Mathematical Biosciences
and Engineering

http://www.aimspress.com/journal/mbe

MBE, 21(1): 1610–1624.
DOI: 10.3934/mbe.2024070
Received: 03 October 2023
Revised: 14 November 2023
Accepted: 21 November 2023
Published: 02 January 2024

Research article

Tri-branch feature pyramid network based on federated particle swarm optimization for polyp segmentation

Kefeng Fan[1], **Cun Xu**[2,*,†], **Xuguang Cao**[2,†], **Kaijie Jiao**[2] **and Wei Mo**[2]

[1] China Electronics Standardization Institute, Beijing 100007, China
[2] School of Electronic and Automation, Guilin University of Electronic Technology, Guilin 541004, China

† These authors contributed equally to this work.

* **Correspondence:** Email： 18899599339@163.com.

Abstract: Deep learning technology has shown considerable potential in various domains. However, due to privacy issues associated with medical data, legal and ethical constraints often result in smaller datasets. The limitations of smaller datasets hinder the applicability of deep learning technology in the field of medical image processing. To address this challenge, we proposed the Federated Particle Swarm Optimization algorithm, which is designed to increase the efficiency of decentralized data utilization in federated learning and to protect privacy in model training. To stabilize the federated learning process, we introduced Tri-branch feature pyramid network (TFPNet), a multi-branch structure model. TFPNet mitigates instability during the aggregation model deployment and ensures fast convergence through its multi-branch structure. We conducted experiments on four different public datasets： CVC-ClinicDB, Kvasir, CVC-ColonDB and ETIS-LaribPolypDB. The experimental results show that the Federated Particle Swarm Optimization algorithm outperforms single dataset training and the Federated Averaging algorithm when using independent scattered data, and TFPNet converges faster and achieves superior segmentation accuracy compared to other models.

Keywords: artificial intelligence; deep learning; federated learning; colon polyps segmentation; multi-branch network

1. Introduction

Colorectal cancer is the third most common cancer worldwide and its incidence is increasing every year [1]. About the precursors of colon cancer, it is commonly accepted that most colorectal cancers evolve from adenomatous polyps [2]. Recent surveys and statistics underline that polypoid lesions are precursors to most (>85%) colorectal cancers [3, 4]. In recent years, the application of artificial intelligence technology in medical imaging has demonstrated remarkable effectiveness. Compared to traditional segmentation methods based on threshold, region and edge, the use of semantic segmentation methods in Artificial Intelligence (AI)-colonoscopy detection can significantly reduce the risks associated with misdiagnosis or missed polyps, colorectal tumor lesions and colorectal cancer at all stages of disease progression due to a variety of factors [5].

In AI technology, the amount and diversity of data is important for model training, as expected [6]. However, strict privacy requirements for medical data limit the amount of data that can be shared for model training. Federated learning obtains a central model on the server by aggregating models trained locally on clients [7]. The federated learning process does not need to expose the raw data so it can solve the problem of insufficient data during model training.

Currently, the most widely used method in federated learning is the Federated Averaging algorithm, which uses an equal-weight aggregation approach. However, the application of equal-weight aggregation has certain limitations given the variations in the volume and diversity of distributed data. Among them, Li et al. [8] pointed out that it is expected to achieve a better aggregation effect by adjusting the aggregation weights with reference to some machine learning methods. To maximize the utilization of distributed independent data and optimize the aggregation effect, we propose the Federated Particle Swarm Optimization algorithm. The main contribution of this algorithm is its ability to improve the aggregation effect by optimizing the aggregation weight of the model uploaded by the federated learning client.

During the training process, we observed unstable oscillations when the client used the server aggregation model. To address this, we propose a Tri-branch Feature Pyramid Network. This network, designed for federated learning, uses a multi-branch structure to mitigate the instability experienced during the deployment of the aggregation model. As a result, it allows a more effective use of the independent scattered data.

The main contributions of this paper are as follows:

1) The Federated Particle Swarm Optimization algorithm is proposed as a means of preserving privacy while maximizing the utility of independent scattered data.

2) The Tri-branch Feature Pyramid Network is proposed, which achieves a better training effect by using scattered data under the Federal Particle Swarm Optimization algorithm.

2. Related works

2.1. Federated learning

Federated learning can ensure data privacy and security as it allows multiple data owners to collaboratively train a machine learning model without requiring access to each other's raw data [9]. The process of updating the global model by aggregating local models serves to maintain data privacy and security. Based on the distinctions in data feature space and sample space, federated learning can be categorized into three types: Horizontal federated learning, vertical federated learning and federated transfer learning [10]. Here, federated transfer learning is designed for situations in which data from different parties differs not only in sample ID space but also in feature space [11]. For the vertical federated learning, the objectives of different clients may be different [12]. Learning for vertical distributed data is called vertical federated learning [13]. Horizontal federated learning enables distributed clients to train a shared model and keep their data privacy [14]. Since each client's task is the semantic segmentation of colon polyps, it becomes imperative to employ a distributed client training model within a shared feature space. Consequently, horizontal federated learning emerges as the most appropriate framework.

In the research based on the framework of horizontal federated learning, You et al. [15] proposed an approach to optimize the overall performance of the server's global model by evaluating the contribution of each client terminal model. Li et al. [16], based on federated learning, proposed a dynamic verification model that exhibited better diagnostic accuracy in situations of data imbalance. Abbas et al. [17], based on the context of the loss factor and class imbalance issues achieved better application of image classification. In addition, Ye et al. [18] proposed a robust decentralized stochastic gradient descent method to reduce communication frequency and speed up convergence. Taking into account the heterogeneity of clients, Yu et al. [19] defined the criteria for local convergence. In application, Liu et al. [20] successfully applied the Federated Sveraging algorithm to lung nodule detection and demonstrated its effectiveness. Hu et al. [21] proposed a federated evolutionary feature selection method based on particle swarm optimization algorithm with multiple participants, which can find the feature subsets with a better comprehensive performance.

