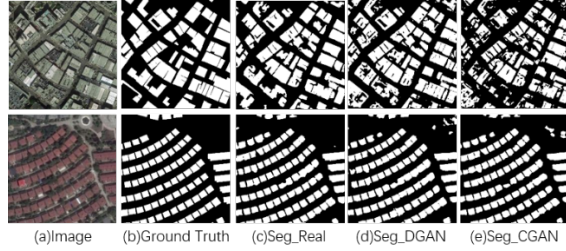


Figure 15: Visualization results on WHU Global City dataset for the semantic segmentation task. (a) Test images. (b) Ground-truth. (c)~(e) are the segmentation results of three models which are trained by using original city remote sensing images, synthetic remote sensing images of DGAN, and synthetic remote sensing images of CGAN, respectively.

discriminators' update strategy, the DGAN can commendably generate synthetic images when different discriminators have remote sensing images of different cities. The examples of synthetic images from DGAN and CGAN are shown in Figures 12, 14. The semantic segmentation results of each method are shown in Figures 13, 15

5 CONCLUSION

In this paper, we apply the DGAN to the distributed learning of remote sensing datasets, and establish an experimental platform which deploys the DGAN on multiple hosts and visualize the training and testing process. During the DGAN training, neither the original data nor the model information is transmitted, which strengthens the privacy protection of original data. Only the labeled images, losses and synthetic images are transmitted between the server and clients. We achieve semantic segmentation tasks for comparing the performance of DGAN and CGAN on different remote sensing datasets. The results show that the performance of the DGAN is almost the same as the CGAN in the semantic segmentation tasks. Compared with CGAN with sharing data of client nodes, the DGAN can better protect data privacy. To get closer to reality, we divide the remote sensing images according to different cities so that different discriminators have the data distributions of different cities. By adjusting the discriminators' update strategy, the change mentioned above in the data distribution does not impact the performance of the DGAN. In the future, we will improve the network structure of the generator and the discriminator to enhance the image synthesis capability. In addition, we will also try to compress the synthetic images and labeled images during communicating for reducing the amount of network traffic.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 61971404 and Grant 61901454, and in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences under Grant 2019168.

REFERENCES

- [1] Li K, Wan G, Cheng G, *et al.* 2020. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159: 296-307.
- [2] Kampffmeyer M, Salberg A B, Jenssen R. 2016. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2016: 1-9.
- [3] Konečný J, McMahan H B, Yu F X, *et al.* 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.
- [4] Wang H, Yurochkin M, Sun Y, *et al.* 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*.
- [5] McMahan B, Moore E, Ramage D, *et al.* 2017. Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics*. PMLR, 2017: 1273-1282.
- [6] Sahu A K, Li T, Sanjabi M, *et al.* 2018. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*.
- [7] Yoon J, Jeong W, Lee G, *et al.* 2021. Federated continual learning with weighted inter-client transfer. *International Conference on Machine Learning*. PMLR, 2021: 12073-12086.
- [8] Sun J, Li A, Wang B, *et al.* 2020. Provable defense against privacy leakage in federated learning from representation perspective. *arXiv preprint arXiv:2012.06043*.
- [9] Bonawitz K, Ivanov V, Kreuter B, *et al.* 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017: 1175-1191.
- [10] Chang Q, Qu H, Zhang Y, *et al.* 2020. Synthetic learning: Learn from distributed asynchronous discriminator GAN without sharing medical image data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020: 13856-13866.
- [11] Bagdasaryan E, Veit A, Hua Y, *et al.* 2020. How to backdoor federated learning. *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020: 2938-2948.
- [12] Truex S, Liu L, Gursoy M E, *et al.* 2019. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 2019.
- [13] Kingma D P, Ba J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [14] Chen L C, Zhu Y, Papandreou G, *et al.* 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 2018: 801-818.
- [15] Kaiser P, Wegner J D, Lucchi A, *et al.* 2017. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 2017, 55(11): 6054-6068.
- [16] Ji S, Wei S, Lu M. 2018. Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set. *IEEE Transactions on Geoscience and Remote Sensing*, 2018, 57(1): 574-586.
- [17] Airbus Ship Detection Challenge. 2018. <https://www.kaggle.com/competitions/airbus-ship-detection/data>
- [18] Cao D, Chang S, Lin Z, *et al.* 2019. Understanding distributed poisoning attack in federated learning. *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019: 233-239.