Federated Learning Pipeline for Wheat Head Instance Segmentation: A Comparison of Federated Learning Approaches and Centralized Learning Techniques

Amman Yusuf
amman.yusuf@ucalgary.ca
University of Calgary
Calgary, Alberta, Canada

Farhad Maleki
farhad.maleki1@ucalgary.ca
University of Calgary
Calgary, Alberta, Canada

ABSTRACT

Centralized learning approaches store data on a central server for analysis, but this approach can pose security, regulatory, and privacy challenges, even in the agricultural field where privacy issues arise when collecting data from different farms. Federated learning aims to solve the privacy and computational limitations posed by centralized learning. The aim of this study is to implement and evaluate a federated learning pipeline for wheat head instance segmentation. The federated learning pipeline is designed to be model agnostic, and will use a modified U-Net model for comparison between clients. The study will compare FedAvg and FedProx approaches for wheat head instance segmentation. The models are then evaluated using Intersection Over Union and a Dice score, and will be compared against a traditional centralized learning baseline. Wheat head images from 18 different institutions are sourced from the Global Wheat Head Dataset. The results showed that both FL approaches matched the centralized approach on most of the selected domains, with FedAvg achieving a Dice score of 0.9355 on the test dataset when compared to the centralized approach with a Dice score of 0.8904. The study also identified the more effective FL approach for wheat head segmentation in a high data variability scenario, which was FedProx.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning**.

KEYWORDS

federated learning, neural networks

ACM Reference Format:
Amman Yusuf and Farhad Maleki. 2022. Federated Learning Pipeline for Wheat Head Instance Segmentation: A Comparison of Federated Learning Approaches and Centralized Learning Techniques. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (CPSC 502).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

CPSC 502, Sept 05, 2022, Calgary, AB

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Centralized learning approaches require storing data in one central server for performing analysis. This data must be collected from multiple centers and then shared with a central server. However, this approach can pose security, regulatory, privacy, and technical challenges when it comes to handling client data, particularly in the medical field where personal health information (PHI) is subject to stringent privacy laws that regulate the distribution and disclosure of this sensitive information. This means that sharing PHI data between hospitals/medical centers would require regulatory scrutiny, impacting the effectiveness of centralized learning as not all clients can participate and upload their data to the central server. This often leads to the lack of large-scale datasets that can be used for training machine learning models in the medical domain, where these models could potentially contribute to automating medical workflow to improve patient health and quality of life. Often, radiologists and medical experts perform the demanding and tedious work of annotating medical images for medical workflows. However, these annotations often cannot be used for developing machine learning models due to the aforementioned issues regarding sharing PHI. Sharing data in centralized learning could potentially lead to data leakage [20]. Patients' data leakage is also an apparent issue in centralized learning [20]. Even if an organization takes steps to anonymize and control the access to the PHI records, one could still potentially re-identify a patient in the data set from certain features[21]. Besides the privacy challenges related to collecting data, centralized learning also poses computational challenges as one would need powerful central servers in both computational and storage capacity to analyze large centralized datasets.

Similar privacy concerns apparent in the medical field are also apparent in agriculture. Accurate segmentation of wheat heads is essential for ensuring optimal growing conditions and yields. Machine learning models have the potential to improve the accuracy and efficiency of this process, which can ultimately have a positive impact on global food production. However, this agricultural data that ensures optimal growing conditions is stored in different institutions, corporations or even the farmers who are hesitant to share this data [25]. The lack of regulatory frameworks around the collection, sharing and use of agricultural data and general concerns regarding privacy and security of this data [27] further leads to farmers not wanting to share their data.

Federated learning (FL) is an approach that aims to solve the privacy and computational limitations posed by centralized learning. Built on data parallelism, McMahan et al. first proposed the concept of FL in 2016 [14]. Instead of aggregating client data into a shared

central server for analysis, in FL, a local model is trained on local data from each center. These models are then aggregated to build a global model, eliminating the need to share data across different centers. This aggregation of the local model updates allows FL to lower the risk of disclosing the private data of the clients. Due to the distributed nature of FL, computation limitations can also be addressed by allowing multiple centers to contribute to the training of the same model. Consequently, FL allows for developing efficient models by utilizing a large number of samples from different centers for domains where data sharing is not practical. The agricultural domain is one such domain.

In this research, we will implement several FL approaches for segmenting wheat heads. We compare the performance of models built using these FL approaches with models built only using local data. Also, a model will be developed using all of the data in a centralized way to measure the performance loss utilizing FL. To build models using FL as well as a centralized approach, we simulate such scenarios using public data from different centers. The expected contribution of this work is to find the most efficient FL approach for medical image segmentation as well as to gain an estimate of performance loss for scenarios where centralized learning cannot be applied.

2 RELATED WORK

The key principle behind FL approaches is determining the optimal global model from the local models trained on local datasets. As outlined in [18], in FL, the requirements for building local models for a specific task are first determined. These include selecting the model architecture that the participating clients will use for local training. Clients can either be randomly chosen or selectively chosen by the aggregation server [19] depending on the computational requirements of the chosen model. Selective client participation allows the aggregation server to potentially optimize the FL process by overcoming bottleneck issues resulting from different device capabilities to run the models. In each round, after selecting the clients, the aggregation server sends the initial global model to the clients. The clients then start training this model on their own local data. After each round of local training, which may vary depending on the client's computing capabilities, the updated model will be sent back to the aggregation server, where the local models are aggregated to generate a new global model. Even though FL reduces network congestion as it avoids transmitting raw medical data [10], some network cost is still incurred from these communication rounds between the aggregation server and clients. Figure 1 provides a schematic overview of FL.