2.2. Image segmentation on colon polyps

About colon polyp segmentation, Hu et al. [22] highlighted in 2023 that precisely polyp segmentation is still an open issue. This paper aims to achieve high-precision polyp segmentation using a pyramid pooling model with a multi-branch structure and a pyramid structure, and the relevant state of the art in this regard is as follows.

On the multi-branch structure, Liu et al. [23] added a branch of boundary attention to the last detail finding branch to improve the segmentation accuracy. Hu et al. [22] made a memory-keeping pyramid pooling module into each side of the branch of the encoder to enhance the effectiveness on feature extraction. Wang et al [24] used an object detection branch and a mask generation branch to implement a highly accurate anchor-free instance segmentation framework. Through our experiments, we have found that the implementation of a multi-branch structure not only improves the efficiency and accuracy of polyp segmentation, but also improves the stability within the federated learning framework.

On the pyramid pool model and feature pyramid structure, Shi et al. [25] designed a context-aware pyramid aggregation module to boost the network's ability to utilize global context. Shen et al. [26] used four null space convolutional pooling pyramids to improve the inference speed and computational efficiency. Sharma et al. [27] used atrous spatial pyramid pooling to handle the problem of segmenting objects at multiple scales. Wang et al. [28] introduced a network called the stepwise feature fusion transformer (SS-Former). Simultaneously, Chang et al. [29] presented an effective stage-wise feature pyramid network (ESFPNet). The two feature pyramid networks proposed in the last two years have shown excellent performance in colon polyp segmentation.

3. Methods

3.1. Overview

The Federated Particle Swarm Optimization algorithm presented in this paper is based on the federated learning framework, which consists of a client-server architecture. The client uses undisclosed raw data to train a local model, while the server aggregates locally uploaded models into a global model, as shown in Figure 1.

To optimize the aggregation weight, we introduce the Federated Particle Swarm Optimization algorithm. This algorithm optimizes the aggregation weight of the client's optimal model by the effect of the client's training. The global model is generated by this weight aggregation, then the global model is sent to the client as the server model. This server model is then aggregated with the model from each client's local training to generate a local optimal model. This process is illustrated in Figure 1.

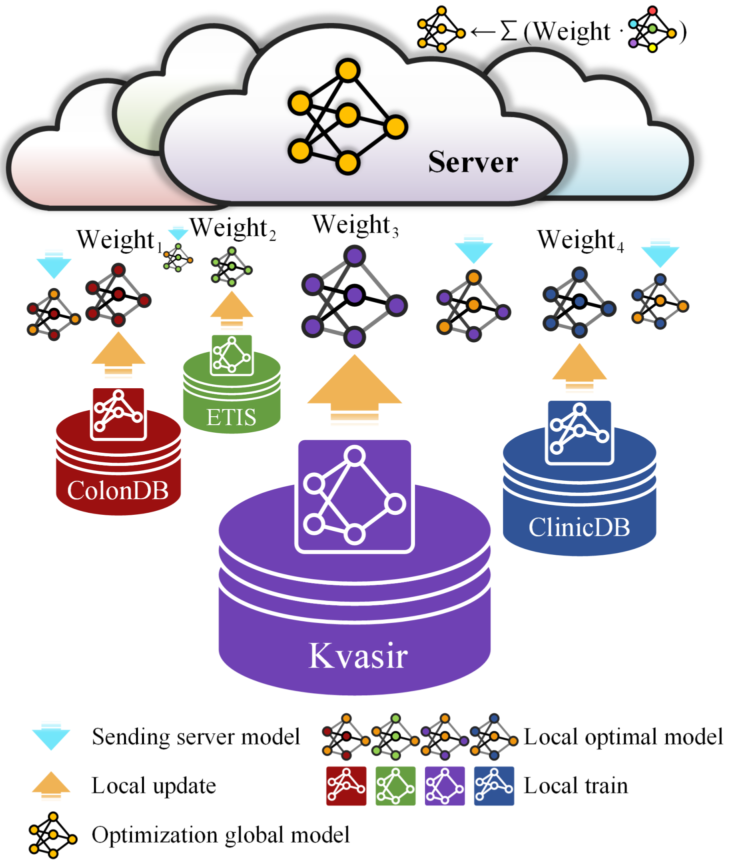


Figure 1. The architecture of the Federated Particle Swarm Optimization algorithm.

Building on this foundation, we propose a Tri-branch Feature Pyramid Network. This network improves and stabilizes the performance of the Federated Particle Swarm Optimization algorithm by exploiting the feature pyramid and multi-branch architecture.

3.2. Federated Particle Swarm Optimization algorithm

Federated learning is often used to deal with fragmented private data. This methodology gained prominence when McMahan et al. [30] introduced the Federated Averaging algorithm. This algorithm mitigates communication frequency by aggregating gradients, making federated learning increasingly suitable for deep learning tasks with high communication overhead. Building on this, another research points out that using the best model from each client's training round during each aggregation cycle can improve the performance of federated learning [31]. Thus, during aggregation, a 20% weight is maintained for each client's superior model.

