Federated Learning for Image Classification Across Different Modalities

Client
Dr Vera Chung, vera.chung@sydney.edu.au

Tutor
Muhammad atif Iqbal, muhammadatif.iqbal@sydney.edu.au

Group members

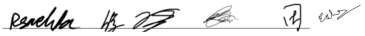
Rajiv Mehta	rmeh0608	540771859
Ngai Chun Fu	ngfu0299	540363470
David Cortés Sánchez	vcor0924	500623734
Eshan Arora	earo0293	540425550
Mohamad Akmal bin Abu Bakar	mbin0316	530432780
HanWen Tian	htia0294	540244065

1 Contribution Statement

Our group CS54-2, taking project titled Federated Learning for Image Classification Across Different Modalities, with group members Rajiv Mehta, Ngai Chun Fu, David Cortés Sánchez, Eshan Arora, Mohamad Akmal bin Abu Bakar and HanWen Tian, would like to state the contributions that each group member has made for this project during this semester:

- **Rajiv Mehta**:
  - Led documentation and reporting including the project proposal and final report sections (Introduction, Literature Review, Methodology Phase 2, Phase 2 Results and Discussion, Schedule, Milestones).
  - Project managed the group including the scoping and planning of the project, resource allocation, running project status meetings, Kanban board and Gantt Chart updates and client communication.
  - Scripted recorded and edited both the final group presentation and project demo presentation.
  - Contributed to literature reviews and all project deliverables through the semester including group progress reports and status checking forms.
- **David Cortes Sanchez**:
  - Conducted research and implementation across all phases.
  - Led the development of Phase 3.
  - Managed dataset exploration and federated learning experiments.

  - Supported setup of collaboration tools and backlog creation.
- **Ngai Chun Fu**:
  - Focused on model development and debugging including VGG11 implementation.
  - Tested models on Google Cloud and supported collaboration infrastructure.
  - Led the development of Phase 1.
  - Managed dataset exploration and federated learning experiments.
- **Hanwen Tian Tian**:
  - Contributed to literature reviews and project scope definition.
  - Authored final report sections including Methodology Phase 1, Limitations and Future Works, and Conclusion.
  - Supported research consolidation and report writing.
- **Akmal Bin Mohamed Abu Bakar**:
  - Contributed to literature reviews and project proposal sections (Implications, Data Analysis).
  - Authored final report sections including Methodology Phase 3, Project Definition, and Resources.
  - Participated in group progress reporting.
  - Helped in data research and EDA for Phase 3.
- **Eshan Arora**:
  - Led implementation of Phase 2 including custom CNN and custom FedAvg setup.
  - Conducted dataset research and testing.
  - Supported tutorials, initial experiments, and formatting tasks.
  - Communication with client and scoping of project.

All group members agreed on the contributions listed on this statement by each group member. Signatures: 

Abstract

Federated Learning (FL) has been studied as an approach to deal with privacy concerns in machine learning, especially in domains such as healthcare, where sharing data is often not even legal. This project proposes a multi-modal federated learning framework with the objective of enhancing machine learning model performance in medical imaging, with a particular focus on the issues produced by modality and class heterogeneity across clients. Initially, the framework is designed to work with clients possessing identical modalities and classes, before involving varied modalities and overlapping or distinct class sets across clients. The project seeks to evaluate the effectiveness of federated learning in medical image classification tasks, comparing its performance with centralised models in terms of accuracy and robustness, and analysing performance in local and global scale. By utilising publicly available medical image datasets, the research aims to explore how federated learning can perform collaborative model training while preserving patient privacy.

Contents

2 Introduction

In a world where security and privacy are at the forefront of minds, federated learning has become an increasingly important methodology as it enables machine learning models to maintain data privacy, empowering organisations to harness collective insights without compromising sensitive information. This is particularly important in industries such as the medical field where hospitals often due to legal requirements and patient confidentiality, data cannot be shared between hospitals easily making the process of training machine learning models difficult, halting advancements within this industry. Furthermore, the challenge is compounded by the diversity of data modalities such as different types of imaging, electronic health records and genetic data, all requiring specialised models. This study aims to tackle the challenges faced by the medical industry as the authors propose a new multi-modal federated machine learning model framework.

3 Literature Review

The rapid advancement in federated learning (FL) has transformed the industry of data sharing and model training while also preserving the privacy with decentralised machine learning. With these advancements, it was first important to research studies that related to the project. This showcased that past literature had rarely explored the idea of multi-modal federated learning, leading the research to be split into four main categories, *Foundations of Federated Learning*, *Addressing Data Heterogeneity in Federated Learning*, *Multi-modal Federated Learning* and *Federated Learning in Medical Imaging*.

3.1 Foundations of Federated Learning

'Active Learning Based Federated Learning for Waste and Natural Disaster Image Classification' by Ahmed et.al explores how federated learning can benefit from the training of unlabelled data from clients using active learning modelling techniques [1]. To achieve this, the authors used two datasets the first a collection of natural disasters related images that were sourced from social media platforms. This collection contained 7000 images with 8 different categories, 5000 split into smaller training datasets with 2450 being used to testing with the remainder remaining unlabelled for validation. The second dataset was used as a benchmark dataset coming from a previous paper *'CNN-RNN: A large-scale hierarchical image classification framework'* by Guo et.al which contained 6 categories of different types of waste containing 2527 images [2]. These 2 datasets were then used in the authors proposed AL-based framework which could be split into three main components starting with the feature extraction where they used a pre-trained ResNet model as the experiment isn't focused on how features are initially identified and therefore, should not impact experimental results. This is then followed with AL where one of the small training sets is used plus samples from the unlabelled image pool through different *'Uncertainty Sampling and Query by Committee* parameters. This is done for multiple clients where an LSTM is then used to push the data into a federated model where a *FedAvg* is used to train all five clients. The experiment found that the AL-based models achieved approximately an 86% accuracy with QBC vote Entropy which closely matched the accuracy of the fully manually labelled baseline while greatly exceeding the loosely labelled dataset. AL-based models achieved up to 90.2% accuracy in federated learning, compared to 91.3% for the fully labelled baseline and 86.2% for the loosely labeled dataset. When testing the number of clients it was worth noting that the increased clients caused a 6.1% decline

in accuracy from 2-8 clients due to the fewer training samples, though more samples and AL could help mitigate this, overall showcasing the effectiveness of AL and Federated modelling.

Alam et.all,in their study of *"Enhancing Image Classification with Federated Learning: A Comparative Study of VGG16 and MobileNet on CIFAR-10"*,investigates the use of Federated Learning (FL) across different research areas and tasks specifically in machine with limited computational resources[3]. Alam (2024) provided an analysis of FL techniques for image classification using the CIFAR-10 dataset, employing popular deep learning architectures such as VGG16 and MobileNet. Initial accuracies without FL were established at 74.5% for VGG16 and 70.8% for MobileNet. Further experiments evaluated how productive various FL algorithms are.Among them are FedAvg,FedProx,FedMA,and FedPAQ and they are measured in terms of their accuracy and data privacy. The author reported that while FedAvg ensure robust privacy, this had led to reduced accuracy of 71.1% for VGG16 and 67.5% for MobileNet. This highlights performance issues in distributed scenarios that arent identical. In contrast, algorithms such as FedProx, FedMA, and FedPAQ significantly improved accuracy.FedMA delivered highest accuracy improvements which increased the VGG16 performance to 76.3% while MobileNet to 73.1%. FedProx contributed by restricting local model updates to match to the global model, and FedMA improved accuracy through strategic matching and averaging of layers. The author emphasized that algorithm selection critically impacts the effectiveness of FL frameworks which suggests the need for balancing privacy with performance for practical scalable software for consumer use(Alam,2024).

The paper *A Comprehensive Survey of Privacy-preserving Federated Learning: A Taxonomy, Review, and Future Directions* by Yin et.al, aimed to investigate the security and privacy of federated models and their potential for privacy leakage by introducing a comprehensive survey based on the authors proposed 5W-scenario-based taxonomy [4]. This survey was designed to to systematically analyse privacy risks by identifying who the attackers are, what types of attacks they employ, when and where privacy leakage occurs, and why such attacks are launched. The authors categorised existing PPFL methods into four main approaches: encryption-based (e.g., homomorphic encryption, secure multiparty computation), perturbation-based (e.g., differential privacy, noise injection), anonymisation-based (e.g., k-anonymity, l-diversity), and hybrid techniques that combine multiple strategies. From their analysis into previous studies, key findings included *Gradient leakage* being a major vulnerability, with attacks capable of reconstructing training data while *active attacks* are more damaging than passive ones due to their ability to manipulate training. Also *hybrid methods* require further optimisation to reduce overhead and *privacy-preserving mechanisms* must be tailored to specific FL architectures (horizontal, vertical, transfer learning). These findings indicating that federated learning though increase the security of training data does show potentially issues without the integration of other security measures.

Horizontal Federated Learning of Takagi–Sugeno Fuzzy Rule-Based Models by Zhu et.al aimed to build in the field of federated learning techniques through their implementation of their proposed fuzzy rule-based modelling approach [5]. To test this approach, the authors sourced a synthetic dataset as well as five public datasets sourced from UCI and KEEL repositories with each dataset being split 80-20% for training and testing. The paper proposed a two-phase horizontal FL framework for the fuzzy rule-based model with the first being federated fuzzy clustering which clients compute local gradients based on their data and send them to a central server and the server aggregates gradients to update global prototypes (fuzzy sets). Then the second phase is the parameter optimisation where each fuzzy rule has a local linear function and the client computes the mean square error locally with the clients respective gradient/weights. These MSE is sent to the server

and the gradients are then aggregated to update the global rule parameters. Using this approach it was found that the fuzzy models achieved comparable to local centralised models though could have a slightly higher error due to Non-IID data distribution amongst clients. Though overall, this approach showed advantages by preserving data privacy and is scalable to multiple clients with heterogeneous data.

3.2 Addressing Data Heterogeneity in Federated Learning

Several reviews have been published to cover the topic of Federated Learning in diverse task contexts and research areas. One of them is *'A Review of Federated Learning Methods in Heterogeneous Scenarios'* [6]. The authors highlight the need for distributed collaborative training to address the problem of data scarcity. However, this approach introduces complexity and heterogeneity in Federated Learning scenarios, which affect the efficiency and accuracy of models trained in this setting. This work aims to fill the gap created by the lack of comprehensive and specific reviews on the heterogeneity of FL. The study provides a definition of Federated Learning and establishes the background of convergence theory. Specifically, the authors define five forms of non-IID distribution that influence the convergence process in a Federated Learning setting: label skew, feature skew, same label with different features, different labels with the same features, and quantity skew. They categorise the heterogeneous challenges in Federated Learning into three types: device heterogeneity, data heterogeneity, and model heterogeneity. Additionally, they discuss various approaches to addressing these challenges and evaluate their effectiveness. However, they fall short in providing a detailed comparison of experimental data.

Qu et al. (2022) in their paper*"Rethinking Architecture Design for Tackling Data Heterogeneity in Federated Learning"* explore the impact of system design choices in federated learning (FL), with an emphasis on how more effective the transformer-based architectures, specifically Vision Transformers (ViT), compared to traditional convolutional neural networks (CNNs) [7]. Their study examined CNNs like ResNet and EfficientNet against transformer models such as ViT and Swin Transformers across multiple metrics which includes CIFAR-10, CelebA, and a retina medical imaging dataset. Performance was measured using test accuracy, convergence speed, and communication efficiency. Transformers significantly outperformed CNNs, specifically under heterogenous data. On highly heterogeneous splits of CIFAR-10, ViTS achieved about 37% higher accuracy compared to RESNets. The improved performance is mainly due to catastrophic forgetting in which transformers' are more likely to be susceptible to .This occurs when the model loses previously learned knowledge when training sequentially on different clients. Furthermore, transformers showed faster convergence that needed fewer communication rounds to reach target accuracy compared to CNN-based models. Additionally, combining transformer architectures with established FL optimization methods, such as FedProx and FedAvg Share, further improve performance. The authors conclude that transformers inherently handle providing an effective and practical alternative to optimization-heavy FL methods, making them suitable for real-world federated scenarios even though having more intense computational demands

In terms of heterogeneity context, data heterogeneity is a special interest for this project. Specifically, modality heterogeneity has been seen as one of the big problems in convergence and accuracy. *'Multimodal Federated Learning: A Survey'* by Che et. al. dug deep into this topic [8]. The authors conducted a literature review on multimodal federated learning, following a well-defined article selection process with specific criteria. Their proposed approach distinguishes between Multimodal Federated Learning (MFL) and traditional Federated Learning, introducing the concepts of modality

combination and modality heterogeneity. To facilitate comparisons between the reviewed studies, they classified MFL research into four categories: horizontal, vertical, transfer, and hybrid MFL, expanding on the initial classification of MFL into congruent and incongruent types. Additionally, they identified common tasks in Multimodal Federated Learning, compiled a benchmark of datasets for MFL, and discussed potential research directions and challenges in the field. Some of these challenges include modality heterogeneity, missing modalities, data complexity, large-scale pre-trained models, privacy concerns, and weakly supervised learning.