The aggregation of the local models is done at the aggregation server with an algorithm such as FedAvg, proposed by McMahan et al. [14]. In FedAvg, a weighted average of the local models is calculated as the weights associated with the new global model. The weight used for each local model is the ratio of the number of samples used for training that local model to the total number of samples used by all clients. After calculating the new global model, it will be sent back to the clients for another training round. The process ends when an acceptable performance is achieved or the global loss function converges.

Zhao et al. [28] performed an experimental study evaluating FL on datasets that are not independent and identically distributed (IID). Using convolutional neural networks (CNNs), they confirmed that FedAvg achieved centralized learning (stochastic gradient descent)-level accuracy when the data is IID. However, when the data was highly skewed non-IID, they observed an accuracy drop of up to 55%.

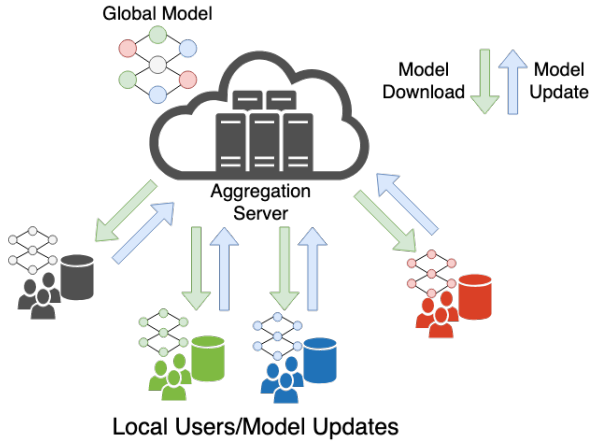

Figure 1: Federated learning algorithm overview.

FedProx [13] was a response to system and statistical heterogeneity that is a part of federated systems. Li et al. [12] stated that the local model updates may be inconsistent with the global model as the distribution of the local dataset can vary from the global distribution. This is what causes the FedAvg algorithm to not perform well under non-IID data. FedProx introduces a regularization term in order to reduce the distance between the local and global models due to non-IID distribution shifts.

3 METHODOLOGY

In this section we will go over the data used for this study, which model architecture will be used, and an overview of FL techniques used in this study. The model architecture will be consistent throughout the study. Both FL approaches, FedAvg and FedProx, will use the model architecture. The centralized approach will use the model as a baseline comparison.

3.1 Data

The data used for this project comes from the Global Wheat Head Dataset (GWHD) dataset [6] [7] [8], which consists of 6,000 images of wheat heads from 5 countries and 18 different domains of different plant growth stages. The dataset was created by the University of Saskatchewan and the University of Tokyo and is publicly available on Zenodo. More specifically, we took a subset of the GWHD dataset that was annotated with segmentation labels. The annotation process was already performed by Najafian et al. [17] and the annotations are available on the University of Saskatchewan's website (https://www.cs.usask.ca/ftp/pub/whs/). The annotations include pixel-level segmentation of wheat heads, including the stem, glumes, and kernels. The subset of the data

used for this project includes 2 manually segmented images from each of 18 different domains, for a total of 36 images. One set of the manually segmented images was set aside as the training dataset while the other as the validation dataset. The decision to use only 2 images per domain was based on resource constraints, as manual segmentation is a time-consuming process. However, this sample size is deemed sufficient for model training based on similar tasks performed with similar dataset sizes in Najafian et al. [17]. Examples of the wheat head images, and their corresponding annotations are shown in Figure 2. For the FL task, it is important to note that

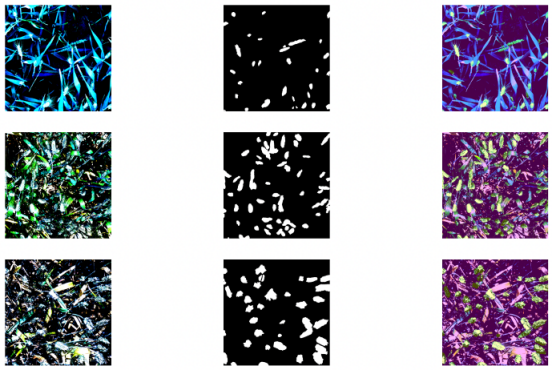

Figure 2: Example of the wheat head images and their corresponding annotations. The first column shows the original wheat head images, the second column shows the pixel-level segmentation annotations, and the third column shows the visual overlay of the original images and the segmentation annotations.

the dataset includes wheat head images from multiple domains, including different plant growth stages and countries. This variety in the dataset will help simulate different institutions with different domains participating in the FL process. The 18 different images are the 18 different private datasets that the institutions will use. Due to the limited size of the training and validation dataset, we add further data augmentation to each dataset to increase variability. The original plan was to rotate the image 360 times and apply strong data augmentation to it. However, due to time constraints and model training time, we went with 36 augmented images at each institution's dataset. All data augmentations applied in this project use the Albumentations package [5]. Augmentation techniques include resizing the images to a resolution of 1024x1024 pixels to reduce computational complexity during training. Further augmentation techniques make use of the Albumentation package to transform the training data. Some transformations include Flip, ColorJitter, ElasticTransform, GridDistortion, RGBShift, Solarize, RandomGamma, GaussianBlur, RandomRain, GaussNoise, Rotate, RandomCrop, and a Normalization. These techniques help the model in each institution generalize better to new data and improve its performance.

