

The background is a dark blue gradient with abstract geometric shapes. On the right, there are several stacked, semi-transparent cubes in shades of teal and blue. On the left, there is a network of thin white lines connecting small yellow dots. In the center-left, there is a grid of medical scan images, including a large chest X-ray and several smaller cross-sectional scans of organs.

Towards Federated Learning in ML Models

Team: Salva Umar, Karan Khubdikar, Hina Bandukwala, Prabhjit Thind

Mentor: Quan Nguyen

Capstone Partner: ALS GoldSpot Discoveries Ltd.

Team Members

Karan Khubdikar



Prabhjit Thind



Salva Umar



Hina Bandukwala



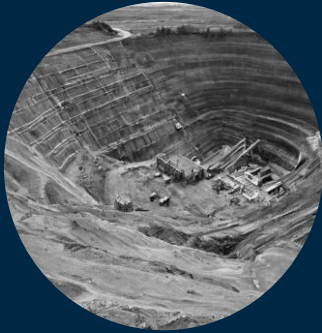
Quan Nguyen
(Mentor)



ALS GoldSpot
Discoveries Ltd.
(Partner)



Capstone Partner: ALS GoldSpot Discoveries Ltd.



Who are they?

- Geoscience expertise with AI and data science for mineral exploration & mining



Why is it important?

- Enhances decision-making in mining operations
- Ensuring sustainable and profitable resource extraction.



Clients don't want to share
their sensitive data for model
training

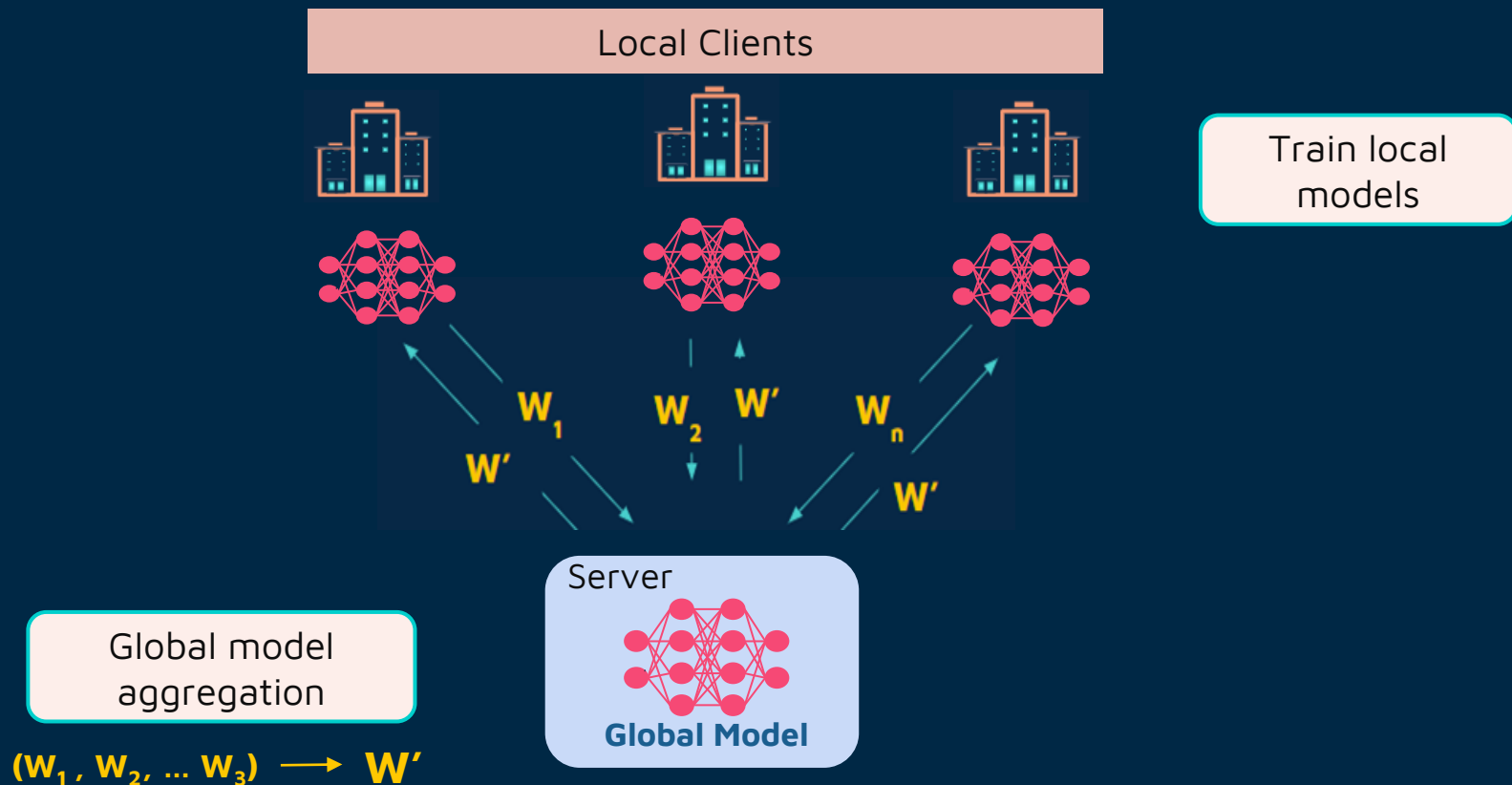
So what's the solution?