Regarding the aggregation weights in federated learning, it is expected to achieve a better aggregation effect by determining the weighting weights of aggregation global models according to the proportion of customer samples and adjusting the aggregation weights with reference to machine learning methods [8]. Therefore, we use the concept of Particle Swarm Optimization algorithm [32] to optimize the remaining 80% of the weight. The proportion of customer samples serves as the initial weight in this optimization process:

$$\begin{cases} c_1^t = 0.5d_1 + 0.5d_1 \cdot (t/T)^2 \\ c_2^t = 0.5d_2 + 0.5d_2 \cdot (t/T)^2 \end{cases} \tag{1}$$

$$v_{id}^{t+1} = c_1^t \cdot \left(s_p^t - s_{pb}\right) + c_2^t \cdot (s_p^t - s_{gb}) \tag{2}$$

$$w_{id}^{t+1} = w_{id}^t + v_{id}^{t+1}, \tag{3}$$

where t represents the aggregation round of federated learning, T is the total aggregation round of federated learning and d_1 and d_2 are tunable parameters. Given the significant variation in previous training rounds for each client, these learning parameters c_1^t and c_2^t incrementally increased by Eq (1). w is the weight optimized for each client's training. s_p^t is the optimal fitness derived from the evaluation of the validation set across all client training iterations within the t-th aggregation round for each client. s_{pb} is the best fitness obtained by evaluating the validation set across all client training iterations within all aggregation rounds. The average of the mean intersection over union (mIoU) and Dice coefficients on the validation set is set to the fitness. s_{gb} is the predefined adjustable expected training fitness.

The specific process of Federated Particle Swarm Optimization algorithm is as follows:

Algorithm 1 Federated Particle Swarm Optimization algorithm

```
procedure Server Update:
 Initial model parameter: $c_1, c_2, w_{id}^0, s_{gb}, a_n$
 for every aggregation epoch $t$ do
     for every client $i$ do
         $\varepsilon_{id}^t, s_p^t \leftarrow$ Client Update ($\varepsilon_{id}^{t-1}$)
         if $s_p^t > s_{pb}$:
             $s_{pb} = s_p^t$
         end if
         $w_{id}^{t+1} \leftarrow$ Eq. (1) and Eq. (2)
         $W_{id}^{t+1} = w_{id}^{t+1} + a_n, a_i = 0.2$
```

```
        $\varepsilon_{id}^{t+1} = \sum W_{id}^{t+1} \cdot \varepsilon_{id}^{t}$
    end for
end for
for every client epoch
    client model ← $\varepsilon_{id}^{t-1}$
    Test result ← Each client model is tested by the test dataset
end for
procedure Client Update: $(\varepsilon_{id}^{t-1})$
client model ← $\varepsilon_{id}^{t-1}$
for every client epoch $\tilde{t}$ do
    The client model is trained by the training dataset.
    $\varepsilon_{id}^{\tilde{t}}$ ← This average corresponds to the sum of model gradients
    mIoU and Dice coefficient ← The client model is verified by the validation dataset
    $s_p^{\tilde{t}} = (mIoU + Dice)/2$
end for
$\tilde{t}_{sp} \leftarrow max(s_p^{\tilde{t}})$
$\varepsilon_{id}^{t}, s_p^{t} = \varepsilon_{id}^{\tilde{t}_{max}}, s_p^{\tilde{t}_{sp}}$
return $\varepsilon_{id}^{t}, s_p^{t}$
```

In the Federated Particle Swarm Optimization algorithm above, ε_{id}^{t} represents the model on the i-th client during the t-th round of aggregation. W is the weight assigned to each client during the global model aggregation, consisting of w and a_n. W symbolizes the weight optimized by Eqs (1) and (2), with a sum of 0.8. On the other hand, a_n is the inherent weight of each client model, which accounts for the remaining 20%. Initially, a_n is set to [0, 0, 0, 0]. This configuration ensures a balanced and fair distribution of weights across all client models.

In addition, w_{id}^{0} is set to [0.22, 0.13, 0.38, 0.07], a value derived from the number of sample images present in each dataset. s_{gb} is set to [0.956, 0.943, 0.935, 0.932], a value determined empirically from previous experience. In addition, numerous experiments have led to the conclusion that initializing c_1 to 0.02 and c_2 to 0.01 significantly improves the performance of the Federated Particle Swarm Optimization algorithm.

3.3. Tri-branch Feature Pyramid Network

Severe loss oscillation may occur in federated learning, which affects the convergence of the joint model [33]. To address this problem, we propose the Tri-branch Feature Pyramid Network with a multi-branch structure. This structure is designed to mitigate instability and reduce oscillatory phenomena during the training process. Each branch contains the Res Linear Uper architecture.

Res Linear Uper architecture (RL-Uper). As shown in Figure 2(a), we use UperNet's pyramid pooling module (PPM) [34] to extract high-dimensional features to obtain comprehensive feature information of polyps. The architecture of Res Linear is shown in Figure 2(b). Unlike a typical linear layer, the Residual Linear module (RL) stabilizes the training process and preserves the characteristics of each class in the binary classification problem, which is achieved by using GroupNormal and sigmoid linear unit (SiLu).