In order to evaluate the FL pipeline, we also utilize 365 manually segmented images from teh GWHD dataset annotated by Najafian et al. [17]. This represents 10% of the training subset of the GWHD dataset.

3.2 Materials and Model Architecture

For the task of wheat head instance segmentation, we use a customized U-Net model architecture proposed by Ronneberger et al. [22] with the EfficientNet B4 [24] encoder pretrained on the ImageNet dataset [9] [11]. The U-Net architecture is widely used in biomedical image segmentation tasks due to its ability to effectively capture spatial features. The architecture consists of an encoder and a decoder network, which work together to perform segmentation. The encoder network down-samples the image and extracts high-level features, while the decoder network up-samples the image and produces a segmentation mask. A depiction of this architecture can be seen in Figure 3. The input to the network is an RGB wheathad image of size (3, 1024, 1024) where height and width are 1024. We resized the images to a resolution of 1024x1024 pixels to reduce computational complexity during training. The encoder is based on the EfficientNet B4 architecture. The EfficientNet B4 model is pre-trained on the ImageNet dataset, which consists of over one million labeled images spanning 1,000 categories [24]. The encoder is used to extract features from the input image at different levels of abstraction. The encoder depth is 5 and we use the ImageNet weights. The modification of the encoder being replaced with the EfficientNetB4 architecture allows the model to leverage the power of transfer learning by using a pre-trained encoder to extract features from the input image. The U-Net architecture is well-suited to the wheat head instance segmentation task, as it enables precise localization and segmentation of the wheat heads.

Training processes for this project includes the training of the model in each institution as well as the averaging of the global model. As this is a segmentation task, the loss functioned used for model training is a summation of Binary Cross-entropy loss and Dice loss functions. BCE computes the cross-entropy loss for each pixel allowing us to handle the imbalance of the wheat heads and the background grass, leaves, stems, dirt, etc. Dice loss is sensitive to small object, making it ideal for the wheat heads. We aim to take the benefits of both loss functions for the wheat head segmentation task. As defined in Najafian et al. [17], the summation of Binary Cross-entropy and Dice loss functions can be defined as follows:

$$\text{Loss} = \frac{1}{\|\Omega\|} \sum_{\Omega_i \in \Omega} \left(1 - \frac{\left(2 \sum_{x \in \Omega_i} p(x) g(x)\right) + \epsilon}{\left(\sum_{x \in \Omega_i} p(x) + \sum_{x \in \Omega_j} g(x)\right) + \epsilon} - \right. \tag{1}$$

$$\left. \frac{1}{\|\Omega_i\|} \times \sum_{x \in \Omega_i} \left(g(x) \log(p(x)) + (1 - g(x)) \log(1 - p(x))\right) \right) \tag{2}$$

To evaluate the model's performance we make use of commonly used Intersection over Union (IoU) and Dice score for this segmentation task. Given an observed segmentation mask O and the expected segmentation mask E, the IoU is given as the following equation:

$$\text{IoU}(O, E) = \frac{\|O \cap E\|}{\|O \cup E\|} \tag{3}$$

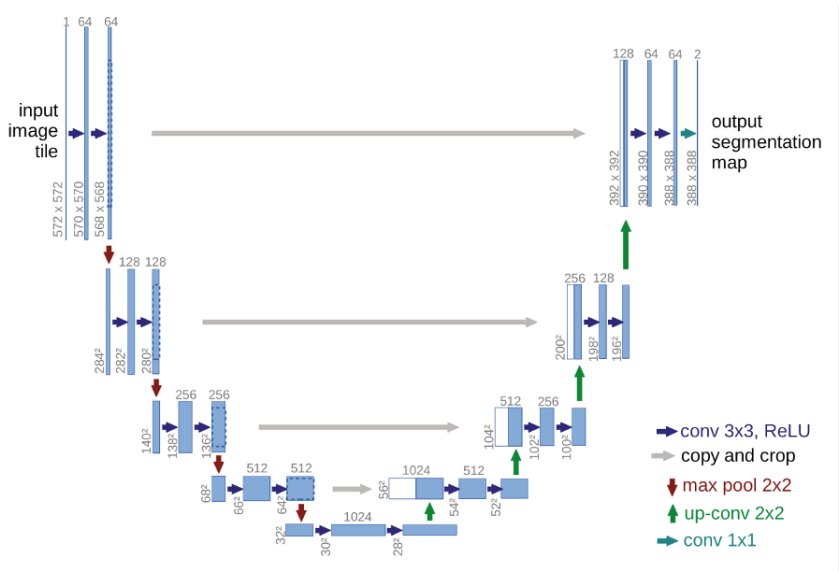

Figure 3: A depiction of the U-Net architecture as per Ronneberger et al. [22]. The arrows represent the different operations that are performed in the model.

while the Dice score is as follows:

$$\text{Dice}(O, E) = \frac{2\|O \cap E\|}{\|O\| + \|E\|} \tag{4}$$

Since the aggregation of the models happens at the central server after they send their weight updates, it was decided to evaluate the performance of the models by calculating the Dice and IoU score of the model on a test dataset stored in the central server after each communication round. This test dataset is the 365 manually annotated images from the GWHD dataset. This gave us a better guideline of how the system is performing (global model performance) rather than evaluating client-by-client performance. The local models are saved after each communication round, allowing us the option to evaluate local performance over time when compared to the central test dataset.