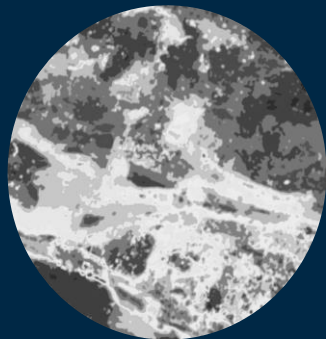
Introduction

01

Federated Learning (Review)



What are the capstone partner's needs?



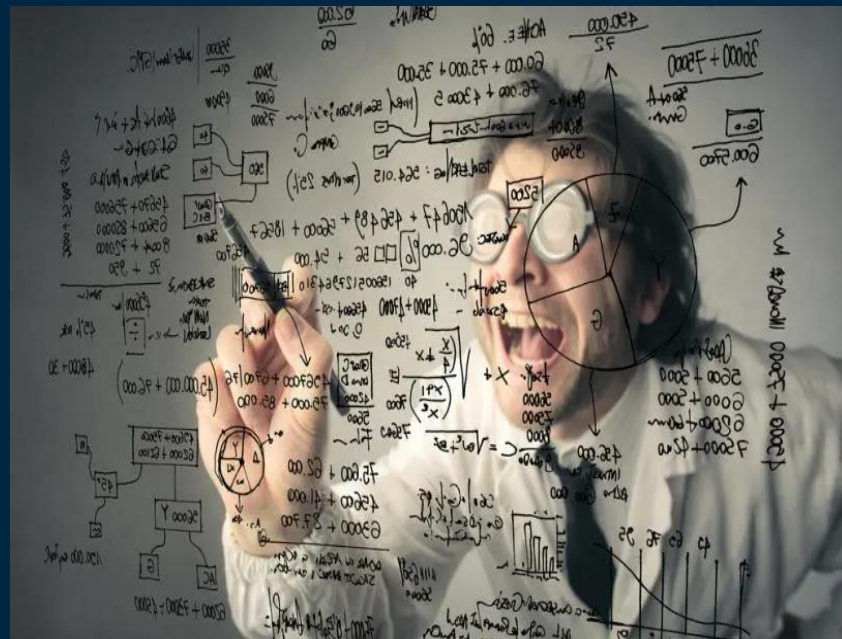
What issues are they facing?

- Communication Efficiency: data is distributed across clients
- Data Heterogeneity: Managing different client data distributions
- Model Training Efficiency: Improving accuracy of ML models in decentralized framework

Scientific Objective



- Investigating effectiveness of decentralized training approaches.
- Creating a federated learning framework to solve business problems in the mining industry.