The study *UniFed: A Universal Federation of a Mixture of Highly Heterogeneous Medical Image Classifications Tasks* by Hassani et.all utilities federated learning models to expand into multi-task learning problems [9]. In this paper the authors aim to deal with the challenges faced in previous studies highlighting the importance of task heterogeneity as well as other challenges such as real-world application where having tasks specific models would be impractical to implement, the communication costs of exchanging weight updates from client to servers and data heterogeneity where clients would naturally have non-identical independent (Non-IID) data, leading to imbalanced characteristics and unequal training contributions. To achieve this the authors propose a FL framework *UniFed* that learns from a mixture of highly heterogeneous tasks by introducing a loss-guided dynamic and sequential model exchange between the server and clients. The baseline model uses three different models, CNN, VGG11 and ResNet18 with an SGD optimizer and learning rate of 0.001. These models are tested with both a *Strongly Non-IID* and *Moderately Non-IID* dataset which is then analysed against three different types of federated learning *FedAvg*, *FedProx*, *FedSeq* and the proposed *UniFed*. UniFed achieves substantial performance gains across all settings—for example, it improves accuracy over the best baseline by +30.93% (CNN), +24.06% (VGG11), and +23.96% (ResNet18) under moderately Non-IID conditions. In strongly Non-IID settings, it outperforms the top baseline by +30.93% (CNN), +24.06% (VGG11), and +23.96% (ResNet18) in accuracy, while also significantly boosting F1 scores and sensitivity, demonstrating robust generalization across model architectures. It also demonstrates the fastest convergence time at 98.43 minutes along with the lowest communication cost among all federated learning methods.

The paper *"Local Learning Matters: Rethinking Data Heterogeneity in Federated Learning"* by Mendieta et al.(2022),propose FedAlign, which is a federated learning (FL) method that addresses data heterogeneity by improving local learning generality [10]. FedAlign utlised a distillation-based regularization approach, specifically aligning the Lipschitz constants of the network's final block using slimmed sub-bloc and control how sensitive the network's last layers are to small input changes, The authors use FedAlign as baseline against common FL methods,including FedAvg, FedProx, and MOON, on CIFAR-100, CIFAR-10, and ImageNet-200 datasets. Performance was measured through accuracy, computational overhead, and second-order metrics such as Hessian eigenvalues.. FedAlign outperformed baseline approaches, achieving approximately 3.9% higher accuracy on CIFAR-100 compared to FedAvg, while needing only slightly higher computation (1.02x MFLOPs). Notably, FedAlign achieved a 1.8% accuracy improvement over MOON, with significantly lower memory (70% less) and compute overhead (65% less). The authors found that regularization techniques greatly reduced Hessian eigen values.This work shows that focusing on local learning generality can increasingly prevent nonIID data challenges in FL, that balances performance gains and resource efficiency.

Data and model heterogeneity in federated learning is always a challenge. *"Model and Data Heterogeneity Federated Learning"* by Madni et.al combines knowledge distillation and symmetric loss to improve model robustness and performance in heterogeneous medical imaging scenarios [8]. The

paper talks about the complexity when hospitals or institutions have both differing data distributions and custom model architectures. Existing methods like FedMD and FedDF are noted for their reliance on shared models and mutual consensus, however the authors argued that they are not scalable in practical FL environments due to privacy and computational overheads. In stead of requiring uniform data structure, the framework in the article allows each client to maintain its own unique model and coordinating training by exchanging knowledge through public data. What's more, the use of symmetric loss mitigates the impact of label noise and non-IID distributions. Experimental validation was conducted on real-world hematological cytomorphology datasets (INT-20 and Matek-19), showing significant improvements in classification accuracy under both homogeneous and heterogeneous setups. The approach offers an effective way to handle differences in client data and models in federated learning, without the need of a central system or assuming all data are the same.

3.3 Multi-modal Federated Learning

The paper *'FEDMM: Federated Multi-Modal Learning with Modality Heterogeneity in Computational Pathology'* by Peng et.al proposed a different framework for federated learning where instead of a focus on single-modal feature extraction the authors propose a model *FedMM* to train models on different data modalities to obtain the benefit of federated learning while preserving data privacy [11]. The FedMM framework utilities a dynamic loss function for training each client model. This is because in the training phase, a global 'pseudo-label' prototype is used which allows the updating of federated feature extractors in the absence of labels caused by privacy constraints. This global prototype acts as a proxy for a given class and is determined by averaging the embeddings within the same class and modality across all clients, noting that each modality has the same number of prototypes as number of classes. The aim is then to minimise the loss functions by using a mix of L2 and BCE loss. To test the effectiveness of their framework, FedMM was evaluated on two datasets, *TCGA-NSCLC* which contains data from patiends with non-small cell lung cancer with two subtypes. The other *TCGA-RCC* included data from patients with renal cell carcinoma (most prevalent type of kidney cancer), with three subtypes. These datasets had two different modalities and two different classes for each set. To test these, the whole slide images were cropped and then had their features extracted using a ResNet-34 model, pre-trained on ImageNet, which then attention pooling is used to aggregate patch-level representations. For Copy Number Variation (CNV) data, preprocessing is done using GISTIC2, followed by feature extraction with a Self-Normalizing Network (SNN) that consists of two hidden layers. The final fully connected layer generates the CNV feature representation. Three clients were used for each test with 100 global rounds used for training and the experiment repeated 20 times. From this study, they found that FedMM surpasses local training and Multi-FedAvg baselines by AUC values of a 2.79% increase and 0.065% increase respectively with the TCGA-NSCLC dataset. The TCGA-RCC dataset backed this with an increase of AUC values of 2.97% and 7.96% respectively showcasing the practical effectiveness of the FedMM framework.

To deal with high modality and task diversity, *'High-Modality Multimodal Transformer: Quantifying Modality and Interaction Heterogeneity for High-Modality Representation Learning'* was proposed [12]. The authors focus on contexts with a high number of diverse tasks and modalities, proposing a single model capable of adapting to different modality configurations and tasks. To achieve this, they first introduce an approach for quantifying heterogeneity using transfer learning between modalities. This approach enables the reuse of certain model components to process modalities that share similarities and allows for the integration of useful information from different modalities to leverage their interactions. In practice, this involves parameter sharing across similar modalities

that exhibit common features or interactions. Their experiment includes a single-model setup with 10 modalities and 15 prediction tasks across five different research areas. HighMMT demonstrates strong overall performance while using only one-tenth of the parameters required for task-specific models.

The article "FedMultimodal: A Benchmark for Multimodal Federated Learning" [13], represents a new benchmark that aims to solve several challenges in multimodal federated learning, especially when each client has different types of data, i.e. modalities. In previous works, the focus was on using only one modality and assuming all clients have similar data, which is not aplicable for real-world challenges. Thus, in this work, the authors design FedMultimodal, which includes a group of datasets and tasks to test how well different federated learning models perform when the data comes from a wide range of modalities. The benchmark includes ten datasets and five real-world applications like human activity recognition and disaster event classification. These datasets use eight different types of data (like audio, text, sensors, etc.) and are distributed in a way that simulates real cases where not all clients have access to the same modalities. They tested different federated learning algorithms, such as FedAvg and FedOpt, and also tried different ways to combine the information from the modalities. Among these, the attention-based fusion method worked the best, especially when the clients had different data or some modalities were missing. One interesting part of the benchmark is that it also considers missing modalities—meaning that in some cases, a client doesn't have access to all the inputs. The results show that models using attention fusion were more robust and could still perform well even when up to 30% of the modalities were missing. After that, performance started to drop, but not so drastically until over half of the modalities were gone. In addition, the experiments proved that using multiple modalities gives better results than using just one, especially for more complex tasks.

3.4 Federated Learning in Medical Imaging

'An efficient federated learning method based on enhanced classification-GAN for medical image classification' by Liu et.al explores the issues faced by the medical industry through the lack of the labeled data and privacy concerns causing image classification models to struggle to accurately classify medical images [14]. To address this, the authors propose a federated learning generative adversarial network implemented with blockchain to improve security of the model while also addressing the issues of reduced number of labels present in medical imaging. The authors evaluated this model using the 'Covid-19 Radiography Database' and 'ChestCOVID', which contain lung images from COVID-19-positive cases, normal lungs, viral pneumonia, and other conditions. These datasets, comprising 900 lung images, were divided into two new independent subsets, derived from prior COVID-19 studies [15] [16] [17]. The proposed FEDBG was then compared to models proposed in previous studies through training time, precision, F1_Score and synthetic image quality which was then evaluated through an ablation study. The results found that on both training sets, the precision, recall and F1_Score to be above 95 for all categories placing it higher then baseline models. It was also found that the training speed appeared to approximately 27–38% faster and overall accuracy increase 0.9–2%, therefore indicating that the proposed FEDBG is an optimal choice in medical imaging classification.

Adnan et al., in their groundbreaking study of *"Federated learning and Differential Privacy For Medical Image Analysis"*investigate the use of Federated Learning (FL) to classify the different type of lung cancers from histopathology images [18]. They utilised Whole Slide Images(WSIs) of lung cancers, specifically targeting two cancer subtypes: LUAD and LUSC.Initially, they extracted

image patches from the WSI and used DenseNet to derive feature vectors. Multiple Instance Learning (MIL) was then applied to classify these slides. In their initial experiment, the authors simulated clients to evaluate FL performance under different data conditions, including IID and non-IID scenarios.They discovered that FL significantly outperformed individual hospitals in training alone and nearly matched centralized training accuracy when the number of clients was limited. In a followup experiment involving real hospitals,the model implemented Differential Privacy (DP)) which resulted in a strong privacy outcome. However, due to this there was also a slight decrease in accuracy on external hospitals, mostly due to domain differences among datasets.The study concluded that FL combined with DP can effectively maintain privacy while also achieving high accuracy. Thus this proposed framework is essential to be studied for medical institutions that intend to collaborate without directly sharing patient data (Adnan et al,2022).

The paper*"Federated Learning for Enhanced Medical Image Analysis"* by Sanaa Lakrouni et.al explores how Federated Learning(FL) enables collaborative model training across multiple medical institutions while preserving data privacy [19]. The authors emphasise that medical datasets are always different due to variations in imaging equipment, scanning protocols, and etc. This non-IID (non-independent and identically distributed) data distribution poses a significant challenge in Federated Learning, as it can lead to performance degradation. In order to duel with that, the study suggested methods such as Vision Transformers (ViTs) and Self-Supervised Learning (SSL) to enhance FL's robustness in handling this non-IID medical image data. The results from the authors' research indicate that the techniques above can significantly improve classification performance compared to traditional Federated Learning models. Another significant contribution in the field is the integration of Generative Adversarial Networks (GANs) into FL frameworks. Some studies suggested using FL-GAN architectures to address the scarcity of labelled medical images while maintaining data security. These models generate synthetic medical images, which can supply for real datasets and improve model generalisation. Blockchain technology is also incorporated in some studies to further secure the environments for Federated Learning, ensuring data integrity and make it hard to be attacked. Privacy-preserving techniques have been another critical aspect in recent FL research. Differential Privacy (DP) and Secure Multi-Party Computation (SMPC) have been explored as methods to enhance the security of FL models. Studies suggest that adding noise to model updates or encrypting data during aggregation can help prevent potential privacy leaks while maintaining model performance. However, a key challenge remains in balancing privacy with model accuracy, as excessive noise can degrade learning efficiency. Also, some studies talk about a novel FL framework designed to handle multi-modal medical imaging data. Unlike conventional FL models that focus on single-modal data, this approach utilises dynamic loss functions and global pseudo-label prototypes to train models across diverse data modalities. The authors can see from the evaluation on cancer datasets that this method significantly outperforms standard FL techniques in terms of classification accuracy and robustness to data variations. What's more, research on Active Learning (AL) combined with FL has shown the ability of reducing the reliance on fully labelled datasets. By leveraging uncertainty-based sampling techniques, AL-FL models selectively request labels for the most informative data points, minimising annotation costs while maintaining the accuracy. In conclusion, the paper covers the recent advancements in federated learning for medical image analysis which mainly focused on addressing challenges related to data heterogeneity, privacy, and limited labelled data. Techniques such as Vision Transformers, self-supervised learning, generative adversarial networks, blockchain security, and active learning have significantly contributed to enhancing the effectiveness of FL in healthcare applications.