Tri-branch Feature Pyramid Network (TFPNet). As shown in Figure 2(c), we use the

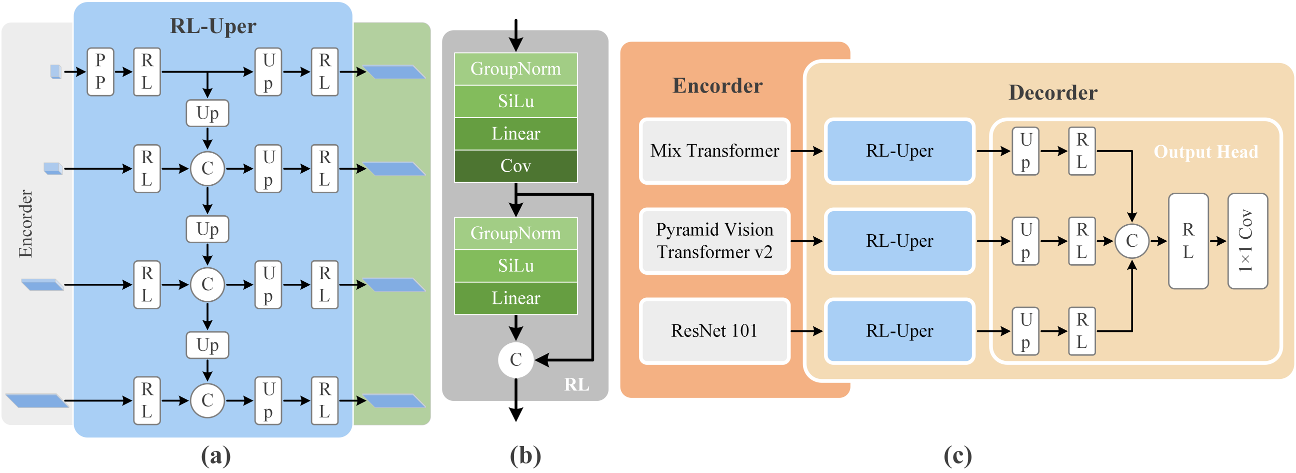


Figure 2. (a) The architecture of Res Linear Uper. (b) The architecture of Res Linear. (c) The architecture of Tri-branch Feature Pyramid Network.

multi-branch structure because it helps to mitigate training fluctuations during distributed training. In addition, we use encoders such as Pyramid Vision Transformer V2 (PVTv2) m4 [35], Mixed Transformer (MiT) m4 [36], and ResNet101 [37] to extract a wide range of basic features. Compared to a single-branch structure, the multi-branch configuration can yield more complex features. The Residual Linear module facilitates the concatenation of feature information from different branches to achieve effective fusion.

4. Experiments

4.1. Dataset

The experimental data for this study was derived from the following public datasets: CVC-ClinicDB [38], CVC-ColonDB [39], Kvasir [40] and ETIS-LaribPolypDB [1], as detailed in Table 1. Given the variance in image size across these datasets, we standardize all images to a resolution of 356 × 356 pixels and establish a batch size of 4. Furthermore, to augment the dataset, we implement random flipping, scaling, rotation and random dilation and erosion operations.

Table 1. The public dataset is used for colon polyp segmentation.

Dataset	CVC-ClinicDB	CVC-ColonDB	Kvasir	ETIS
Images number	612	380	1000	196
Max ratio object	45.88%	63.15%	62.13%	29.05%
Min ratio object	0.34%	0.30%	0.79%	0.11%
Input size	384 × 288	574 × 500	Variable	1225 × 966

Each dataset is allocated to individual clients to emulate the distributed federated learning process across diverse data sources. From each dataset, 80% is utilized for training purposes, 10% dedicated to validation and the other 10% for testing.

4.2. Environment

In the experimental section, all models use the AdamW optimizer with an initial learning rate set to 0.001. In the context of binary classification between polyps and background, it is critical for our model to focus primarily on the distinctive features of polyps while learning to discriminate those of the background. Accordingly, we chose the cross-entropy and the mIoU loss functions to ensure that both the background and the polyps receive adequate attention, and the Dice loss function is used specifically to focus only on polyp characteristics only. In the federated learning process, the server model is frequently sent to the client, which exacerbates the challenge of achieving convergence during model training. To accelerate this convergence, three different loss functions are used. Together, these three loss functions form the training process for the optimization model in our experimental segment. The experiments were run on multiple servers equipped with NVIDIA V100-32GB GPU and Intel® Xeon processors (Skylake, IBRS) CPU.

4.3. Evaluation metrics

To evaluate the binary segmentation accuracy between polyps and background, we selected the mIoU as the binary segmentation evaluation metric for polyps and background and the Dice coefficient was chosen as the segmentation metric for polyps. The formulas for mIoU and Dice are detailed below:

$$Dice = \frac{2 \times n_{PP}}{\sum_j^k n_{Pj} + \sum_j^k n_{jP}} \tag{4}$$

$$mIoU = \frac{1}{2} \sum_i^{k \in (P,B)} \frac{n_{ii}}{\sum_j^k n_{ij} + \sum_j^k n_{ji} - n_{ii}}, \tag{5}$$

where n_{ii} denotes the set of true predicted values as j. k denotes the categories of polyps and background (with P as an abbreviation for polyps and B for background). n_{ii} represents the number of accurately predicted values, while n_{ij} and n_{ji} represent false positives and false negatives, respectively.

4.4. The experiment of federated learning

A comparative experiment is conducted to validate the effectiveness of federated learning in using scattered data for colon polyp segmentation. In this experiment, TFPNet is designated as the benchmark model, Each dataset is independently trained, validated and tested as a comparative method: “Independently”. The cross-entropy loss function supervises the training of each dataset. In the Federated Averaging and Federated Particle Swarm Optimization algorithms, each dataset is deployed across multiple clients for federated learning, with the Federated Averaging algorithm [30] serving as an additional comparison method. The number of server-side epochs for both the Federated Averaging algorithm and the Federated Particle Swarm Optimization algorithm is set to 50, while the number of client-side epochs is set to 3. This configuration is chosen to balance communication costs and stability; too few client-side epochs would increase the cost of sending information to the server, while too many epochs could result in excessive fluctuations during aggregation. Thus, three epochs are chosen as the client epochs. Additionally, to compare fairness, the Federated Averaging algorithm also extracts all clients for training during server epochs. For additional experimental parameters and configuration details, refer to the Federated Particle Swarm Optimization subsection in Section 3 and the Data, Environment and Evaluation metrics subsection in Section 4. The experimental results are shown in Figure 3(a) and Table 2.