Data Science Workflow

02

Data Product

Successful Project Delivery will entail:

- ★ **A report:** covering objectives, methodology, experiments, results.
- ★ **A Git repository:** with reproducible code, documentation, experiment configurations.
- ★ **A data pipeline:** processes and tools needed to simulate the federated learning framework.



Data Source

Osteosarcoma dataset:

- Bone cancer images
- Open source dataset is used for testing the applicability of the decentralised framework

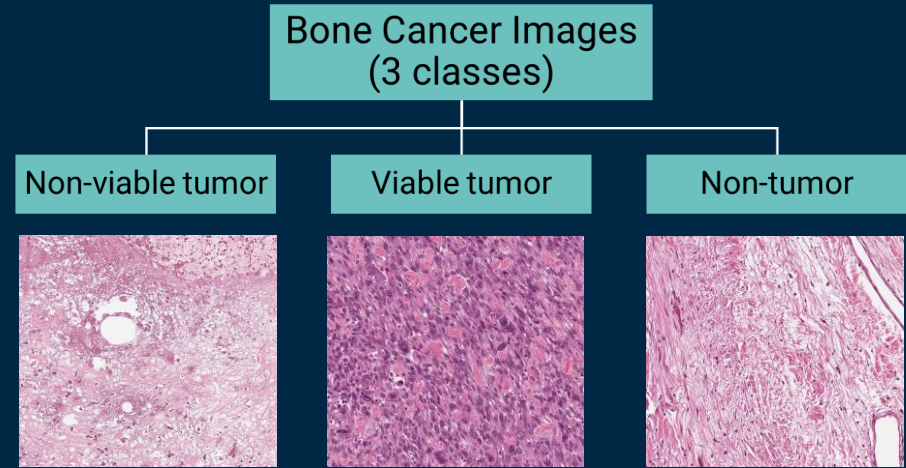
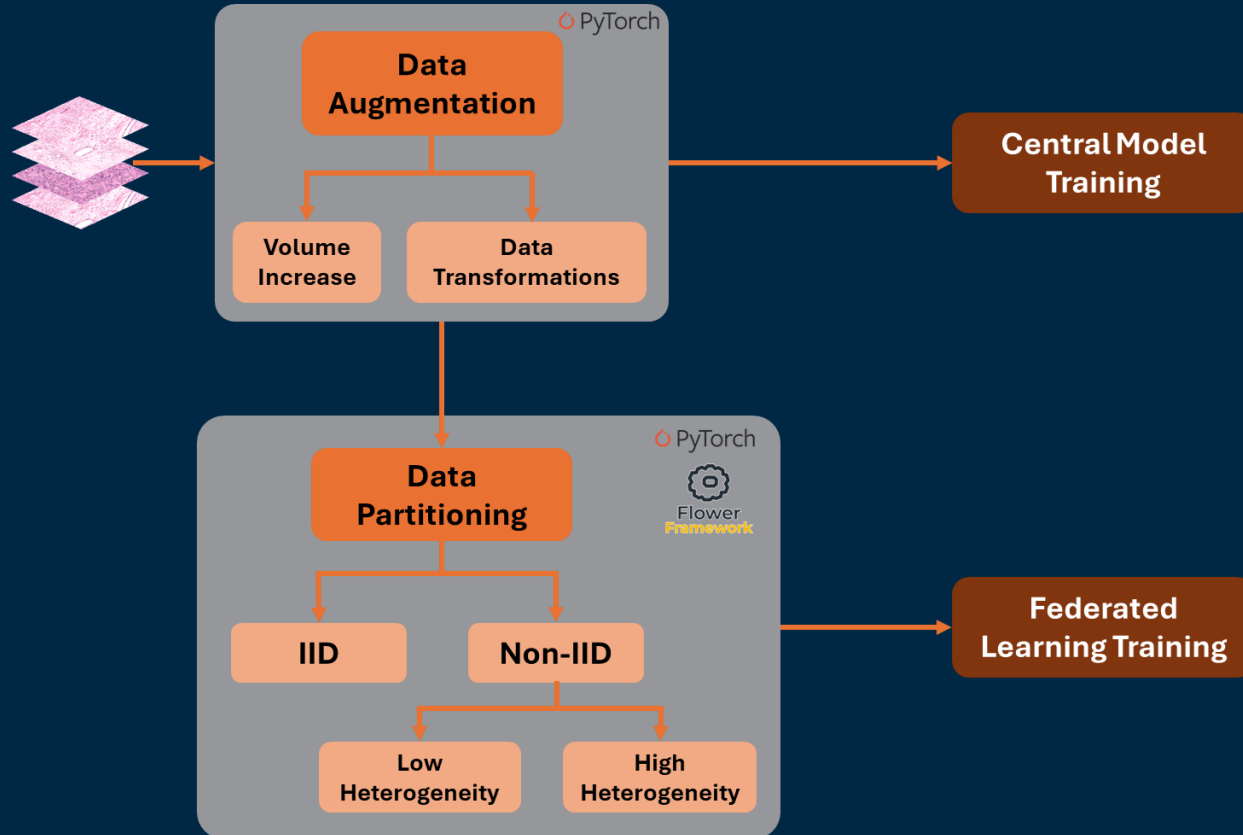