An SGD optimizer [23] was used for the FedAvg experiments with a learning rate of 0.01. For the FedProx experiment, the FedProx optimizer [13] was used to perform local updates with a μ of 0.01.

3.3 Model Architecture and Hardware Specifications

The two FL approaches, FedAvg and FedProx, and the centralized approach trained on 16 cores of the 4x Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz (Skylake, 2019) with a x GA100 A100 PCIe 80GB GPU. Figure 7 is the detailed model architecture used, it is in Appendix A. These models were trained using Pytorch.

3.4 Federated Learning

Federated learning methods [14] have an aggregation server that coordinates the global learning objective from connected clients/institutions. The aim for FL methods is to minimize the following function:

$$\min_w f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \quad \text{where} \quad F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w) \tag{5}$$

where w is the model parameters, K is the number of clients, P_k is the set of indices of data at client k (the partition), and $n_k = |\mathrm{P}_k|$. $f_i(w)$ is the loss of the prediction on sample $(x_i, y_i; w)$ where $f_i(w) = l(x_i, y_i, ; w)$. The psuedocode of FedAvg can be viewed in algorithm 1. This is a formal overview of the federated averging process described in the related work.

FedAvg is one of the FL methods we will be evaluating in this paper. The other is FedProx [13] which aims to solve the shortcomings of FedAvg; mainly it's sensitivity to non-IID data distribution. FedProx introduces a proximal term to the local subproblem to minimize the impact of variable local updates. It updates the minimizing objective to the following equation:

$$\min_w h(w; w^t) = F_k(w) + \frac{\mu}{2}||w - w^t||^2 \tag{6}$$

This term address the issue of non-IID data by restricting local updates to be close to the global model. FedProx also adds a hyperparameter μ.

4 RESULTS

The configurations for the experiments were chosen to be able to compare the different approaches. The FedAvg and FedProx approach were ran for 20 communication rounds and all 18 institutions were selected for the training process. Due to resource limitations and the need to have a fair comparison, the centralized approach was only ran for 100 epochs. The same hyperparameters were kept for each underlying model in each experiment. An SGD optimizer with a learning rate of 0.01, a weight decay of 0.00001, and a momentum value of 0.95 was chosen. A batch size of 4 was kept to

Algorithm 1 Psuedocode for FedAvg and FedProx. The difference is in the client update. There are K clients indexed by k. B is the local batchsize. E is the local epochs. η is the learning rate. w_0 is the global model.

Server Executes:
initialize w_0
for each communication round $t = 1, 2..$ **do**
  send current global model w_{t-1} to all connected clients
  for each selected client k **do**
    Receive update w_{t+1}^k from each client k
    $w_{t+1}^k \leftarrow ClientUpdate(k, w_t)$
  end for
  Aggregate and update global model
  $w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} w_t^k$
end for
Client Executes: $ClientUpdate(k, w)$ on client k
if FedAvg **then** $L(w; b) = F_k(w)$
end if
if FedProx **then** $L(w; b) = F_k(w) + \frac{\mu}{2}||w - w^t||^2$
end if
initialize local model with w
for each local epoch i from 1 to E **do**
  for batch $b \in B$ **do**
    $w \leftarrow w - \eta \nabla L(w; b)$
  end for
end for
send w back to the server

reduce the memory requirements of the model. For FedProx, a μ value of 0.01 was chosen. Overfitting techniques, hyperparameter tuning, and in-depth optimizations were not explored in detail in this study. The evaluation pipeline taken to get the results can be seen in Figure 4.

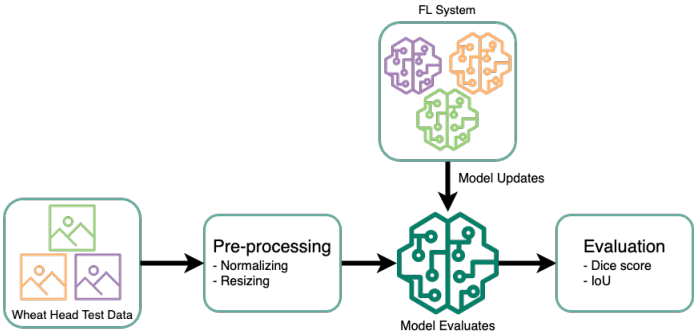

Figure 4: A visualization of the the evaluation pipeline.

Training time for the FL approaches took approximately 8 hours for 20 rounds of communication when all 18 clients were selected. This is longer than the training time of the centralized approach which was at 3 hours. This may be due to the lack of parallelization in the FL approaches as we ran the experiments in a sequential manner because of resource constraints. A plot of the metrics over epoch/communication round can be seen in Figure 5 and Figure 6.

The centralized approach depicts the training and validation metrics/loss of the model over 100 epochs. There is still room for some improvement for the model. However, we would either have to acquire more data, explore a more sophisticated image synthesis technique to generate more data, or other underfittnig techniques. This is the baseline model that we will compare our FL approaches to. The FL metric plot depicts the validation metrics of the global model after each communication round. The loss trend lines of Figure 5 may indicate that both approaches could have improved on the model performance given more communication rounds.