Lutnick et al. (2022) in his seminal paper *"A tool for federated training of segmentation models on whole-slide images"* developed a tool called Histo-Cloud for federated training of segmentation models more specifically on whole-slide images (WSIs) [20]. They aimed to handle privacy concerns by training models across several client sites without transferring data. They tested their method using two case studies: interstitial fibrosis and tubular atrophy (IFTA) and glomerulus segmentation in kidney biopsies. Performance was measured with ROC-AUC for IFTA and Matthews Correlation Coefficient (MCC) for glomerulus segmentation. Federated training achieved similar results to traditional pooled datasets. For IFTA segmentation, federated models had an internal hold-out ROC-AUC of 0.95 and an external ROC-AUC of 0.90. These scores outperformed or matched centrally trained models with 0.95 internal and 0.88 external). Glomerulus segmentation also showed similar performance where using federated MCC it obtained 0.91 internally and 0.80 externally, compared to 0.91 and 0.83 for central training. Single-site models performed significantly worse. Training times were almost 2 times longer for federated approaches. This was theorised by the authors as due to communication overhead. However, federated training was said to enhance model generalizability across different staining methods and clients. The authors suggest federated training is achievable and match with centralized training accuracy results , butt also noted limitations. The study concludes that federated learning maintains data privacy effectively, but the longer training duration and lack of hard evident privacy methods raise some concerns.

With the consideration of both disease diversity and modality heterogeneity in federated medical imaging, *"Feasibility of Federated Learning from Client Databases with Different Brain Diseases and MRI Modalities"* by Wagner et al. explores the possibility of training a unified segmentation model across decentralized brain MRI datasets using Federated Learning (FL) [21]. Unlike traditional models, which are limited to specific pathologies and fixed modality sets, the study proposes a practical FL-based framework where each client have its unique combinations of MRI modalities and lesion types. To handle the varying MRI types across sites, the authors design a U-Net architecture to accept all possible modalities, applying zero-filling for missing ones and randomly dropping modalities during training in order to improve the model's ability to generalize. Additionally, they examine the impact of different normalization strategies, such as client-specific BatchNorm, InstanceNorm, GroupNorm, and a normalization-free (NF) approach—on both original training data and new, unseen client data. Using 7 brain MRI datasets with 5 disease types and 6 modality variants for testing, they confirm the that it's possible to train one model that works well across diverse input combinations, even for cases it hasn't seen before, including zero-shot generalization to previously unseen datasets. This work is an important milestone when applying FL to highly diverse clinical settings, being a practical way to train shared models without sharing sensitive medical data.

4 Project Definition

4.1 Project Questions

- Federated learning can achieve better or equal performance compared to the centralised alternative.
- Different but overlapping sets of classes between clients can improve overall performance in Federated Learning with uni modal clients.
- Multi-modality, with different modalities and classes (partially overlapping) for each client, increases performance in Federated Learning contexts (highly heterogeneous conditions) due to shared knowledge between modalities and classes.

4.2 Aims and Objectives

4.2.1 Aims

- Develop and implement the core concept of the project - Establish a strong theoretical and practical foundation through research and data collection in phase 1.
- Optimise and test the proposed solution - refine and iterate on the solution, conducting small-scale testing to ensure feasibility and practicality in phase 2.
- Deliver and evaluate the final project outcomes - implement the solution, assess effectiveness, and propose improvements for long-term sustainability.

4.2.2 Objectives

- Design and implement a federated learning framework that enables decentralised model training without compromising data privacy.
- Optimise the performance of the federated learning system through iterative testing and feedback loops.
- Evaluate the accuracy, efficiency, and security of the federated model compared to traditional centralised approaches.
- Deploy the federated learning solution in a real or simulated environment and analyse its performance based on key metrics.

4.3 Scope

- Implementation Of Federated Learning: Design a federated learning framework to enable model training across multiple devices or clients without sharing the raw data.
- Optimisation: Testing and optimising the model to improve it's accuracy, efficiency and scalability.Also get a comparison with the traditional centralised machine model to access advantages.
- Real-World Applications: Identifying and applying federated learning in a relevant domain, specifically in healthcare area in this research.
- Security and Privacy: Address key challenges related to data privacy, security risks, and compliance with regulations such as GDPR in order to protect it against attacks and data breaches.

5 Methodology

To achieve the objectives set out by the team the development of the model was split into three core phases as seen in Figure 1. *Phase 1* aims to establish a baseline framework for the team to build on where testing of different machine learning models could occur while the team familiarised themselves with federated learning. *Phase 2* aimed to gradually increase the complexity of the model to allow for clients to have a different number of classes with the potential for overlap between them. In this phase the addition of a custom FedAvg model was also developed for phase 3 implementation. *Phase 3* significantly increased the challenge with the addition of multiple modalities where the team aimed to optimise the framework by testing different federated learning methods.

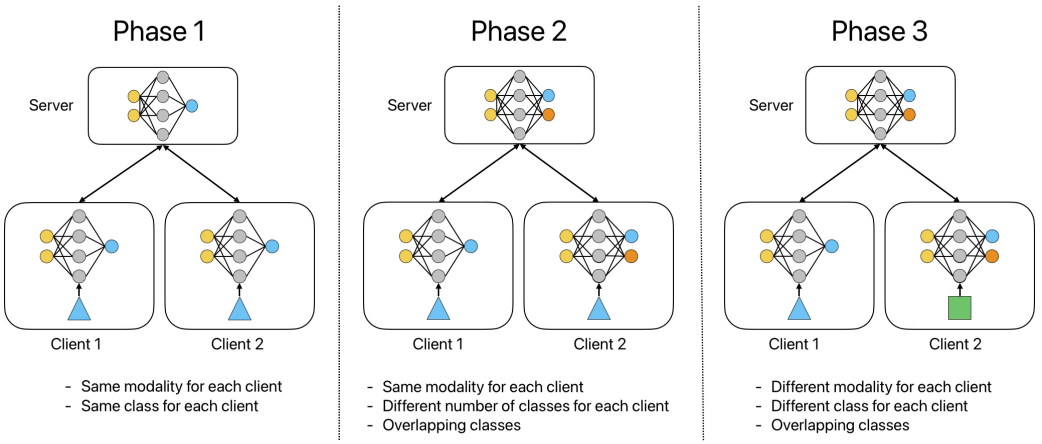


Figure 1: Three Development Phases

5.1 Phase 1

5.1.1 Dataset and EDA

The dataset used for phase 1 was the colon section of the *Lung and Colon Cancer Histopathological Images (LC25000)*. This dataset includes 25,000 color images generated through augmentation of 1,250 HIPAA-compliant originals. The histopathological images were 768px x 768px. The selected portion of the dataset is used in the phase 1 to test different model performance on federated learning. A detailed exploratory data analysis (EDA) was performed, starting with examining the directory structure (Table 1) and image counts to ensure correct labeling and identify class imbalances. The quality, correct labeling, and visual clarity of the sample images were reviewed, followed by checking for corrupt files and inconsistencies in the size of the images. The team also performed color distribution analysis using color histograms to detect variations due to imaging conditions and help to decide if images need further preprocessing. The data preprocessing steps ensured the quality, uniformity, and suitability of the dataset to build robust and reliable machine learning models.

Path	**Subclass**	**Distribution**
/colon_image_sets/colon_n	Colon benign tissue	50%
/colon_image_sets/colon_aca	Colon adenocarcinoma	50%

Table 1: Colon tissue subclasses [22]

After downloading dataset via Kaggle, few data preprocessing jobs need to be done. Since the image size are not uniformed which ranges between 640 × 480 px and 1024 × 768 px and also too large for efficient training in early phases, we Converts the images from PIL format to normalized tensors with pixel values scaled to [0.0, 1.0], and all images are now resized to a uniform size of 28×28 pixels to ensure consistent input dimensions for the CNN. Then split the data into training (80%) and testing (20%) subsets using random split, ensuring reproducibility with a fixed seed. Then we have to do test for both IID and non-IID distribution, and these two figures are the partition situation.

No advanced augmentation techniques like ColorJitter, Normalize, or RandomCrop were used in this phase, which keeps the setup minimal and ideal for initial benchmarking. We made the transfor-

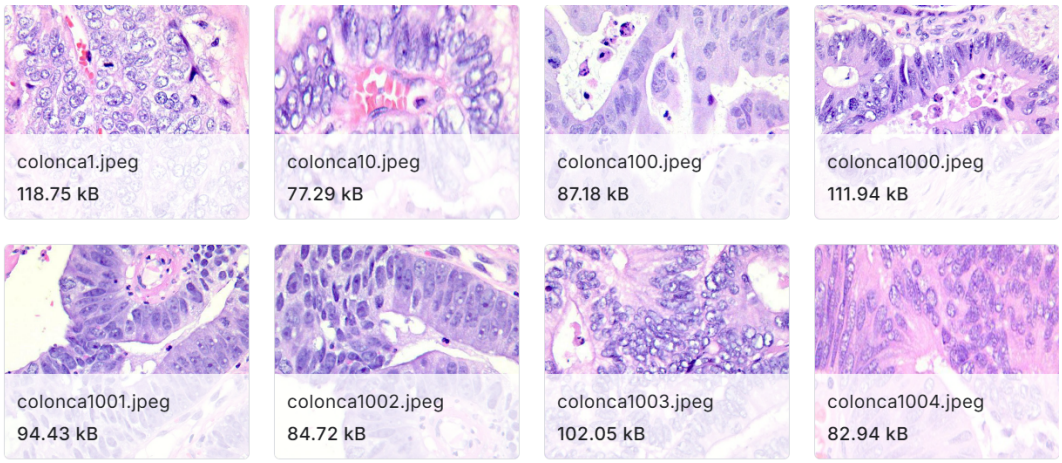


Figure 2: colon$_a casubclassexample$

mation pipeline simple and effective to meet the goal of phase 1: provide a clean, controlled learning environment using a single class and modality.

5.1.2 Framework

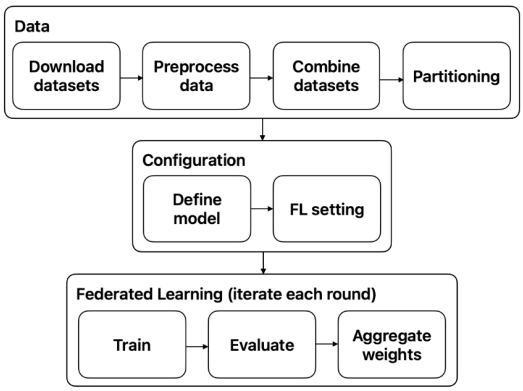


Figure 3: Framework

A simple CNN was defined with two convolutional layers followed by fully connected layers. Then tailored the network for binary classification in a controlled environment. We simulate separate institutions or nodes and make each client hold a private data partition (IID or non-IID) and train a local model, and the local updates were computed independently while doing this local training. After local training, the server coordinated global training and handled model aggregation using FedAvg. During iteration, the evaluation was done locally after each communication round, and metrics such as accuracy and loss were logged for each client and the global model. After that, the Flower server performed weighted averaging of model parameters using FedAvg and the aggregated model was then redistributed to the clients for the next round.