Image specifications

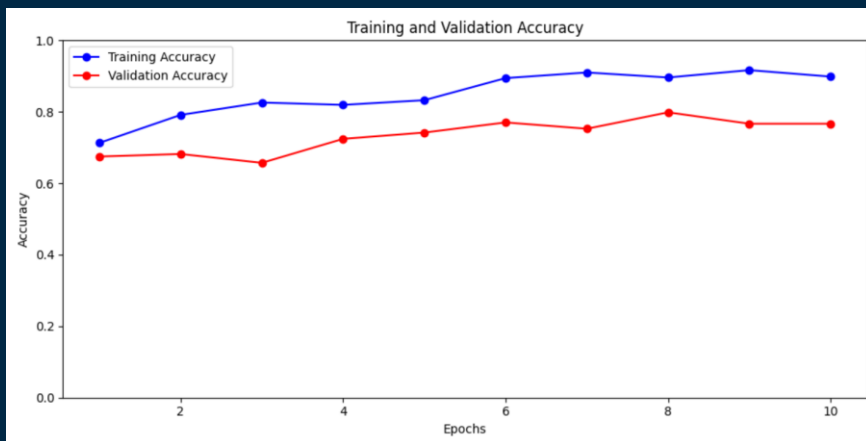
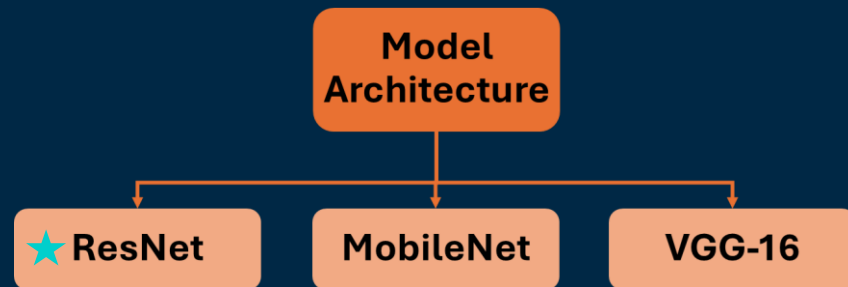
- ❑ 1,144 images
- ❑ 1024 x 1024 at 10X resolution