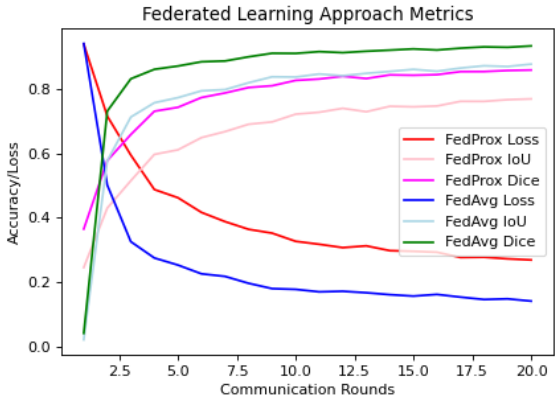

Figure 5: FedAvg and FedProx metrics over 20 communication rounds.

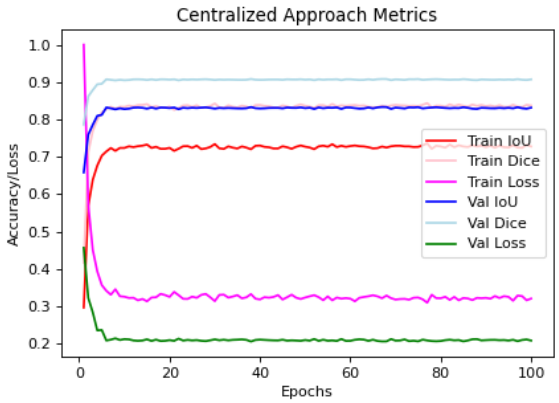

Figure 6: Centralized approach train/validation metrics.

The performance measures of the resulting models from the FedAvg and FedProx FL approaches can be seen in Table 1 and Table 2. The 'Central' column indicates the model trained using only one institution where the entire training and validation dataset is available for model training. Both tables show the IoU and Dice score on the test dataset of 365 annotated images from the GWHD dataset. These images were not introduced in the training process, but 36 different images from the same domains were used. On

the entire test dataset, FedAvg achieved an IoU of 0.8788 and a Dice score of 0.9355. FedProx achieved an IoU of 0.7706 and a Dice score of 0.8605. The centralized approach achieved an IoU of 0.8173 and a Dice score of 0.8904. These results show that FedAvg has comparable performance to a traditional, centralized learning approach with the added benefits of some degree of data privacy. Although we were not able to verify this, we would expect that if the centralized approach only had access to one institution's data subset, the performance of the centralized approach would not be as strong.

Even though FedProx did not match the perfomance of FedAvg or the centralized approach as closely, it did match or even beat FedAvg when evaluating by institutions. FedProx had better performance than FedAvg in Arvalis3, Arvalis6, Arvalis7, and Arvalis8 in Table 1 and Arvalis10, Arvalis12, ETHZ1, Inrae1, NMBU1, NMBU2, and ULiege in Table 2. This indicates that the proximal term in FedProx has helped the FL approach adapt to variations between data, allowing it to handle these different domains better than FedAvg. It is worth noting that the FedAvg results in NMBU2 are missing due to an error in data collection/coding.

5 DISCUSSION

In this study we implemented and evaluated a FL pipeline for wheat head instance segmentation. We sought to solve the privacy and computational limitations posed by traditional centralized learning techniques and to improve the accuracy and efficiency of wheat head instance segmentation. Specifically, we compared two different FL approaches (FedAvg, FedProx) and evaluate and compare their performance to a traditional, centralized learning technique. The project will used data from the GWHD, which a subset was further augmented using data augmentation techniques to increase the sample size. The expected outcome of this study was to produce a FL pipeline for wheat head instance segmentation that matches or outperforms traditional centralized learning techniques in terms of accuracy and efficiency. The project also sought to identify the most effective federated learning approach for this task. In terms of achieving the first goal, we cannot definitively say that FL techniques outperform centralized approaches for this task. The results in Table 1 and Table 2 show that both FL approaches match the centralized approach on most of the selected domains. However, due to some limitations that will be discussed in the following section, it would not be a fair statement that FL outperforms centralized techniques. For the second goal, we were able to identify the more effective Fl approach for wheathead segmentation. Since this experiment focused on non-IID data, evident by the different samples in each domain, we were able to show how FedProx can improve on FedAvg in this scenario. Even though FedProx did not perform as well as FedAvg when considering the entire test dataset, its ability to improve on different domains over FedAvg makes it a good choice for this task. However, there are various scenarios and FL approaches that were not explored in this study that can impact this judgment.

5.1 Limitations and Future Work

One limitation of this study is the limited experiments performed to evaluate these different approaches. Ideally, a comparison of IID data would have helped in understanding the impact that non-IID data has on FL techniques. Furthermore, client selection fixed to the 18 institutions. The server having the ability to select which institutions can participate in the training processes can show how FL techniques improve on centralized approaches (dropping slow institutions that are taking a long time to send an update) or not improve (selection of clients can introduce variability in the distribution of the detests).

Another limitation in this study is the limited amount of experiment runs that were performed. Since we had to train 18 different models in the task of instance segmentation, where each model had 25,257,449 trainable parameters, resource constraints were a bottleneck. Local epochs and communication rounds were kept low so that the training time would not increase. This resulted in the aforementioned 8 hour training time for either of the FL approaches. As seen in Figure 5, the FL approaches could train for more rounds to potentially improve on their performance. Techniques to improve on this limitation include parallelizing local modal training so that each client computes their update concurrently. Client selection can improve on this limitation as well. Reducing the number of clients participating in these experiments, or only selecting a fraction of them each time, could improve on the resource constraints. These improvements can help in repeating more experiments, resulting in better quality data and inferences on it.