Another model being tried in phase 1 is VGG11, since Pre-trained VGG11 model is known for its depth and powerful feature extraction capabilities. Feature extraction layers (vgg11.features) were frozen to retain learned low-level visual features. The final fully connected layer was replaced to output a single value for binary classification using a sigmoid activation and this architecture signif-

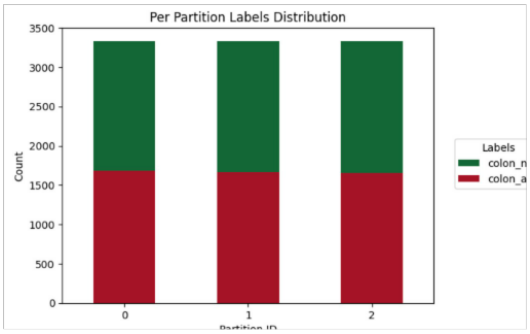


Figure 4: IID partition

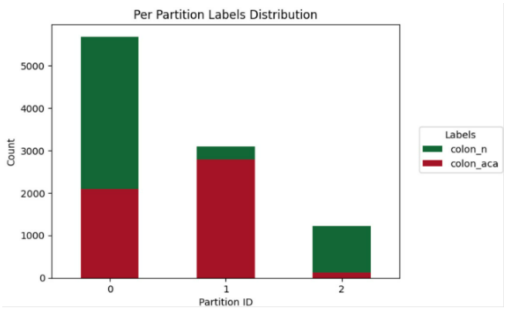


Figure 5: non-IID partition

icantly increases representational capacity which suits well with high-resolution histopathological images. The FL environment pretty much remained similar to the simple CNN setup. Data was partitioned across clients. And for each client: we loaded local histopathology images (resized). Fine-tuned only the last layer of VGG11 for binary classification. Then using binary cross-entropy loss and Adam or SGD optimizers. Then perform local validation after each round to track the performance and metrics like accuracy and loss were calculated per client and logged. After local training, the updated model weights were sent back to the Flower server. The server then averaged the client models using the FedAvg algorithm (weighting based on client sample size). At last, the updated global model was redistributed for the next round.

5.2 Phase 2

5.2.1 Dataset and EDA

The dataset used for phase 2 of the project was the *Cervical_Cancer-cancer* dataset sourced from Amadou Alwaly Ndiaye listed on Hugging Face which consisted of 25,000 coloured images with the size of the images being 224px by 224px [23]. The images are then classified into five different types of cell diseases noted as String values seen in Table 2. Each category is distributed equally throughout the original dataset with examples of the 5 image categories can be seen in Figure 6.

From data exploration, it was identified that the images seemed a bit faded, so the team had implemented the following PYTORCH libraries *ToTensor* which transforms 'PIL Images' (shape H x W x C and pixel values in the range [0, 255]) into PyTorch tensors (shape C x H x W and values in the range [0.0, 1.0]) due to PyTorch models requiring a specific input size. *ColorJitter* is used to increase the contrast and brightness of the image which for the *Phase 2* model was increased by

Path	Subclass	Description	Distribution
/cervix_dyk	Dyskeratotic	Abnormal cell growth	20%
/cervix_koc	Koilocytotic	Cells showing changes from viral infections (e.g., HPV)	20%
/cervix_mep	Metaplastic	Cells changed from one type to another (precancerous)	20%
/cervix_pab	Parabasal	Immature squamous cells	20%
/cervix_sfi	Superficial-Intermediate	More mature squamous cells	20%

Table 2: Cervical cell subclasses and their descriptions [23]

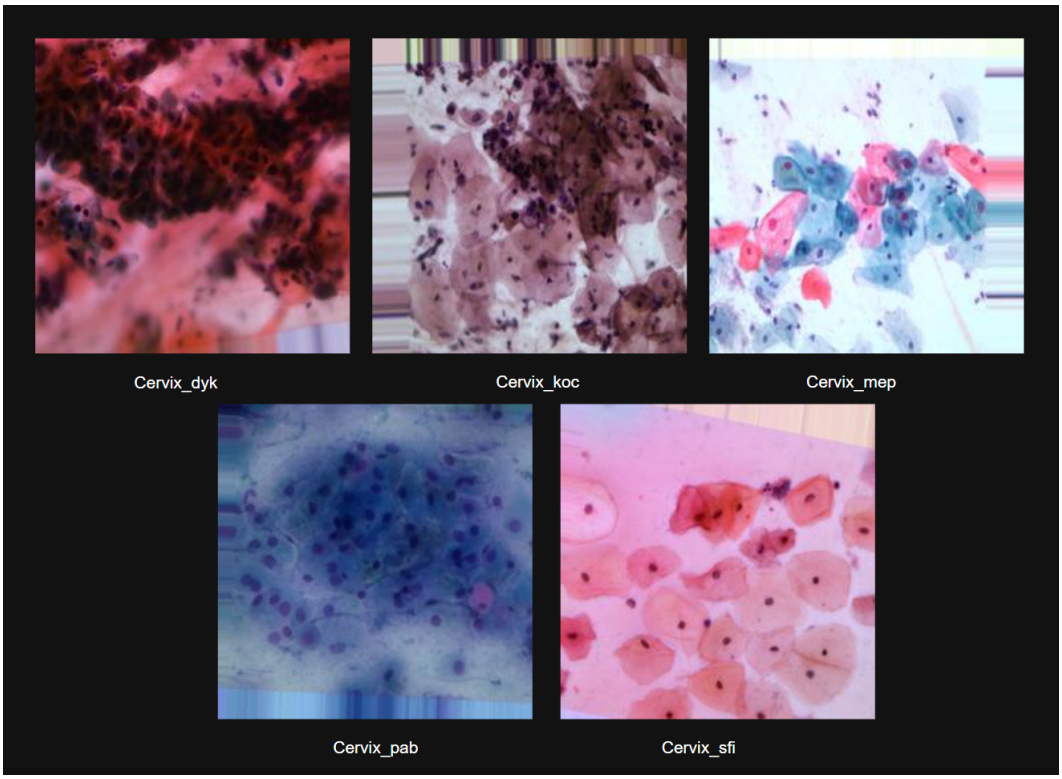


Figure 6: Cervical Cell Subclass Examples

0.1 for both. Lastly *Normalize* was applied with the mean [0.485, 0.456, 0.406] and standard deviation of [0.229, 0.224, 0.225] to speed up the convergence of the model by standardising the input distribution.

5.2.2 Framework

Building on the framework discussed in *Phase 1*, *Phase 2* aimed on testing the framework by varying the amount of classes per each client. For example, 'Client 1' could have three classes while 'Client 2' could have four classes and 'Client 3' could have two classes with every client having the possibility to have overlapping classes. This framework has two key components that separate it

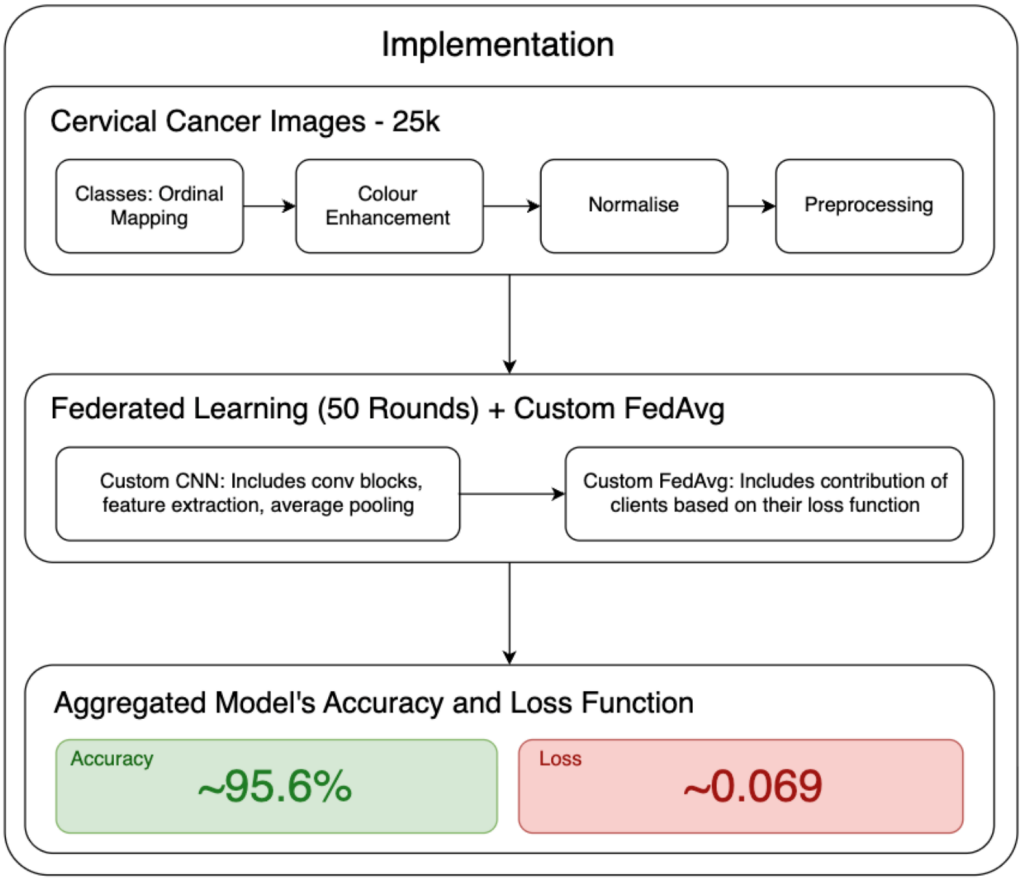


Figure 7: Phase 2 Framework

to that seen in *Phase 1*, a *Custom Convolutional Neural Network* and a *Custom Federated Average* model as illustrated in Figure 7.

A *Custom CNN* was implemented into this phase to try and minimise the information loss when images are inputted into the model that could be caused by using pre-built models such as VGG11 which require a minimum input size while also allowing the team to perform greater customised hyper-parameter tuning where required. The *Custom CNN* starts by first initialising a convolutional block, which process images with three channels (RGB) applying *batch normalisation* and using a ReLu activation function. After this layer, three feature extraction layers are used where they reduce the spatial dimensions of the images changing the dimensions of the outputs to 14 x 14 pixels after the final layer. *Average Pooling* and *Dropout* are then used to reduce the size of each feature map while also reducing bias in the training where a final fully connected layer outputs the class score. Custom initialised weights are then used to improve the stability of the model. The parameters for the *Custom CNN* are listed below:

- **Initial Convolutional Layer**: `nn.Conv2d(3, 32, kernel_size=3, stride=2, padding=1, bias=False)`
- **Batch Normalisation**: `nn.BatchNorm2d(32)`
- **ReLu**: `nn.ReLU(inplace=False)`

- **Layer1**: `_make_layer(32, 64, 2, stride=2)`
- **Layer2**: `_make_layer(64, 128, 2, stride=2)`
- **Layer3**: `_make_layer(128, 256, 2, stride=2)`
- **Avgpool**: `nn.AdaptiveAvgPool2d((1,1))`
- **Dropout**: `nn.Dropout(0.3)`
- **Fully Connected**: `nn.Linear(256, NUM_CLASSES)`

The second key change from *Phase 1* was the implementation of the *Custom FedAvg* learning method to calculate the weight/contribution of each client to the *Custom CNN* model. If the loss of the client was greater it would reduce the contribution of that client to the model and vice versa, as seen in Figure 8. It does this by using the *Flower* server and *FedAvg* libraries to first initialise a server instance. From here it initialises the models to obtain the parameter structure and extracts weights and metrics from the inputs. These inputs are then converted to *NumPy* parameters to validate the values match the expected strucutre. From here a weighted aggregation occurs which allows the server to store the clients parameters with the number of training examples they used. This then allows the server to compute a *sample-weighted average* and obtain the loss and accuracy metrics weighted by the number of examples. Weight aggregation then occurs and performs a weighed sum of each of the model's layer parameters. The definitions of the parameters for the *Custom FedAvg* are seen below:

- **Model:** `Net()` – Custom PyTorch model for classification.
- **Strategy:** `SimpleFedAvg` – Inherits from `FedAvg`, with custom aggregation.
- **Initial Parameters:** `get_weights(net)` – Extracts model weights.
- **Aggregation Function:** `_aggregate_weights` – Computes sample-weighted average.
- **Validation Function:** `_validate_params` – Ensures parameter structure matches the model.
- **Metrics Aggregation:** `weighted_average` – Computes weighted average of loss and accuracy.
- **Server Configuration:**
  - `num_rounds` – Number of federated training rounds.
  - `fraction_fit` – Fraction of clients used for training each round.
  - `fraction_evaluate = 1.0` – All clients used for evaluation.
  - `min_available_clients = 2` – Minimum clients required to proceed.
- **ServerApp:** `ServerApp(server_fn=server_fn)` – Launches the federated server.

5.3 Phase 3

5.3.1 Dataset and EDA

Phase 3 dataset used two complementary dataset sources that were recombined to give us a dataset that has the following characteristics: multiple modalities, multiple classes and overlapping classes (i.e. labels). The first source (Dataset A) is Lung and Colon Cancer Histopathological Images also known as the LC25000 previously used in Phase 1 of our project. The extended details of this dataset can be observed in Phase 1 section of the Dataset and EDA.