Dataflow Pipeline



Central Model

- ★ Multi-Class Classification task
- ★ Unfreeze last few layers for fine tuning

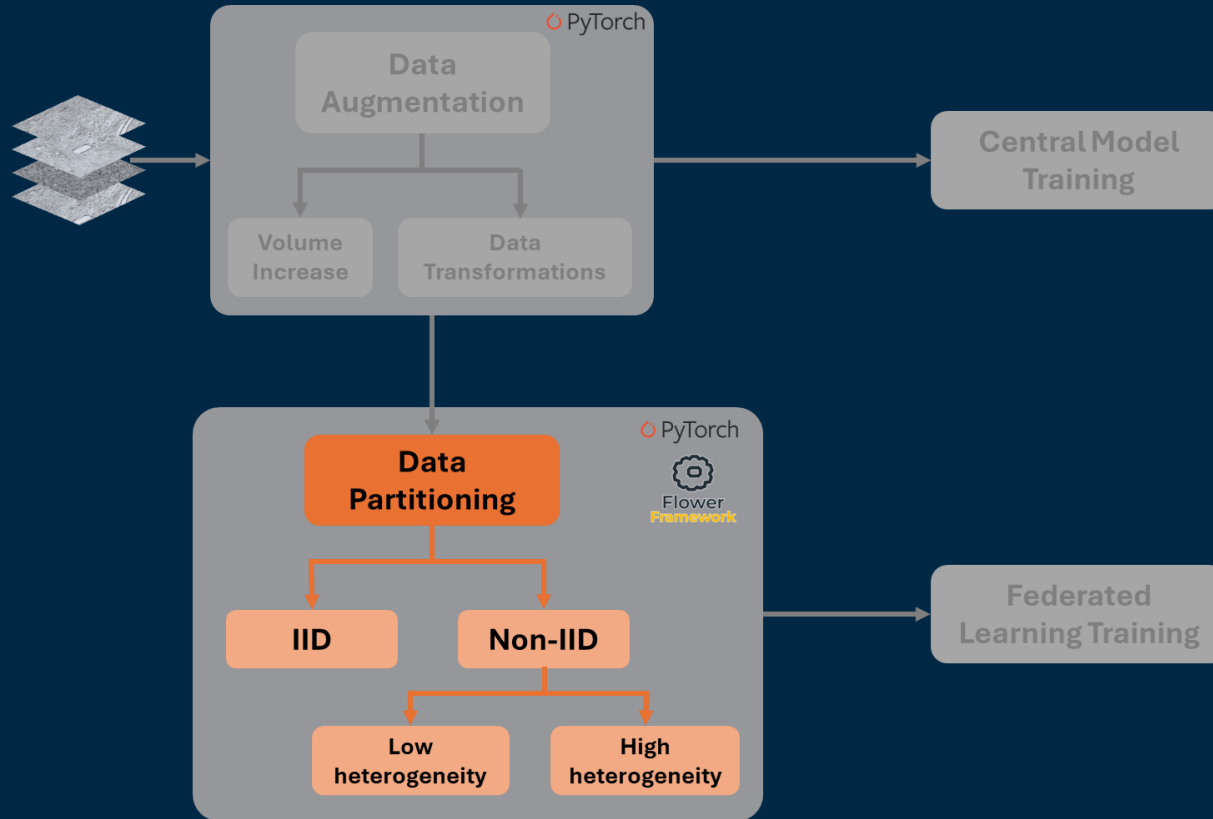


Metric	Value
Balanced Accuracy	0.77
F1 Score	0.81
Precision	0.77
Recall	0.77

Federated Training

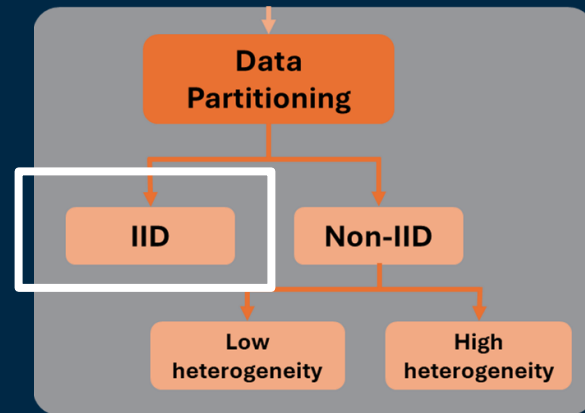
03

Data Partitioning



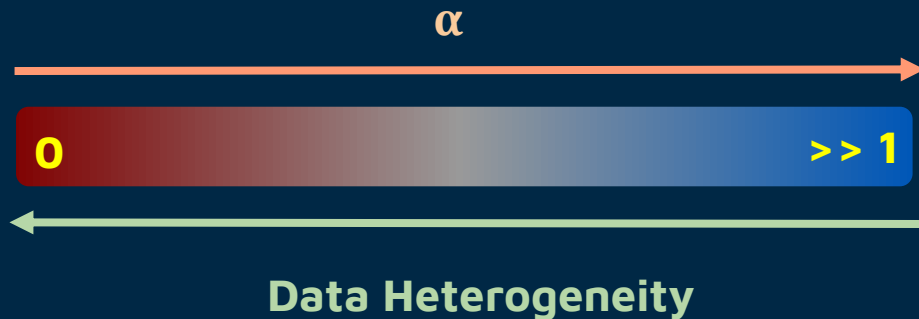
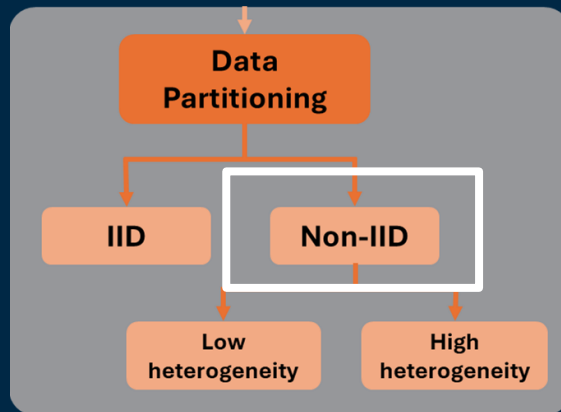
Data Partitioning: IID

- ★ PyTorch *random_split* function
- ★ Each client receives a random sample from the training set
- ★ Minimal heterogeneity between clients



Data Partitioning: Non-IID

- Dirichlet Distribution:
- Controlled by α (Concentration Parameter)
 - Multivariate generalization of beta distribution
- Used to split a dataset among multiple clients
- Exhibit heterogeneity in their class distributions