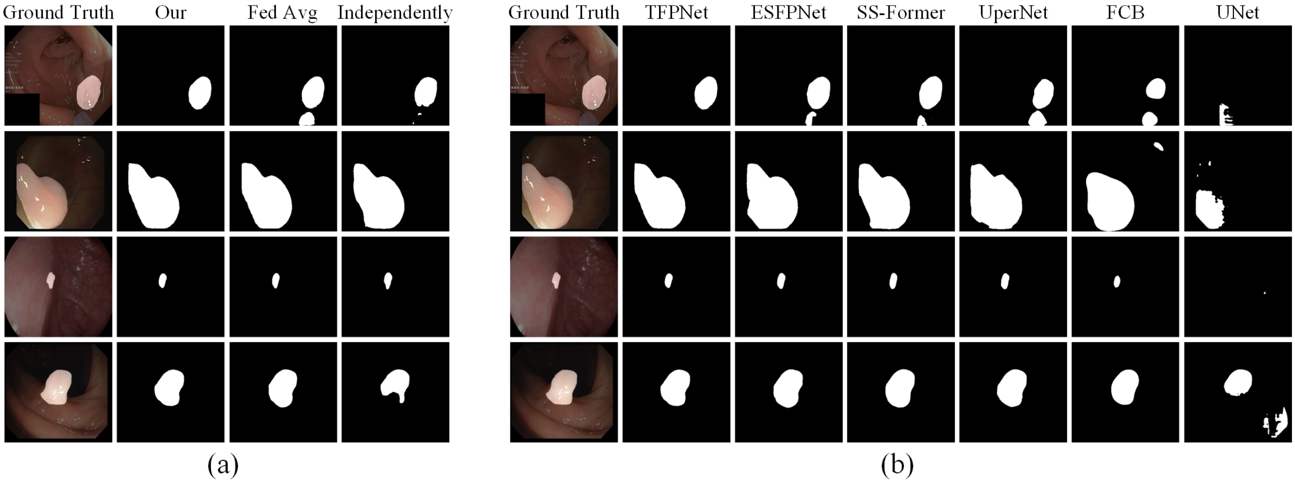


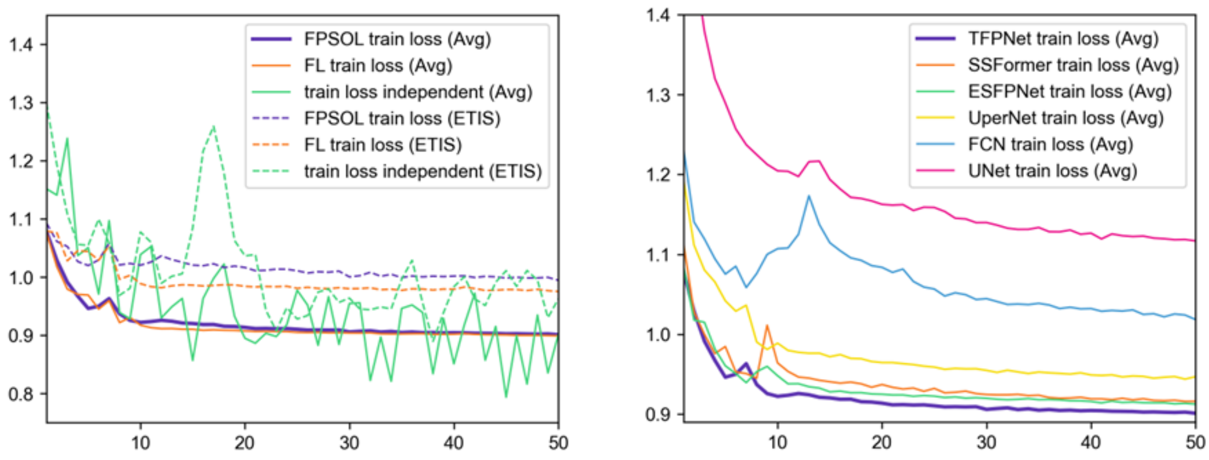
Figure 3. Visual segmentation results (From top to bottom, there are Kvasir, CVC-ClinicDB, ETIS and CVC-ColonDB datasets). (a) In the experiment of federated learning. (b) In the experiment with different baseline networks.

Table 2. The results of the compare experiment with different baseline methods.

Method	Kvasir		ClinicDB		ETIS		ColonDB	
	Dice	mIoU	Dice	mIoU	Dice	mIoU	Dice	mIoU
Independently	0.895	0.897	0.938	0.940	0.828	0.890	0.891	0.898
Fed Avg	0.900	0.900	0.937	0.940	0.828	0.885	0.901	0.905
Our	0.903	0.902	0.946	0.947	0.826	0.878	0.910	0.912

In Table 2 and Figure 3(a), “Independently” indicates that each dataset is trained, validated and tested independently. “Fed Avg” represents the Federated Averaging algorithm and “Our” represents our proposed Federated Particle Swarm Optimization algorithm. Table 2 shows that the training effectiveness of our Federated Particle Swarm Optimization algorithm on the Kvasir, CVC-ClinicDB and CVC-ColonDB datasets surpasses that of the Federated Averaging algorithm and single dataset training independently. To further analyze the performance of the Federated Particle Swarm Optimization algorithm compared to individual training on each dataset and compared to the Federated Averaging algorithm, we used the scores of the metric results generated by the number of samples in the test dataset as data samples. These scores correspond to the mIoU and Dice coefficients shown in Table 2. According to the t-test calculation, it is concluded that the corresponding p-value of the Federated Particle Swarm Optimization algorithm and the Federated Averaging algorithm is less than 0.01 when they are better than the individual training on each dataset. This indicates that both algorithms are significantly better than individual training. This result is particularly relevant in the medical field, where privacy concerns limit data sharing. Distributed training through federated learning could help overcome the problem of insufficient medical data. Similarly, according to the t-test calculation, the corresponding p-value of the Federated Particle Swarm Optimization algorithm was better than the Federal Averaging algorithm by less than 0.01. This indicates that this algorithm significantly outperforms the Federated Averaging algorithm. This suggests that the aggregate weight optimization in the Federated Particle Swarm Optimization Algorithm can improve the segmentation accuracy of colon polyps, providing a clear advantage over the Federated Averaging algorithm.