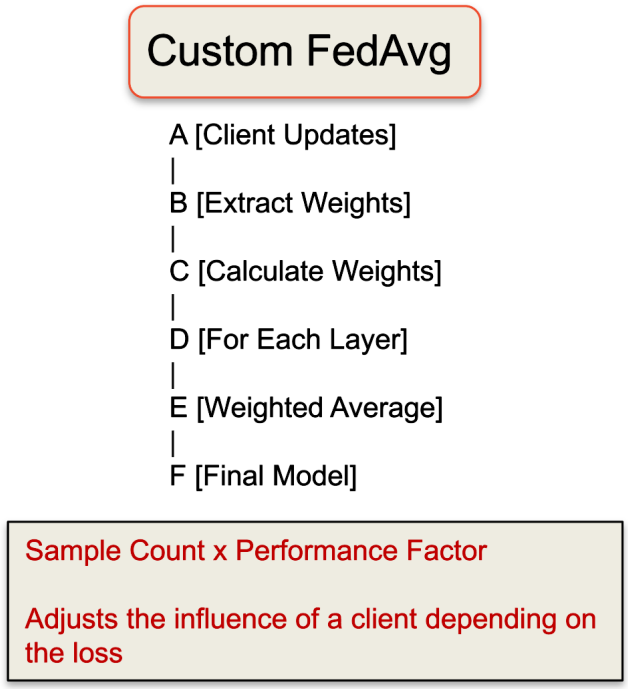


Figure 8: Custom FedAvg

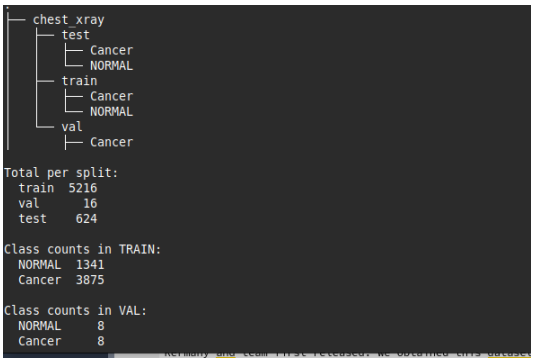


Figure 9: File Tree of Dataset B in Phase 3

The second source of the dataset (Dataset B) is a well known dataset of paediatric chest-X-ray collection that Kermany and team first released. We obtained this dataset from the Kaggle user quynhle_CL, who reposted the original dataset collected by Kermany. It holds 5863 JPEG images already sorted into three folders: train (5216 files), validation (16), and test (624). Each of those folders has two sub-folders named Normal and Lung Cancer (we relabel that class to Cancer so it fits our study). The training part is clearly imbalanced with only 1341 images are normal while 3875 show cancer, giving a 1:2.9 ratio. Image sizes are not uniform and ranges from between 512×512 px and 1024×1800 px. All scans are anterior–posterior views of children aged one to five years from Guangzhou Women & Children's Medical Center in China. Metadata and DICOM headers were not available, so further more detailed examination was limited.

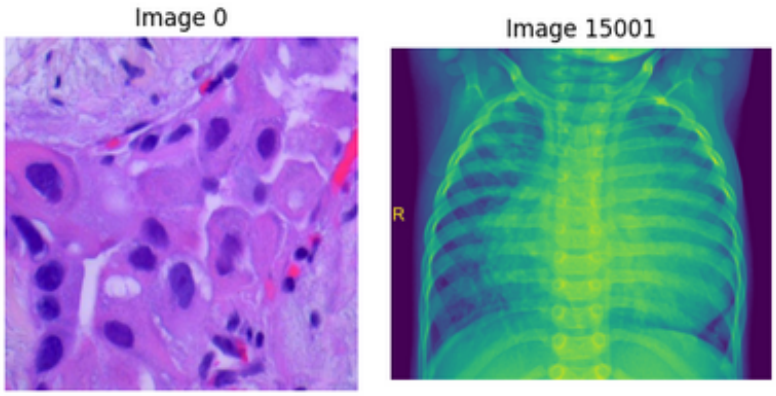


Figure 10: Before Preprocessing

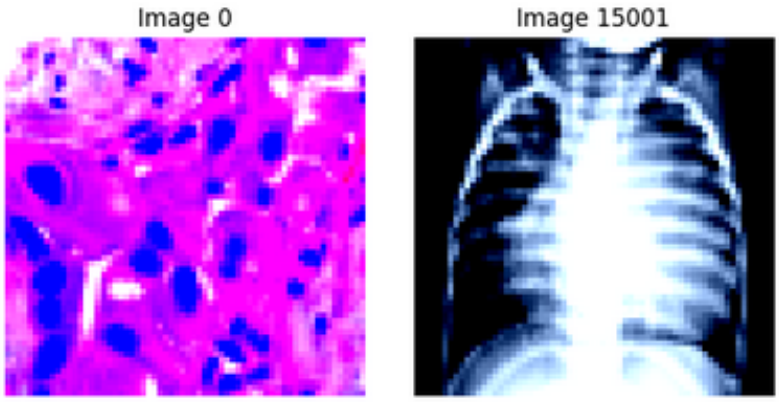


Figure 11: After Preprocessing

For the pre-processing of the data, we define a single preprocessing pipeline with TorchVision. Every image regardless of their modality were resized to 64×64 pixels, converted to a tensor, and normalised with the standard ImageNet mean and standard deviation. A custom MultiModalDataset class is then instantiated. It receives the dataset root, the list of modality folders, and the common transform that was configured. Figure 10 shows the image sample from each partition where we reverse the normalisation, and display them side by side. The need for pre-processing for this dataset is for federated model batches to have uniform shape and numeric range while still preserving the genuine modality differences that the network is meant to learn. Figure 11 shows the same images, but after pre-processing.

Both of these datasets were pre-processed and recombined to give us the dataset for Phase 3. Figure 12 summarises the label composition of the Phase dataset after we harmonised the two source datasets and assigned them to two clients. The five original LC25000 histopathology classes were filtered into three clinically meaningful super-classes—label 0: benign or normal cell, label 1: malignant lung adenocarcinoma and label 2: malignant lung squamous. This was done to create a label space that the chest-X-ray collection could partially share 15000 histology images were retained for Phase 3 and were placed in Partition 0. This was done to ensure an even distribution across the three labels. The paediatric chest-X-ray images (n = 5856) were mapped to labels 0 and 1, because the dataset contains only two classes, and transferred in to to Partition 1. Consequently, the left bar in

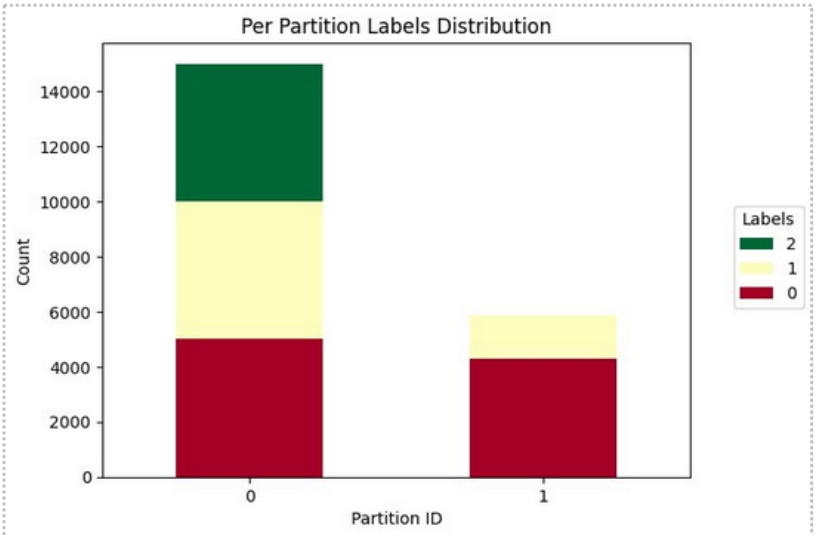


Figure 12: Per Partition Label Distribution

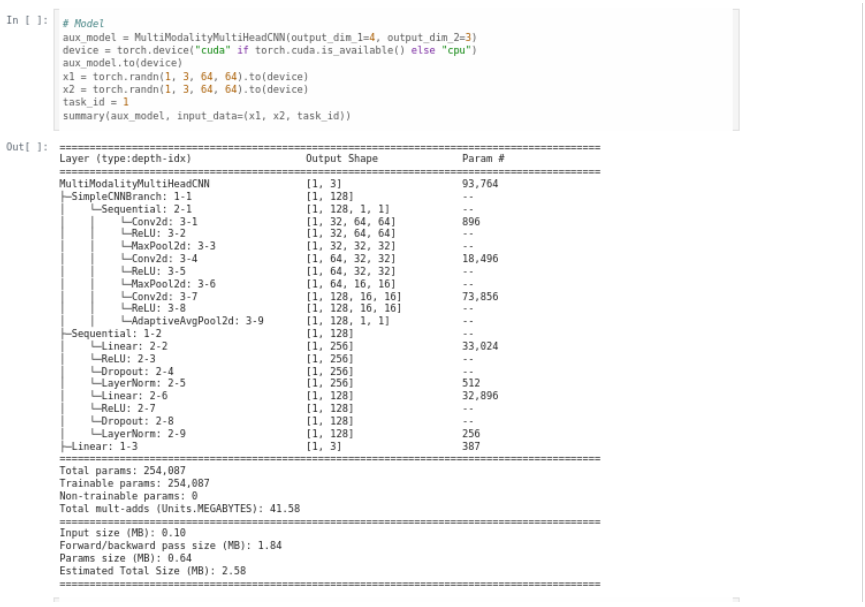
the chart shows three equal stacks (red, yellow, green), whereas the right bar contains only the red and yellow stacks at lower counts. This setup gives each client a different amount of data and not all the same labels, so their data are intentionally uneven for our federated learning tests.

5.3.2 Framework

We create MultiModalityMultiHeadCNN with two output heads: one head can return three logits, the other return two logits. Two dummy tensors, each shaped (1×3×64×64), stand in for a single histology image and a single chest Xray after resizing. There are 3 blocks that make up the architecture of the MultiModalityMultiHeadCNN which is shown in Figure 13 and itemized as follows:

- SimpleCNNBranch which concist of three Conv-ReLU-MaxPool layers together with an AdaptiveAvgPool that squeezes each 64 × 64 image down to a 128-element feature vector.
- Fully-connected trunk which consist of two Linear layers of ReLU, Dropout, and LayerNorm. It mixes the features and keeps only 128 units.
- Linear head which is a single Linear layer that maps 128 numbers to the chosen logit count.