In terms of future work in this area, comparisons of different models for different tasks using this pipeline. One such task would be utilizing a widely used medical imaging dataset called the Brain Tumor Segmentation (BraTS) dataset [1–4, 15] to evaluate the performance of machine learning algorithms for the segmentation of brain tumors in MRI scans. Adding more state-of-the-art FL approaches such as FedMa [26] or more recently FedProx [16] which has improved on the effect that non-IID data has on FL systems.

6 CONCLUSION

The study aimed to address the privacy and computational limitations of centralized learning approaches by implementing and evaluating a FL pipeline for wheat head instance segmentation. The study compared two different FL approaches (FedAvg and FedProx) and evaluated and compared their performance to a traditional centralized learning technique. The results showed that both FL approaches matched the centralized approach on most of the selected domains, but it was not a fair statement to say that FL outperforms centralized techniques. However, the study identified the more effective FL approach for wheat head segmentation in a high data variability scenario, which was FedProx. The study had limitations in terms of the limited experiments performed to evaluate different approaches and the resource constraints, but improvements can be made to repeat more experiments for better quality data and inferences. Future work includes comparisons of different models for different tasks using this pipeline and adding more state-of-the-art FL approaches.

A MODEL ARCHITECTURE

REFERENCES

[1] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. 2017. Advancing The Cancer Genome Atlas glioma MRI collections with expert

Institution		FedAvg		FedProx		Central	
		IoU	Dice	IoU	Dice	IoU	Dice
Arvalis1		0.7878	0.8813	0.7563	0.8602	0.7722	0.8705
Arvalis2		0.8453	0.9162	0.7902	0.8821	0.8164	0.8987
Arvalis3		0.7345	0.8469	0.7657	0.8668	0.8367	0.9109
Arvalis4		0.7319	0.8452	0.6740	0.7999	0.6958	0.8162
Arvalis5		0.7689	0.8694	0.6183	0.7152	0.6568	0.7431
Arvalis6		0.7799	0.8763	0.8331	0.9087	0.8998	0.9472
Arvalis7		0.7643	0.8664	0.8115	0.8958	0.8395	0.9127
Arvalis8		0.6815	0.8106	0.7771	0.8728	0.8143	0.8970
Arvalis9		0.7750	0.8733	0.7473	0.8548	0.7939	0.8850

Table 1: A table of the performance of the FL approaches FedAvg and FedProx on 9 institutions. Central is the traditional centralized approach where the model is trained at a single institution with access to all of the training dataset. The performance of the resulting models is measured on the test dataset of 365 annotated images from the GWHD dataset. N/A entries occured to do a coding error.

segmentation labels and radiomic features. *Sci Data* 4 (9 2017), 170117. https://doi.org/10.1038/sdata.2017.117

[2] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. 2017. *Segmentation Labels and Radiomic Features for the Pre-operative Scans of the TCGA-LGG collection [Data Set].* https://doi.org/10.7937/K9/TCIA.2017.GJQ7R0EF

[3] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. 2017. *Segmentation Labels for the Pre-operative Scans of the TCGA-GBM collection [Data set].* https://doi.org/10.7937/K9/TCIA.2017.KLXWJJ1Q

[4] Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, et al. 2018. Identifying the Best Machine Learning Algorithms for Brain Tumor Segmentation, Progression Assessment, and Overall Survival Prediction in the BRATS Challenge. (2018). https://doi.org/10.48550/ARXIV.1811.02629

[5] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. 2020. Albumentations: Fast and Flexible Image Augmentations. *Information* 11, 2 (2020). https://doi.org/10.3390/info11020125

[6] Etienne David, Simon Madec, Pouria Sadeghi-Tehran, Helge Aasen, Bangyou Zheng, Shouyang Liu, Norbert Kirchgessner, Goro Ishikawa, Koichi Nagasawa, Minhajul A., et al. 2020. Global Wheat Head Detection (GWHD) Dataset: A Large and Diverse Dataset of High-Resolution RGB-Labelled Images to Develop and Benchmark Wheat Head Detection Methods. *Plant Phenomics* 2020 (2020). https://doi.org/10.34133/2020/3521852 arXiv:https://spj.science.org/doi/pdf/10.34133/2020/3521852

[7] Etienne David, Mario Serouart, Daniel Smith, Simon Madec, Kaaviya Velumani, Shouyang Liu, Xu Wang, Francisco Pinto Espinosa, Shahameh Shafiee, Izzat S. A. Tahir, et al. 2021. Global Wheat Head Dataset 2021: more diversity to improve the benchmarking of wheat head localization methods. arXiv:2105.07660 [cs.CV] https://arxiv.org/abs/2105.07660

[8] Etienne David, Mario Serouart, Daniel Smith, Simon Madec, Kaaviya Velumani, Shouyang Liu, Xu Wang, Francisco Pinto, Shahameh Shafiee, Izzat S. A. Tahir, et al. 2021. Global Wheat Head Detection 2021: An Improved Dataset for Benchmarking Wheat Head Detection Methods. *Plant Phenomics* 2021 (2021). https://doi.org/10.34133/2021/9846158 arXiv:https://spj.science.org/doi/pdf/10.34133/2021/9846158

[9] Pavel Iakubovskii. 2019. Segmentation Models Pytorch. https://github.com/qubvel/segmentation_models.pytorch Publication Title: GitHub repository.