To further investigate the reasons for the suboptimal performance of the Federated Particle Swarm Optimization algorithm and Federated Averaging algorithm on the ETIS dataset compared to training on independent datasets, we analyzed the average polyline of the loss curves across multiple datasets. In addition, we examined the polyline on the ETIS dataset in these three cases, as shown in Figure 4(a).



(a) (b)

Figure 4. The loss curves. (a) Under the Federated Particle Swarm Optimization algorithm, Federated Averaging algorithm, and training each dataset independently. (b) Under Federated Particle Swarm Optimization algorithm.

In Figure 4(a) and Table 2, the comparative analysis of the loss curves for the Federated Particle Swarm Optimization algorithm, the Federated Averaging algorithm and the independent dataset training on the ETIS dataset shows a higher loss function for the Federated Particle Swarm Optimization algorithm. This shows that the poor performance is due to the underfitting effect during training. Furthermore, as shown in Figure 4(a), the average loss curve for the independent dataset training is lower than that of the two federated learning algorithms on the four datasets. At the same time, according to Table 2, the performance on the remaining three datasets is superior for the federated learning algorithms compared to training each dataset independently, indicating that federated learning across multiple independent datasets effectively prevents overfitting. Therefore, we can infer that the underfitting effect of federated learning on the ETIS dataset is due to the differences between the independent datasets. Although this leads to underfitting when training on the ETIS dataset, it helps prevent overfitting on the other three island datasets. Thus, from a holistic perspective, the Federated Particle Swarm Optimization algorithm proves to be more proficient in utilizing scattered data. In addition, as shown in Figure 4(a) and Table 2, TFPNet exhibits strong performance not only in the Federated Particle Swarm Optimization algorithm but also in the

Federated Averaging algorithm, indicating that the multi-branch structure has significant potential in the context of the federated learning framework.

4.5. The experiment with different baseline networks

To demonstrate that our proposed TFPNet, which is based on the Federated Particle Swarm Optimization algorithm, is better suited to the Federated Particle Swarm Optimization algorithm framework than alternative models, we conducted comparative experiments using different network models as baselines within the Federated Particle Swarm Optimization algorithm framework. Since TFPNet uses MiT and PvTv2 as encoders and the Res Linear Uper is part of the decoder is constructed based on the pyramid structure of UperNet [34], we chose the pyramid networks SS-Former proposed by Wang et al. [28] and ESFPNet proposed by Chang et al. [29], which also use these two encoders, as baseline networks. We also chose the classical network UperNet, which is similar to TFPNet, and other classical networks, Fully Convolutional Networks (FCN) [41] and a network specifically for biomedical image segmentation: UNet [42], as additional baseline networks. In the experiment conducted, we set the number of server epochs to 50 and the number of client epochs to 3. Since the architectures of SS-Former, ESFPNet and UperNet are similar to that of TFPNet, and considering that UNet and FCN are widely used as comparison models in various scenarios, we adopted the same parameters and environment configuration as those used by TFPNet, as detailed in the Environment subsection of this section. The final epoch of training was reserved for testing, and the resulting test results are shown in Figure 3(b) and Table 3.

Table 3. The results of the compare experiment with different baseline networks.

NetWorks	Kvasir		ClinicDB		ETIS		ColonDB	
	Dice	mIoU	Dice	mIoU	Dice	mIoU	Dice	mIoU
UNet	0.900	0.900	0.934	0.937	0.728	0.829	0.821	0.859
FCN	0.865	0.868	0.931	0.932	0.681	0.809	0.829	0.852
UperNet	0.898	0.900	0.927	0.933	0.801	0.868	0.844	0.869
SS-Former	0.896	0.897	0.946	0.946	0.864	0.900	0.898	0.905
ESFPNet	0.899	0.898	0.945	0.946	0.865	0.901	0.904	0.907
TFPNet	0.903	0.902	0.946	0.947	0.826	0.878	0.910	0.912

Table 3 shows that although the mIoU and Dice coefficients of TFPNet are not the highest for the ETIS dataset, these coefficients are indeed the highest for the proposed model in the Kvasir, CVC ClinicDB and CVC-ColonDB datasets. To further analyze the performance of TFPNet against the five comparative models on the four datasets mentioned above, we used the score results generated by the number of samples in the test set as data samples. These scores correspond to the mIoU and Dice coefficients shown in Table 3. According to the t-test calculation, the corresponding p-value of the TFPNet was better than the five comparison models by less than 0.01. This indicates that TFPNet significantly outperforms the five comparison models, combined with the visual segmentation effect in Figure 3(b). It is further evident that the segmentation accuracy of the multi-branch feature pyramid network TFPNet, which is designed based on the Federated Particle Swarm Optimization algorithm, surpasses that of ESFPNet, SS-Former, UperNet, FCN and UNet.