Our phase 3 framework evaluates three federated learning aggregation strategies, which are FedAvg, FedOpt, and FedProx. Each method runs for a total of 50 rounds. At the start of each experiment, ”resetheads()” is called to ensure each strategy avoids averaging the last block because their weights belong to each client and they are used as somewhat perzonalized models. The model parameters are extracted and set as the initial global parameters for federated training. The first strategy tested is FedAvg, a basic federated averaging approach. All clients participate in each round (fractionfit and fractionevaluate set to 1.0). Client results are averaged based on sample sizes to calculate a global accuracy metric. The experiment simulates two client nodes. FedOpt is the second strategy examined. It builds on FedAvg by adding a server-side optimiser. Parameters like eta, etal, beta1, and beta2 control the optimiser learning rate and momentum. Apart from this optimiser, FedOpt uses the same settings as FedAvg to ensure comparability. The third implemented strategy is FedProx. It modifies the local training loss by including a proximal term. This proximal term prevents



```
In [ ]:
# Model
aux_model = MultiModalityMultiHeadCNN(output_dim_1=4, output_dim_2=3)
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
aux_model.to(device)
x1 = torch.randn(1, 3, 64, 64).to(device)
x2 = torch.randn(1, 3, 64, 64).to(device)
task_id = 1
summary(aux_model, input_data=(x1, x2, task_id))

Out[ ]:
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
MultiModalityMultiHeadCNN                [1, 3]                    93,764
├─SimpleCNNBranch: 1-1                   [1, 128]                  --
│    └─Sequential: 2-1                   [1, 128, 1, 1]            --
│    │    └─Conv2d: 3-1                  [1, 32, 64, 64]           896
│    │    └─ReLU: 3-2                    [1, 32, 64, 64]           --
│    │    └─MaxPool2d: 3-3               [1, 32, 32, 32]           --
│    │    └─Conv2d: 3-4                  [1, 64, 32, 32]           18,496
│    │    └─ReLU: 3-5                    [1, 64, 32, 32]           --
│    │    └─MaxPool2d: 3-6               [1, 64, 16, 16]           --
│    │    └─Conv2d: 3-7                  [1, 128, 16, 16]          73,856
│    │    └─ReLU: 3-8                    [1, 128, 16, 16]          --
│    │    └─AdaptiveAvgPool2d: 3-9       [1, 128, 1, 1]            --
├─Sequential: 1-2                        [1, 128]                  --
│    └─Linear: 2-2                       [1, 256]                  33,024
│    └─ReLU: 2-3                         [1, 256]                  --
│    └─Dropout: 2-4                      [1, 256]                  --
│    └─LayerNorm: 2-5                    [1, 256]                  512
│    └─Linear: 2-6                       [1, 128]                  32,896
│    └─ReLU: 2-7                         [1, 128]                  --
│    └─Dropout: 2-8                      [1, 128]                  --
│    └─LayerNorm: 2-9                    [1, 128]                  256
├─Linear: 1-3                            [1, 3]                    387
==========================================================================================
Total params: 254,087
Trainable params: 254,087
Non-trainable params: 0
Total mult-adds (Units.MEGABYTES): 41.58
==========================================================================================
Input size (MB): 0.10
Forward/backward pass size (MB): 1.84
Params size (MB): 0.64
Estimated Total Size (MB): 2.58
==========================================================================================
```

Figure 13: Enter Caption

local client models from diverging significantly from the global model. The strength of this term is controlled by the parameter proximal_mu. After running all 3 experiments, metrics from each round are collected and serialised into a JSON file for further analysis. The accuracy of each method is plotted for visual comparison. This helps us to analyse how quickly and effectively each strategy converges under the non-IID conditions of Phase 3.

6 Resources

6.1 Hardware and Software

Hardware

- **Local development machines:** Apple MacBook Air (13-inch, M1, 2021) equipped with:
  - Apple M1 8 core CPU
  - 8 GB RAM
  - 512 GB SSD
  - macOS Monterey

  Each team member uses a personal workstation with similar hardware specification for coding, experimentation, and debugging.
- **Cloud compute environments:**
  - **GitHub Codespaces:** Managed cloud IDE and container environment federated-framework deployment.
  - **Google Compute Engine VMs** Free-tier instances for large dataset experiments such as VGG11 training and multimodal fusion tests.
- **External GPUs:** Speed up the parallel computations needed for training and inference of deep-learning models.

- T4 GPU
- 32 GB RAM

Software

- **Flower:** Federated-learning framework for machine learning orchestration
- **PyTorch:** Core deep-learning library for machine learning models
- **Jupyter Notebooks:** Exploratory data analysis and rapid prototyping
- **Python libraries:** `numpy`, `pandas`, `matplotlib`, plus dataset loaders.
- **Version control:** Git & GitHub for source control, reviews and repository

6.2 Roles and Responsibilities

Table 3: Team Members, Uni Keys, and Roles/Responsibilities

Name	Uni Key	Roles/Responsibilities
Rajiv Mehta	rmeh0608	Project Manager, Data Research, Reporting Team
Ngai Chun Fu	ngfu0299	Data Scientist,Software Engineer – Phase 1
Eshan Arora	earo0293	Data Scientist,Software Engineer – Phase 2
David Cortes Sánchez	vcor0924	Data Scientist,Software Engineer – Phase 3
Mohamad Akmal bin Abu Bakar	mbin0316	Data Research and Engineering, Reporting Team
HanWen Tian (Simon)	htia0294	Data Research and Engineering, Reporting Team

7 Milestones/Schedule

7.1 Milestones

To complete the proposed project, the project was split into five core phases, the first being a project proposal and group formation phase, three project development phases and one final reporting phase. The milestones for this project came from the original scoping of these phases as seen in the Gantt Chart in Figure 14.

The key milestones for each phase is listed below:

Phase 0 – Project Allocation and Proposal Document

- **Group Formation**: Six team members are assigned or self-organise into groups.
- **Project Allocation**: Each group is assigned a specific project topic or selects one from a list.
- **Project Definition and Scope**: Clarify the problem statement, objectives, and boundaries of the project.
- **Group Progress Report (Deliverable)**: A brief update on team formation, initial planning, and early progress.

Figure 14: Gantt Chart - Proposed Project Timeline

- **Project Proposal (Deliverable)**: A formal document outlining the project goals, previous literature, methodology, expected outcomes, and timeline.

Phase 1 – Singular Dataset, Singular Modality, Singular Class

- **Data Collection**: Gather a single dataset with one type of data (e.g., images, text) and one class/category.
- **Data Exploration/Transformation**: Analyse and pre-process the data to make it suitable for modelling.
- **Build Framework**: Develop the initial model architecture or pipeline.
- **Train**: Train the model using the prepared dataset.
- **Test**: Evaluate the model's performance on unseen data.
- **Analyse Results**: Interpret the model's performance and identify areas for improvement.

Phase 2 (MVP) – Multiple Datasets, Singular Modality, Multiple Classes

- **Data Collection**: Collect multiple datasets of the same type (e.g., all images) but with multiple classes.
- **Data Exploration/Transformation**: Merge, clean, and pre-process the datasets.
- **Build Framework**: Extend the model to handle multiple classes.
- **Train**: Train the model on the expanded dataset.
- **Test**: Evaluate performance across all classes.
- **Analyse Results (Deliverable)**: Provide a detailed analysis of the model's performance and insights gained.

Phase 3 – Multiple Datasets, Multiple Modalities, Multiple Classes

- **Data Collection**: Gather datasets from different modalities (e.g., images + text) with multiple classes.
- **Data Exploration/Transformation**: Integrate and process multi-modal data.

- **Build Framework**: Design a model capable of handling and learning from multiple data types.
- **Train**: Train the multi modal model.
- **Test**: Evaluate the model's performance across modalities and classes.
- **Analyse Results (Deliverable)**: Present a comprehensive analysis of the final model's capabilities and limitations.

Phase 4 – Final Report and Presentation

- **Final Report (Deliverable)**: A complete documentation of the project, including methodology, results, and conclusions.
- **Group Presentation (Deliverable)**: A formal presentation summarising the project journey, findings, and impact.

7.2 Schedule

In terms of the scheduling for the project, the project had tended to steer off course compared to the original proposed 5 phase plan set out in the project proposal seen in Figure 14. To compare how the project had gone off course, this section will cover the 5 phases in the sections below referring the Gantt Chart seen in Figure 15 and describe the blockers the team faced during each phase.

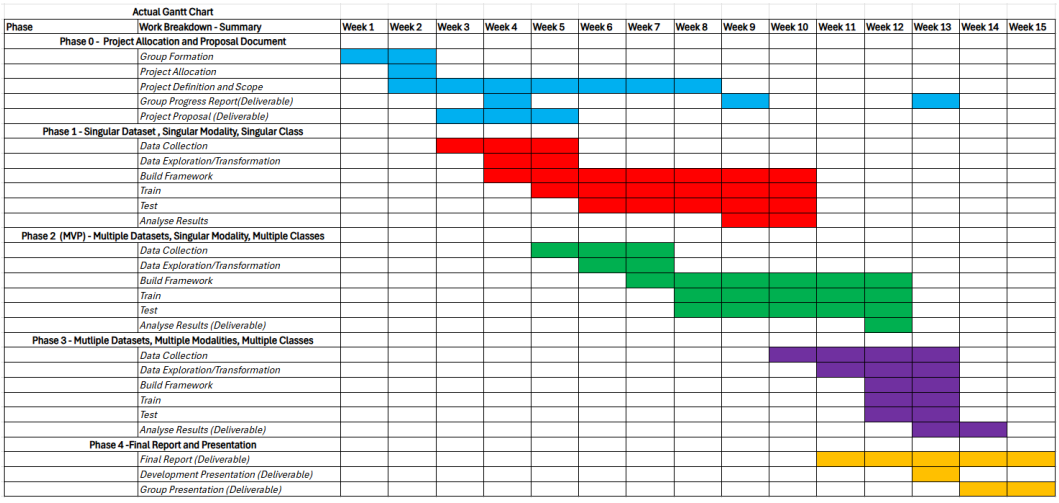


Figure 15: Gantt Chart - Real Project Timeline

7.2.1 Phase 0 - Project Allocation and Proposal Document

Phase 0 consisted primarily of the initial group formation and project proposal that the team undertook. Originally this phase was aimed to be completed within the first few weeks of the project where aspects such as the project definition and scope would be defined by week 4 and having the *Project Proposal* submitted by the end of week 5. Though as seen in the *'Actual Gantt Chart'* it can be seen that though the *Project Proposal* was completed at the end of week 5, misunderstandings by the team and client's requirements had caused blockers and delays with defined project development scope for *Phase 2*.

7.2.2 Phase 1 - Singular Dataset, Singular Modality, Singular Class

Phase 1 aimed to complete a base model and to teach the team on the core concepts of federated learning but also build the base framework for the *Phase 2* and *Phase 3* models. As a result, the original proposed plan for this phase was relatively short with the data collection and transformation taking 2 weeks, the building and training of the model being 3 weeks plus the testing and analysis occurring within a week. Though in practice, the development of this phase hit some rather large roadblocks causing the completion of *Phase 1* to extend by 4 weeks to the end of week 10. This was caused mainly due to the various implementations of different machine learning models that the team used to determine for future phases as well as blockers from the reporting team on providing the research on which models to try implementation. Also with the large period of time required for *Phase 1* but also it not being the MVP or final product the team aimed to achieve, the team had re-allocated resources into the future phases of the project.

The extra time spent on the development of phase 1, did provide benefits for the remaining phases, such as having a simpler environment to test the parameters of different models, leading to different implementations such as a *Custom CNN*, *VGG11* and *VGG16* which reduced the development time of future phases.

7.2.3 Phase 2 (MVP) - Singular Dataset, Singular Modality, Multiple Classes

The development of *Phase 2* began on the proposed schedule even though *Phase 1* was being delayed due to the previously mentioned blockers. The team did this by re-allocating two of the team members to try different frameworks to achieve the goal of the phase, though with only one machine learning model instead of the multiple trialled in *Phase 1*.

Similarly to *Phase 1*, the development of this phase also ran into its share of blockers causing the project to extend by one week, though unlike *Phase 1*, by week 9 the team had a working federated learning model aiming for an early completion. Though, when reviewing the model, the client had noted two parts, firstly there was a misunderstanding on what the scope of *Phase 2* with the modalities and classes and the client had requested that the team implement a custom federated learning model, opting the team to start the development of a *Custom Federated Average* model. This caused the development of *Phase 2* to extend to the end of week 11, while the model was being optimised. It is also worth noting that the building, training and testing of the model were happening in the same weeks as the work in *Phase 1* had familiarised the team with federated learning making the phase easier to develop.

7.2.4 Phase 3 - Multiple Datasets, Multiple Modalities, Multiple Classes

The last development phase, followed the proposed plan for development with only minor changes such as training and testing, caused by the one major blocker caused in this phase being the dataset collection. For this phase, the team had struggled to find two to three suitable datasets that would match the input sizes or desired modalities and same classes. Though this did not impact the completion date for *Phase 3*, it did cause a limited amount of different model testing as seen in *Phase 1*.

7.2.5 Phase 4 - Final Report and Presentation

The last phase of the project focused on the final reports and presentations, which when compared to the proposed timeline had two minor adjustments. The first was that due to the changes in the development phases, the final report was started one week later then expected and the addition of the *Development Video Presentation* which was not initially scoped for the project. Though overall *Phase 4* was completed on schedule with no major blockers.

8 Results and Discussion

8.1 Phase 1

Starting with Phase 1, this phase as previously mentioned focused on building the knowledge base for the team on federated learning, i.e. how to implement a basic version, types of datasets and the types of machine learning models that should be attempted in *Phase 2* and *Phase 3*. Therefore, with the setup showcased in the *Phase 1 framework* where each client shared the same modality and the same number of class with a singular dataset, the team used this environment to test the multiple machine learning models.

Basically only the Lung and Colon Cancer Histopathological Images dataset also known as the LC25000 is used in phase 1 having each client the same modality and the same class but having different tests of both iid and non-iid distribution of the data.