[10] Latif U. Khan, Shashi Raj Pandey, Nguyen H. Tran, Walid Saad, Zhu Han, Minh N. H. Nguyen, and Choong Seon Hong. 2020. Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism. *IEEE Communications Magazine* 58, 10 (2020), 88–93. https://doi.org/10.1109/MCOM.001.1900649

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60, 6 (May 2017), 84–90. https://doi.org/10.1145/3065386 Place: New York, NY, USA Publisher: Association for Computing Machinery.

[12] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. 965–978. https://doi.org/10.1109/ICDE53745.2022.00077

[13] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks, I Dhillon, D Papailiopoulos, and V Sze (Eds.). *Proceedings of Machine Learning and Systems* 2, 429–450. https://proceedings.mlsys.org/paper/2020/file/38af86134b65d0f10fe33d30dd76442e-Paper.pdf

[14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. *20th International Conference on Artificial Intelligence*

Institution		FedAvg		FedProx		Central	
		IoU	Dice	IoU	Dice	IoU	Dice
Arvalis10		0.7225	0.8389	0.8777	0.9341	0.9089	0.9515
Arvalis11		0.6635	0.7977	0.6170	0.7528	0.8723	0.9316
Arvalis12		0.7349	0.8472	0.7516	0.8580	0.7759	0.8737
ETHZ1		0.7882	0.8816	0.8342	0.9093	0.8487	0.9180
Inrae1		0.7982	0.8878	0.8464	0.9166	0.8919	0.9427
NMBU1		0.6558	0.7922	0.7324	0.8445	0.8431	0.9147
NMBU2		N/A	N/A	0.6837	0.7820	0.7693	0.8362
Rres1		0.8789	0.9355	0.8370	0.9105	0.8940	0.9439
ULiege		0.6535	0.7905	0.6793	0.8085	0.8441	0.9154

Table 2: A table of the performance of the FL approaches FedAvg and FedProx on the remaining 9 institutions. Central is the traditional centralized approach where the model is trained at a single institution with access to all of the training dataset. The performance of the resulting models is measured on the test dataset of 365 annotated images from the GWHD dataset. N/A entries occured to do a coding error.

and Statistics (AISTATS), 1273–1282.

[15] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Yuliya Kirby Justin, Burren, Nicole Porz, et al. 2014. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Trans Med Imaging* 34 (12 2014), 1993–2024. Issue 10. https://doi.org/10.1109/TMI.2014.2377694

[16] Xutong Mu, Yulong Shen, Ke Cheng, Xueli Geng, Jiaxuan Fu, Tao Zhang, and Zhiwei Zhang. 2023. FedProc: Prototypical Contrastive Federated Learning on Non-IID data. *Future Generation Computer Systems* 143 (2023), 93–104. https://doi.org/10.1016/j.future.2023.01.019

[17] Keyhan Najafian, Alireza Ghanbari, Mahdi Sabet Kish, Mark Eramian, Gholam Hassan Shirdel, Ian Stavness, Lingling Jin, and Farhad Maleki. 2023. Semi-Self-Supervised Learning for Semantic Segmentation in Images with Dense Patterns. *Plant Phenomics* 0 (2023). https://doi.org/10.34133/plantphenomics.0025 arXiv:https://spj.science.org/doi/pdf/10.34133/plantphenomics.0025

[18] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. 2022. Federated Learning for Smart Healthcare: A Survey. *ACM Comput. Surv.* 55 (2 2022). Issue 3. https://doi.org/10.1145/3501296

[19] T Nishio and R Yonetani. 2019. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 1–7. https://doi.org/10.1109/ICC.2019.8761315

[20] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. 2020. The future of digital health with federated learning. *NPJ Digital Medicine* 3, 1 (2020), 1–7. https://doi.org/10.1038/s41746-020-00323-1

[21] Luc Rocher, Julien M Hendrickx, and Yves-Alexandre de Montjoye. 2019. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature Communications* 10 (2019), 3069. Issue 1. https://doi.org/10.1038/s41467-019-10933-3

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (Eds.). Springer International Publishing, Cham, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

[23] Sebastian Ruder. 2017. An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs.LG] https://arxiv.org/abs/1609.04747

[24] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 6105–6114. https://proceedings.mlr.press/v97/tan19a.html

[25] Simone van der Burg, Leanne Wiseman, and Jovana Krkeljas. 2021. Trust in Farm Data Sharing: Reflections on the EU Code of Conduct for Agricultural Data Sharing. *Ethics and Inf. Technol.* 23, 3 (sep 2021), 185–198. https://doi.org/10.1007/s10676-020-09543-1

[26] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. 2020. Federated Learning with Matched Averaging. *8th International Conference on Learning Representations (ICLR)* (2020). arXiv:2002.06440 [cs.LG] https://openreview.net/pdf?id=BkluqlSFDS

[27] Leanne Wiseman, Jay Sanderson, Airong Zhang, and Emma Jakku. 2019. Farmers and their data: An examination of farmers' reluctance to share their data through the lens of the laws impacting smart farming. *NJAS - Wageningen Journal of Life Sciences* 90-91 (2019), 100301. https://doi.org/10.1016/j.njas.2019.04.007