In designing the multi-branch TFPNet, our goal was to use a diverse set of encoders across multiple branches to extract a wider range of features while also using the multi-branch architecture to mitigate training instability. To determine whether TFPNet stabilizes the oscillation of the loss function during the training process, we analyzed the average loss function from four clients, each using different data sets during training. The results of this analysis are shown in Figure 4(b).

As shown in Figure 4(b), the rate of decrease for the TFPNet loss function is the fastest, indicating that TFPNet exhibits greater stability throughout the federated learning process. In addition, the loss function of TFPNet is the lowest after ten generations, suggesting that TFPNet provides the most effective adaptation to the training set. Taken together with the performance of TFPNet presented in Table 3 and Figure 3(b) for the test set, it is clear that TFPNet achieves an optimal fit to the training set without appearing to overfit the test set.

5. Conclusions

In this research, we introduced the Federated Particle Swarm Optimization algorithm, an approach that optimizes the aggregation weight of the server to make more efficient use of independent distributed datasets. Meanwhile, we proposed TFPNet, whose multi-branch structure proved to be more stable and outperformed other networks during the training phase, as shown in our experiments. Both the Federated Particle Swarm Optimization algorithm and TFPNet demonstrated improved use of independent distributed datasets. However, given the performance limitations of the Federated Particle Swarm Optimization algorithm when applied to small datasets, such as the ETIS-LaribPolypDB dataset, we plan to explore the potential of ensemble multi-model training based on federated learning to more effectively utilize these datasets in our future work.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62271456 and in part by Science and Technology Innovation 2030 Major Project of China under Grant 2021ZD0200406.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. Silva, A. Histace, O. Romain, X. Dray, B. Granado, Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer, *Int. J. Comput. Assisted Radiol. Surg.*, **9** (2013), 283–293. https://doi.org/10.1007/s11548-013-0926-3

2. E. Salmo, N. Haboubi, Adenoma and malignant colorectal polyp: pathological considerations and clinical applications, *Gastroenterology*, **7** (2018), 92–102. https://doi.org/10.33590/emjgastroenterol/10313443
3. J. H. Bond, Polyp guideline: diagnosis, treatment, and surveillance for patients with colorectal polyps, *Off. J. Am. Coll. Gastroenterol.*, **95** (2000), 3053–3063. https://doi.org/10.7326/0003-4819-119-8-199310150-00010
4. K. Wallace, H. M. Brandt, J. D. Bearden, Race and prevalence of large bowel polyps among the low-income and uninsured in South Carolina, *Dig. Dis. Sci.*, **61** (2016), 265–272.
5. M. Akbari, M. Mohrekesh, E. Nasr-Esfahani, S. M. Reza Soroushmehr, N. Karimi, S. Samavi, et al., Polyp segmentation in colonoscopy images using fully convolutional network, in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2018. https://doi.org/10.1109/embc.2018.8512197
6. R. Bezen, Y. Edan, I. Halachmi, Computer vision system for measuring individual cow feed intake using RGB-D camera and deep learning algorithms, *Comput. Electron. Agric.*, **172** (2020), 105345. https://doi.org/10.1016/j.compag.2020.105345
7. Y. Chen, X. Sun, Y. Jin, Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation, *IEEE Trans. Neural Netw. Learn. Syst.*, **30** (2019), 4229–4238.
8. L. Li, Y. Fan, M. Tse, K. Y. Lin A review of applications in federated learning, *Comput. Industr. Eng.*, **149** (2020), 106854.
9. T. Wang, Y. Du, Y. Gong, K. R. Choo, Y. Guo, Applications of federated learning in mobile health: scoping review, *J. Med. Int. Res.*, **25** (2023), e43006. https://doi.org/10.2196/43006
10. Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: concept and applications, *ACM Trans. Intell. Syst. Technol.*, **10** (2019), 1–19.
11. S. Feng, B. Li, H. Yu, Y. Liu, Q. Yang, Semi-supervised federated heterogeneous transfer learning, *Knowl. Based Syst.*, **252** (2022), 109384. https://doi.org/10.1016/j.knosys.2022.109384
12. X. Yin, Y. Zhu, J. Hu, A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions, *ACM Comput. Surv.*, **54** (2021), 1–36. https://doi.org/10.1145/3460427
13. Y. Zhang, Y. Hu, X. Gao, D. Gong, Y. Guo, An embedded vertical-federated feature selection algorithm based on particle swarm optimisation, *CAAI Trans. Intell. Technol.*, **8** (2023), 734–754. https://doi.org/10.1049/cit2.12122
14. X. Wang, W. Chen, J. Xia, Z. Wen, R. Zhu, T. Schreck, HetVis: A visual analysis approach for identifying data heterogeneity in horizontal federated learning, *IEEE Trans. Visual. Comput. Graph.*, **29** (2022), 310–319. https://doi.org/10.1109/tvcg.2022.3209347
15. X. You, X. Liu, X. Lin, J. Cai, S. Chen, Accuracy degrading: toward participation-fair federated learning, *IEEE Int. Things J.*, **10** (2023) 10291–10306. https://doi.org/10.1109/jiot.2023.3238038
16. Y. Li, Y. Chen, K. Zhu, C. Bai, J. Zhang, An effective federated learning verification strategy and its applications for fault diagnosis in industrial IoT systems, *IEEE Int. Things J.*, **9** (2022), 16835–16849. https://doi.org/10.1109/jiot.2022.3153343

17. Q. Abbas, K. M. Malik, A. K. J. Saudagar, M. B. Khan, Context-aggregator: An approach of loss- and class imbalance-aware aggregation in federated learning, *Comput. Biol. Med.*, **163** (2023), 107167. https://doi.org/10.1016/j.compbiomed.2023.107167