In the initial experiments using a simple CNN, we compared model accuracy under IID and non-IID data splits over 80 communication rounds. As shown in Figure 12, the model under IID conditions achieved smooth convergence, reaching approximately 0.95 accuracy. The accuracy increased steadily across rounds, showing low variance and consistent improvement. This indicates that the updates from each client were aligned, allowing the global model to benefit from coherent aggregation.

In contrast, the non-IID curve displayed far more volatility in the early training rounds, with noticeable fluctuations in the first 20 rounds. These instabilities might due to label imbalance or class skew across clients. However, after approximately 30 rounds, the non-IID model stabilized and eventually reached similar performance to the IID case. This demonstrates that although non-IID distributions introduce training noise and slower early convergence, the FedAvg strategy remains effective and capable of achieving high final accuracy with sufficient training.

In a breakdown by client under IID conditions (Figure 13), all three clients demonstrated consistent convergence, improving from around 0.75 to more than 0.95 accuracy over 80 rounds. The weighted average (black line) tracked closely with the individual client performances, indicating stable and balanced global learning.

To complement these results, we conducted experiments using the VGG11 architecture under both IID and non-IID conditions, but only 5 rounds with the computation constraint. Under IID settings (Figure 14), all clients converged rapidly within just two rounds, reaching around 0.99 accuracy. The curves of all clients and the global model were nearly indistinguishable, showing the strong learning capacity of VGG11 and its effectiveness in uniform data settings. However, when we trained it under non-IID conditions (Figure 15), client performance diverged. Due to the difference in distribution, the accuracy curves showed instability, especially for Client 2, whose accuracy oscillated sharply between 0.97 and 1. The global model (black line) also reflected these fluctuations, though to a

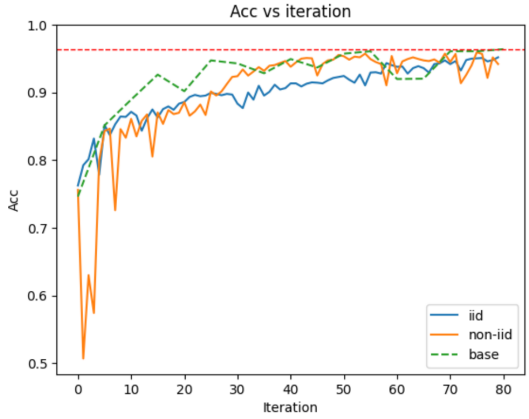


Figure 16: Phase 2 Results - 80 Rounds

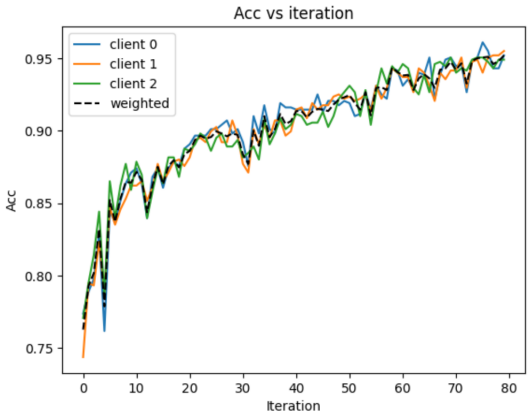


Figure 17: Phase 2 Results - 80 Rounds

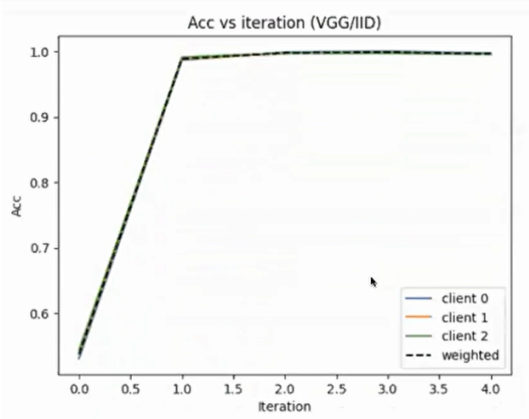


Figure 18: Phase 1 Results - 5 Rounds

lesser extent. These results underscore that while VGG11 is capable of high accuracy, non-IID distributions can still disrupt local convergence and destabilize federated training in the early rounds.

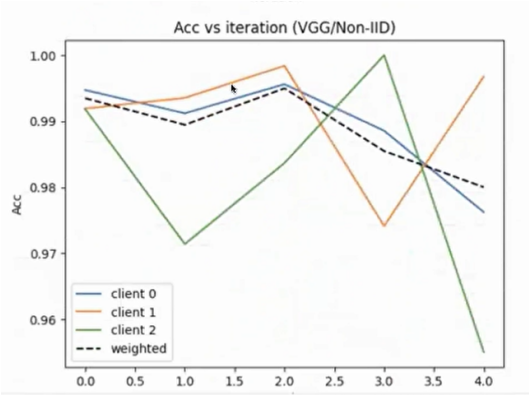


Figure 19: Phase 1 Results - 5 Rounds

8.2 Phase 2

The second phase of development for the project focused on building upon the insights of phase by adjusting the model to have each client contain a different number of classes where some could be overlapping but still keeping the image modality consistent between the clients. Using the results showcased in *Phase 1*, the team had opted to use a custom CNN as it had outputted similar results to the VGG11 while allowing the team to have the ability to add and remove layers and test parameters. The results in the earlier phase also showcased that FedAvg was the optimal choice for the problem presented in this report. Instead of implementing a previously made version of FedAvg, the team decided to implement a custom FedAvg for greater model customisation as discussed in the *Phase 2 Framework*.

For *Phase 2* the model was first run with twenty rounds as seen in Figure 20. In this experiment it can be seen that the accuracy had achieved an accuracy of 91.5% with a distributed loss of 0.189. Though when closely looking at the 'Loss Progression Over Rounds' graph, it can be seen that the loss curve had not stabilised indicating that the number of rounds needed to be further increased to achieve maximum accuracy.

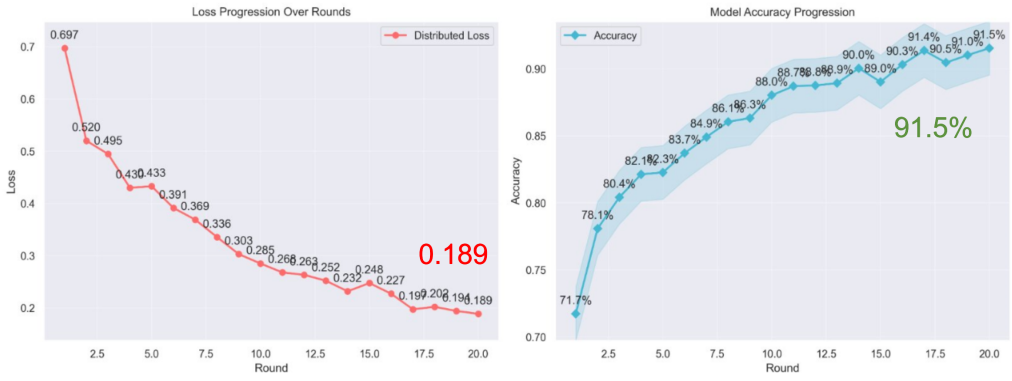


Figure 20: Phase 2 Results - 20 Rounds

Therefore, the model was tested with the parameters mentioned in section 5.2.2 with fifty rounds as seen in Figure 21. Starting with the 'Loss Progression Over Rounds' graph, it can be seen that

the distributed loss had reached a much lower value of 0.069 with the model accuracy showcasing a 95.6% accuracy. Comparing the results with the initial twenty round test, it is clear that more rounds on the model showed greater results with the distributed loss curve becoming closer to flattening out. Though, when looking deeper into this graph, it can be seen that the loss still had the potential to reduce meaning that the accuracy on the model could still be increased with further training rounds. Though due to time and hardware constraints, only fifty rounds was tested and further optimisation of the model parameters would fall into future works with *Phase 3* being the project focus.

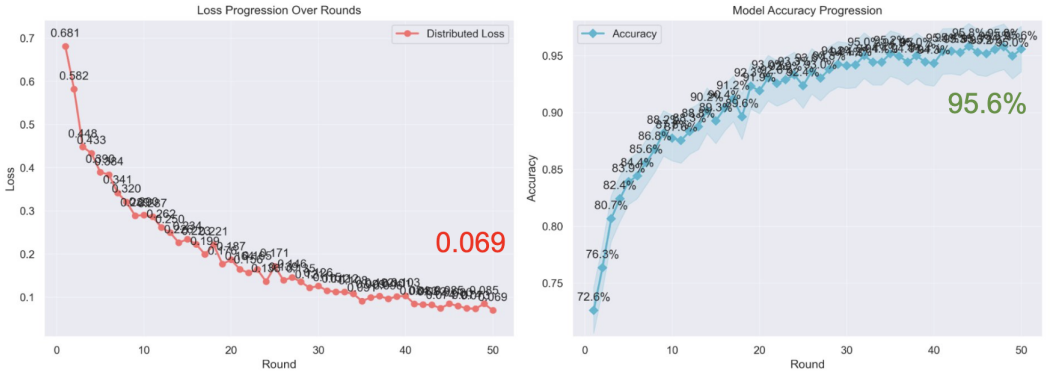


Figure 21: Phase 2 Results - 50 Rounds

Comparing the results with *Phase 1*, it is clear that the overall percentage was not as high as seen in *Phase 1*. This is mainly due to the reduced complexity of the model when compared to VGG11 and the expected increased complexity of this phase, the overall reduction in accuracy was minimal with the choices of these models being implemented in the future phase being the optimal step forward.

8.3 Phase 3

Building on the outcomes of *Phase 2*, which demonstrated the effectiveness of a custom CNN architecture and a tailored FedAvg implementation under varied class distributions, *Phase 3* aimed to increase the model's capabilities by modifying the model to be able to handle different modalities with each client having a different number of classes. For this a MultiModalityMultiHeadCNN was implemented as showcased in the *Phase 3* framework, where instead of testing different machine learnign models, the focused switched to testing three different types of federated learning strategies, *FedAvg, FedOpt* and *FedProx*.

For *Phase 3*, the first plot as shown in Figure 22 below illustrates the performance of FedAvg over fifty communication rounds. In each round, all clients train locally, and the server averages their weights by sample count. Client 0, which holds a balanced histopathology set, rapidly attains high accuracy. It surpasses 0.95 by round 5 and approaches 0.99 by round 50. The blue curve is smooth because each class is equally represented, so local updates are consistent. Client 1, with its imbalanced X-ray data, starts lower around 0.75. Its orange curve shows more fluctuation early on. Nonetheless, by round 20 it exceeds 0.90 and by the final rounds reaches about 0.96. This behaviour highlights how FedAvg can handle differing dataset sizes but may struggle briefly with class imbalance.

The second plot shows FedOpt shown in Figure 23, which layers a server-side optimiser on top standard averaging model which is also the FedAvg model. The adaptive updates use parameters

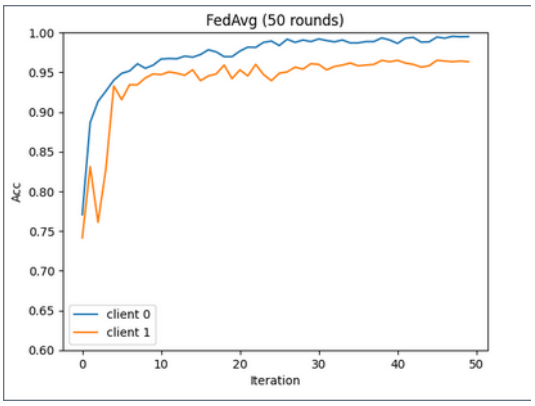


Figure 22: Phase 3 Results: FedAvg

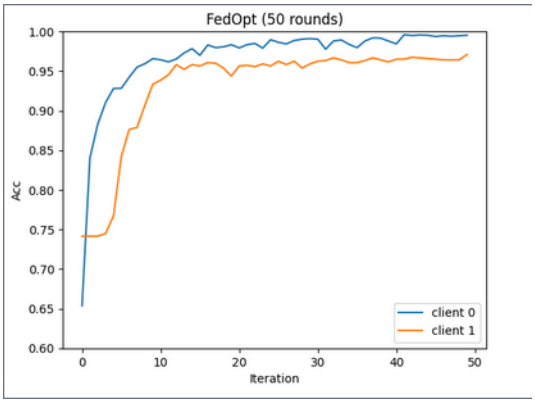


Figure 23: Phase 3 Results: Accuracy FedOpt

eta, beta_1, and beta_2 to adjust learning rates based on past gradients. Client 0 benefits most, reaching 0.95 accuracy by round several rounds earlier than FedAvg and climbing steadily to 0.99. The steeper ascent reflects how the server side optimisers limits noisy updates and accelerates convergence under nonIID noise. Client 1 also seen to improve faster. After initial volatility, it crosses 0.90 by round 10 and settles near 0.97 by round 50.