[28] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated Learning with Non-IID Data. (2018). arXiv:1806.00582 [cs.LG] https://arxiv.org/abs/1806.00582

```
==========================================================================================
Layer (type:depth-idx)                                  Output Shape              Param #
==========================================================================================
Unet                                                    [4, 1, 1024, 1024]        --
├─EfficientNetEncoder: 1-1                             [4, 3, 1024, 1024]        806,400
│    └─Conv2dStaticSamePadding: 2-1                     [4, 48, 512, 512]         1,296
│    │    └─ZeroPad2d: 3-1                              [4, 3, 1025, 1025]        --
│    └─BatchNorm2d: 2-2                                 [4, 48, 512, 512]         96
│    └─MemoryEfficientSwish: 2-3                        [4, 48, 512, 512]         --
│    └─ModuleList: 2-4                                  --                        --
│    │    └─MBConvBlock: 3-2                            [4, 24, 512, 512]         2,940
│    │    └─MBConvBlock: 3-3                            [4, 24, 512, 512]         1,206
│    │    └─MBConvBlock: 3-4                            [4, 32, 256, 256]         11,878
│    │    └─MBConvBlock: 3-5                            [4, 32, 256, 256]         18,120
│    │    └─MBConvBlock: 3-6                            [4, 32, 256, 256]         18,120
│    │    └─MBConvBlock: 3-7                            [4, 32, 256, 256]         18,120
│    │    └─MBConvBlock: 3-8                            [4, 56, 128, 128]         25,848
│    │    └─MBConvBlock: 3-9                            [4, 56, 128, 128]         57,246
│    │    └─MBConvBlock: 3-10                           [4, 56, 128, 128]         57,246
│    │    └─MBConvBlock: 3-11                           [4, 56, 128, 128]         57,246
│    │    └─MBConvBlock: 3-12                           [4, 112, 64, 64]          70,798
│    │    └─MBConvBlock: 3-13                           [4, 112, 64, 64]          197,820
│    │    └─MBConvBlock: 3-14                           [4, 112, 64, 64]          197,820
│    │    └─MBConvBlock: 3-15                           [4, 112, 64, 64]          197,820
│    │    └─MBConvBlock: 3-16                           [4, 112, 64, 64]          197,820
│    │    └─MBConvBlock: 3-17                           [4, 112, 64, 64]          197,820
│    │    └─MBConvBlock: 3-18                           [4, 160, 64, 64]          240,924
│    │    └─MBConvBlock: 3-19                           [4, 160, 64, 64]          413,160
│    │    └─MBConvBlock: 3-20                           [4, 160, 64, 64]          413,160
│    │    └─MBConvBlock: 3-21                           [4, 160, 64, 64]          413,160
│    │    └─MBConvBlock: 3-22                           [4, 160, 64, 64]          413,160
│    │    └─MBConvBlock: 3-23                           [4, 160, 64, 64]          413,160
│    │    └─MBConvBlock: 3-24                           [4, 272, 32, 32]          520,904
│    │    └─MBConvBlock: 3-25                           [4, 272, 32, 32]          1,159,332
│    │    └─MBConvBlock: 3-26                           [4, 272, 32, 32]          1,159,332
│    │    └─MBConvBlock: 3-27                           [4, 272, 32, 32]          1,159,332
│    │    └─MBConvBlock: 3-28                           [4, 272, 32, 32]          1,159,332
│    │    └─MBConvBlock: 3-29                           [4, 272, 32, 32]          1,159,332
│    │    └─MBConvBlock: 3-30                           [4, 272, 32, 32]          1,159,332
│    │    └─MBConvBlock: 3-31                           [4, 272, 32, 32]          1,159,332
│    │    └─MBConvBlock: 3-32                           [4, 448, 32, 32]          1,420,804
│    │    └─MBConvBlock: 3-33                           [4, 448, 32, 32]          3,049,200
├─UnetDecoder: 1-2                                     [4, 32, 1024, 1024]       --
│    └─Identity: 2-5                                    [4, 448, 32, 32]          --
│    └─ModuleList: 2-6                                  --                        --
│    │    └─DecoderBlock: 3-34                          [4, 512, 64, 64]          5,163,008
│    │    └─DecoderBlock: 3-35                          [4, 256, 128, 128]        1,899,520
│    │    └─DecoderBlock: 3-36                          [4, 128, 256, 256]        479,744
│    │    └─DecoderBlock: 3-37                          [4, 64, 512, 512]         138,496
│    │    └─DecoderBlock: 3-38                          [4, 32, 1024, 1024]       27,776
├─SegmentationHead: 1-3                                [4, 1, 1024, 1024]        --
│    └─Conv2d: 2-7                                      [4, 1, 1024, 1024]        289
│    └─Identity: 2-8                                    [4, 1, 1024, 1024]        --
│    └─Activation: 2-9                                  [4, 1, 1024, 1024]        --
│    │    └─Identity: 3-39                              [4, 1, 1024, 1024]        --
==========================================================================================
Total params: 25,257,449
Trainable params: 25,257,449
Non-trainable params: 0
Total mult-adds (G): 596.74
==========================================================================================
Input size (MB): 50.33
Forward/backward pass size (MB): 19675.48
Params size (MB): 31.32
Estimated Total Size (MB): 19757.13
==========================================================================================
```

Figure 7: Model architecture.