18. H. Ye, L. Liang, G. Y. Li, Decentralized federated learning with unreliable communications, *IEEE J. Selected Topics Signal Process.*, **16** (2022), 487–500. https://doi.org/10.1109/jstsp.2022.3152445

19. X. Yu, L. Li, X. He, S. Chen, L. Jiang, Federated learning optimization algorithm for automatic weight optimal, *Comput. Intell. Neurosci.*, **2022** (2022), 19. https://doi.org/10.1155/2022/8342638

20. L. Liu, K. Fan, M. Yang, Federated learning: a deep learning model based on resnet18 dual path for lung nodule detection, *Multim. Tools Appl.*, **82** (2023), 17437–17450. https://doi.org/10.1007/s11042-022-14107-0

21. Y. Hu, Y. Zhang, D. Gong, X. Sun, Multiparticipant federated feature selection algorithm with particle swarm optimization for imbalanced data under privacy protection, *IEEE Trans. Artif. Intell.*, **4** (2023), 1002–1016. https://doi.org/10.1109/TAI.2022.3145333

22. K. Hu, W. Chen, Y. Z. Sun, X. Hu, Q. Zhou, Z. Zheng, PPNet: pyramid pooling based network for polyp segmentation, *Comput. Biol. Med.*, **160** (2023), 107028. https://doi.org/10.1016/j.compbiomed.2023.107028

23. G. Liu, M. Zhao, L. Bai, Z. Guo, Cooperation of boundary attention and negative matrix L1 regularization loss function for polyp segmentation, in *26th International Conference on Pattern Recognition*, (2022), 82–88. https://doi.org/10.1109/ICPR56361.2022.9956700

24. D. Wang, S. Chen, X. Sun, Q. Chen, AFP-Mask: anchor-free polyp instance segmentation in colonoscopy, *IEEE J. Biomed. Health Inform.*, **26** (2022), 2995–3006.

25. L Shi, Z Li, J Li, Y Wang, H Wang, Y Guo, AGCNet: a Precise adaptive global context network for real-time colonoscopy, *IEEE Access*, **11** (2023), 59002–59015.

26. T. Shen, X. Li, Automatic polyp image segmentation and cancer prediction based on deep learning, *Frontiers Oncol.*, **12** (2023), 1087438.

27. P. Sharma, A. Gautam, P. Maji, Li-SegPNet: encoder-decoder mode lightweight segmentation network for colorectal polyps analysis, *IEEE Trans. Biomed. Eng.*, **70** (2022), 1330–1339.

28. J. Wang, Q. Huang, F. Tang, J. Meng, J. Su, S. Song, Stepwise feature fusion: local guides global, in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, (2022), 110–120. https://doi.org/10.1007/978-3-031-16437-8_11

29. Q. Chang, D. Ahmad, J. Toth, R. Bascom, W. E. Higgins, ESFPNet: efficient deep learning architecture for real-time lesion segmentation in autofluorescence bronchoscopic video, *Med. Imaging 2023*, **12468** (2023), 1246803. https://doi.org/10.1117/12.2647897

30. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. Y. Arcas, Communication-efficient learning of deep networks from decentralized data, *Artif. Intell. Stat.*, (2017), 1273–1282.

31. E. H. Houssein, A. Sayed, Boosted federated learning based on improved particle swarm optimization for healthcare IoT devices, *Comput. Biol. Med.*, **163** (2023), 107195. https://doi.org/10.1016/j.compbiomed.2023.107195

32. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-international conference on neural networks.*, **4** (1995), 1942–1948.

33. L. Xu, H. Sun, H. Zhao, W. Zhang, H. Ning, H. Guan, Accurate and efficient federated-learning-based edge intelligence for effective video analysis, *IEEE Int. Things J.*, **10** (2023), 12169–12177. https://doi.org/10.1109/jiot.2023.3241039

34. T. Xiao, Y. Liu, B. Zhou, Y. Jiang, J. Sun, Unified perceptual parsing for scene understanding, in *Proceedings of the European conference on computer vision*, (2018), 418–434.

35. W. Wang, E. Xie, X. Li, D. P. Fan, K. Song, D. Liang, et al., PVTv2: improved baselines with pyramid vision transformer, *Comput. Visual Media*, **8** (2022), 415–424. https://doi.org/10.1007/s41095-022-0274-8

36. Q. Chen, Q. Wu, J. Wang, Q. Hu, T. Hu, E. Ding, et al., MixFormer: mixing features across windows and dimensions, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2022), 5249–5259. https://doi.org/10.1109/cvpr52688.2022.00518

37. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 770–778. https://doi.org/10.1109/cvpr.2016.90

38. J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, F. Vilariño, WM-DOVA maps for accurate polyp highlighting in colonoscopy: validation vs. saliency maps from physicians, *Comput. Med. Imaging Graph.*, **43** (2015), 99–111. https://doi.org/10.1016/j.compmedimag.2015.02.007

39. N. Tajbakhsh, S. R. Gurudu, J. Liang, Automated polyp detection in colonoscopy videos using shape and context information, *IEEE Transactions on Medical Imaging.*, **35** (2015), 630–644. https://doi.org/10.1109/tmi.2015.2487997

40. D. Jha, P. H. Smedsrud, M. A. Riegler, P. Halvorsen, T. D. Lange, D. Johansen, et al., Kvasir-SEG: a segmented polyp dataset, *MultiMedia Modeling*, (2020), 451–462. https://doi.org/10.1007/978-3-030-37734-2_37

41. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2015), 3431–3440. https://doi.org/10.1109/TPAMI.2016.2572683

42. O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, (2015), 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

AIMS Press