The third plot in Figure 24 presents FedProx, which adds a proximal penalty term to each client's local loss. This penalty, weighted by proximal_mu, discourages local model weights from drifting too far from the global model. Client 0 again converges quickly, exceeding 0.95 by round 4 and nearing 0.99 by the end. Client 1 shows a slightly slower initial rise compared to FedOpt but smoother updates than plain FedAvg. It crosses 0.90 by round 8 and ends around 0.96–0.97. The proximal constraint reduces oscillations in the client 1 curve. This stability confirms that FedProx effectively limit extreme local updates, especially when a client data are highly non-IID.

The FedOpt chart as shown in Figure 25 displays accuracy for fifty federated rounds when a server-side optimiser is applied. Client 0 (histology) begins near 0.65 and soar past 0.90 by round 3. It touches 0.95 around round 6 and slowly climb toward 0.99 by the end. These steep early gains show how the optimiser adaptive step size accelerates convergence on a balanced data client. Client 1 (X-ray) starts higher then near 0.75, but still benefits from the optimiser. Its accuracy climbs above 0.90 by round 8 and plateaus just under 0.97 after round 30. The orange curve is smoother than in

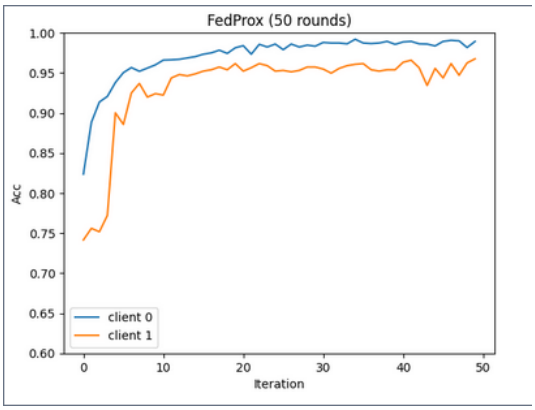


Figure 24: Phase 3 Results: Accuracy FedProx

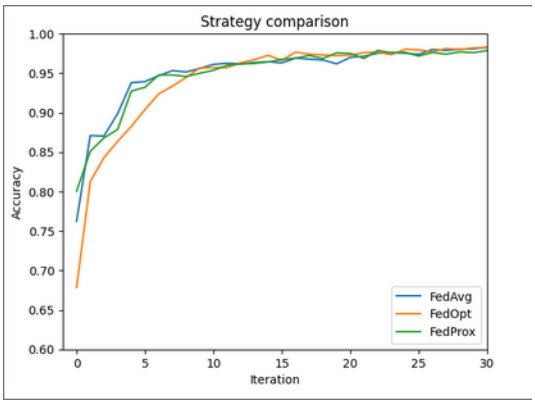


Figure 25: Strategy Comparison

FedAvg but is a bit noisier than client 0. This suggest a potential class imbalance. Between the three strategies implemented, FedOpt delivers the quickest rise among them. It achieves this by reducing early oscillations while also achieve final accuracies of 0.99 in client 0 and 0.97 in client 1.

Phase 3 demonstrated the model's ability to handle increased heterogeneity through the use of a MultiModalityMultiHeadCNN and evaluated the performance of three federated learning strategies under these conditions. While all three methods—FedAvg, FedOpt, and FedProx achieved high accuracies across both clients, FedOpt consistently delivered the fastest convergence and highest stability, specifically in the initial rounds. FedAvg performed well but showed sensitivity to class imbalance, while FedProx offered a balanced trade-off between stability and performance by mitigating the effects of non-IID data through its proximal term. This suggests that while FedAvg remains a strong baseline, adaptive and regularised approaches like FedOpt and FedProx may offer significant advantages in more complex and real-world federated learning scenarios. This could help steer the direction for future works in federated learning image classification models.

9 Limitations and Future Works

9.1 Limitations

While we demonstrated the feasibility and advantages of federated learning (FL) across various imaging modalities and heterogeneous conditions in this project, there are still limitations about it that offer us opportunities for future improvement:

- **Limited number of clients and rounds**: Due to resource constraints, the number of simulated clients and training rounds was relatively small, and we can see that the accuracy might even be higher if we train the model with more rounds. Thus, this may not fully represent the scalability challenges of real-world FL deployments.
- **Simplified simulation environment**: Although Flower provided a flexible simulation platform, it does not replicate all practical deployment factors, such as unreliable client availability, asynchronous updates, or network failures.
- **Security and privacy mechanisms**: Although federated learning inherently enhances data privacy, the current setup of the model we built did not include differential privacy, secure aggregation, or adversarial robustness which are all important in healthcare FL applications.
- **Data complexity**: Although we implemented a custom CNN and explored architectures like VGG11, we only evaluated the model on relatively small datasets. larger, real-world clinical datasets is needed to assess generalizability and further testing.

9.2 Future Works

Future work will focus on several key directions to build upon the current system and make it more robust, scalable, and practical for real-world deployment.

- **Broader model experimentation**: Implement and evaluate a wider range of machine learning models, to better understand their suitability for federated learning scenarios.
- **Stronger privacy protections**: Integrating privacy-preserving techniques such as differential privacy, noise injection, or encrypted model updates etc. to help protect sensitive data and improve the system's resilience against inference attacks.
- **Optimisation**: Future iterations of our aggregator will explore dynamic weighting strategies, adaptive learning rates, and potential integration with personalised FL methods to improve learning across highly heterogeneous clients.
- **Deployment in real environments**: To deploy the framework on actual real world systems such as Raspberry Pi clusters or cloud-edge setups to validate its real-time performance, fault tolerance in real-world settings and etc.

10 Conclusion

This project successfully designed, implemented, and evaluated a federated learning pipeline for medical image classification across multiple phases of experimentation. Starting from Phase 1, trying different methodologies and having the baseline testing with same class and same modality for all the clients to see the core functionality and impact of IID vs. non-IID data distributions on

model convergence. Then optimizing custom CNN architecture and a tailored FedAvg aggregation strategy in phase 2, which significantly improved accuracy across diverse imaging modalities. And extended the framework in phase 3 to simulate heterogeneous clients and validated its performance under more realistic, multi-modal data conditions. Overall, the project proves that federated learning being scalable, privacy-friendly, and flexible enough to handle different data types and devices. Moreover, this project provides insights of deploying collaborative machine learning systems in distributed healthcare environments by combining robust model design, federated training, and privacy awareness.

References

[1] L. Ahmed, K. Ahmad, N. Said, B. Qolomany, J. Qadir, and A. Al-Fuqaha, "Active learning based federated learning for waste and natural disaster image classification," *IEEE Access*, vol. 8, pp. 208 518–208 531, 2020.

[2] Y. Guo, Y. Liu, E. M. Bakker, Y. Guo, and M. S. Lew, "Cnn-rnn: A large-scale hierarchical image classification framework," *Multimedia Tools and Applications*, vol. 77, no. 8, pp. 10 251–10 271, Apr 2018.

[3] E. E. Alam, "Enhancing image classification with federated learning: A comparative study of vgg16 and mobilenet on cifar-10," *arXiv*, 2024.

[4] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Computing Surveys*, vol. 54, no. 6, pp. 131:1–131:36, 2022. [Online]. Available: https://doi.org/10.1145/3460427

[5] X. Zhu, D. Wang, W. Pedrycz, and Z. Li, "Horizontal federated learning of takagi–sugeno fuzzy rule-based models," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 9, pp. 3537–3547, 2022. [Online]. Available: https://doi.org/10.1109/TFUZZ.2021.3118733

[6] J. Pei, W. Liu, J. Li, L. Wang, and C. Liu, "A review of federated learning methods in heterogeneous scenarios," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 5983–5999, 2024.

[7] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin, "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 061–10 071.

[8] H. A. Madni, R. M. Umer, and G. L. Foresti, "Federated learning for data and model heterogeneity in medical imaging," *arXiv preprint*, vol. abs/2308.00155, 2023. [Online]. Available: https://arxiv.org/abs/2308.00155

[9] A. Hassani and I. Rekik, "Unifed: A universal federation of a mixture of highly heterogeneous medical image classification tasks," in *Proceedings of the International Workshop on Machine Learning in Medical Imaging (MLMI), MICCAI*. Springer, July 2024, pp. 32–42. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-73290-4_4

[10] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8397–8406.

[11] Y. Peng, J. Bian, and J. Xu, "Fedmm: Federated multi-modal learning with modality heterogeneity in computational pathology," 2024. [Online]. Available: https://arxiv.org/abs/2402.15858

[12] P. P. Liang, Y. Lyu, X. Fan, J. Tsaw, Y. Liu, S. Mo, D. Yogatama, L.-P. Morency, and R. Salakhutdinov, "High-modality multimodal transformer: Quantifying modality & interaction heterogeneity for high-modality representation learning," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2203.01311

[13] T. Feng, D. Bose, T. Zhang, R. Hebbar, A. Ramakrishna, R. Gupta, M. Zhang, S. Avestimehr, and S. Narayanan, "Fedmultimodal: A benchmark for multimodal federated learning," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge*

Discovery and Data Mining (KDD '23). ACM, 2023, pp. 4035–4045. [Online]. Available: https://doi.org/10.1145/3580305.3599447

[14] W. Liu, Y. Zheng, Z. Xiang, Y. Wang, Z. Tian, and W. She, "An efficient federated learning method based on enhanced classification-gan for medical image classification: An efficient federated learning method," *Multimedia Systems*, vol. 31, no. 1, 2025.

[15] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. A. Emadi, M. B. I. Reaz, and M. T. Islam, "Can ai help in screening viral and covid-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132 665–132 676, 2020.

[16] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images," *Pattern Recognition Letters*, vol. 138, pp. 638–643, 2020.

[17] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. A. Maadeed, S. M. Zughaier, M. S. Khan, and M. E. H. Chowdhury, "Exploring the effect of image enhancement techniques on covid-19 detection using chest x-ray images," *Computers in Biology and Medicine*, vol. 132, 2021.

[18] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis," *Scientific Reports*, vol. 12, p. 1953, 2022. [Online]. Available: https://doi.org/10.1038/s41598-022-05539-7

[19] S. Lakrouni, S. Bah, M. Sebgui, N. Gupta, A. Castañeda, and C. Enea, "Federated learning for enhanced medical image analysis," *Networked Systems*, vol. 14783, pp. 157–170, 2024. [Online]. Available: DOIordirectlinkifavailable

[20] B. Lutnick, D. Manthey, J. U. Becker, M. C. Montalto, R. Yacoub, and P. Sarder, "A tool for federated training of segmentation models on whole-slide images," *Journal of Pathology Informatics*, vol. 13, p. 100101, 2022.

[21] F. Wagner, W. Xu, P. Saha, Z. Liang, D. Whitehouse, D. Menon, N. Voets, J. A. Noble, and K. Kamnitsas, "Feasibility of federated learning from client databases with different brain diseases and mri modalities," *IEEE Access*, 2023.

[22] A. A. Borkowski, M.-M. Bui, L. B. Thomas, C. P. Wilson, L. A. DeLand, and S. M. Mastorides, "Lung and colon cancer histopathological image dataset (lc25000)," *arXiv preprint arXiv:1912.12142v1*, 2019.

[23] A. A. Ndiaye, "Cervical_cancer-cancer," https://huggingface.co/datasets/Alwaly/Cervical_Cancer-cancer, 2025, accessed: 2025-05-30.

11 Appendix

11.1 Acknowledgement of AI Usage

Part A: Have you used AI tools in the completion of this assignment? If your answer to this part is "No", you can leave the following Part B and Part C as blank.

Yes

Part B: What automated writing or generative AI tools you have used in the completion of this assignment? Clearly state the name(s) of the tool(s) and including a link to each tool.

Microsoft Co-Pilot: https://copilot.microsoft.com/

Part C: How have you used automated writing or generative AI tools in the assessment?

For this final report, Microsoft Co-Pilot was used in the contribution statement to transform the content from the team's Kanban board into the contributions list for the authors. Microsoft Co-Pilot was also used to help with phrasing in the literature review.