$\alpha = 0.1$

$\alpha = 0.5$

More skewed

Less skewed

Client 1



Client 2



Client 3



Client 4

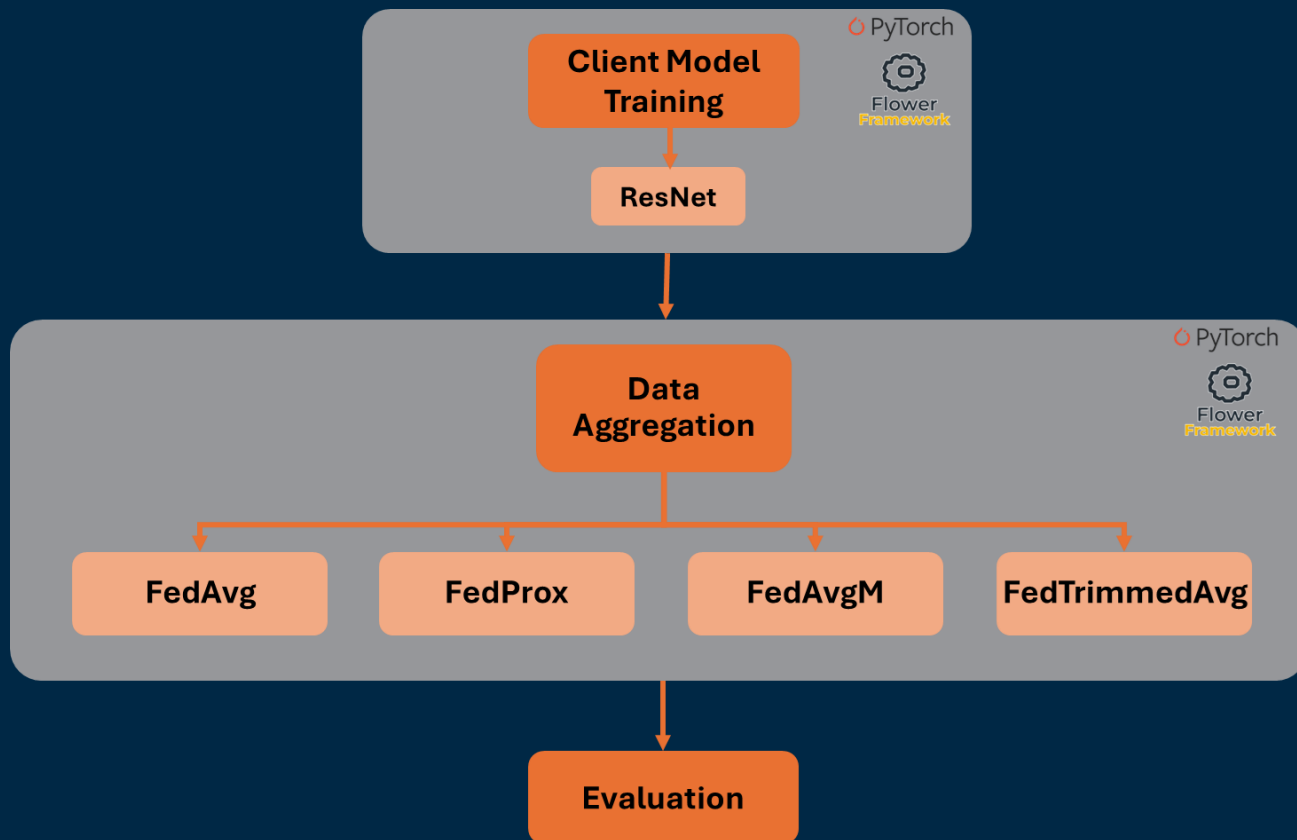


Proportion



Proportion

Client Training



Results

04

Experimental Setup & Evaluation

- ★ We carried out 12 experiments where we tested the following:

- Data Partitioning:

- *iid*
 - *Non-iid* ($\alpha = 0.5$ and 0.1)

- Aggregation strategies:

- FedAvg, FedProx, FedAvgM, FedTrimmedAvg

- ★ All experiments were conducted with:

- 4 clients
 - 5 rounds of FL
 - Learning rate = 0.002
 - Momentum = 0.9

Evaluation Metrics:

BALANCED ACCURACY

F1 -SCORE

RECALL

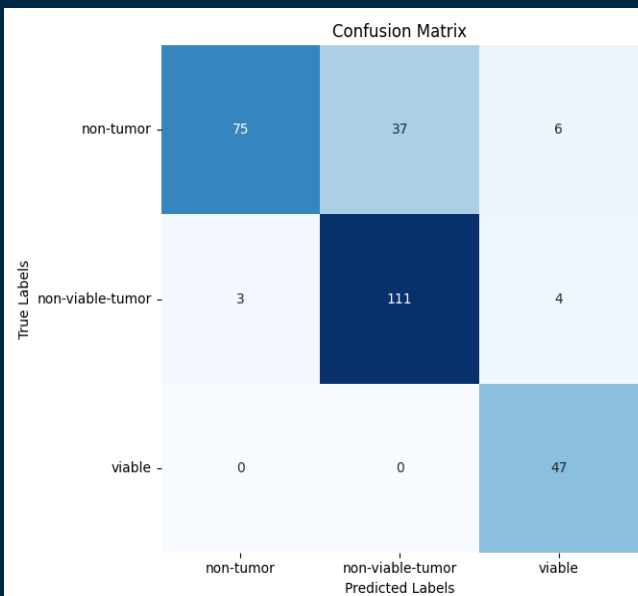
PRECISION

Server Aggregation Methods

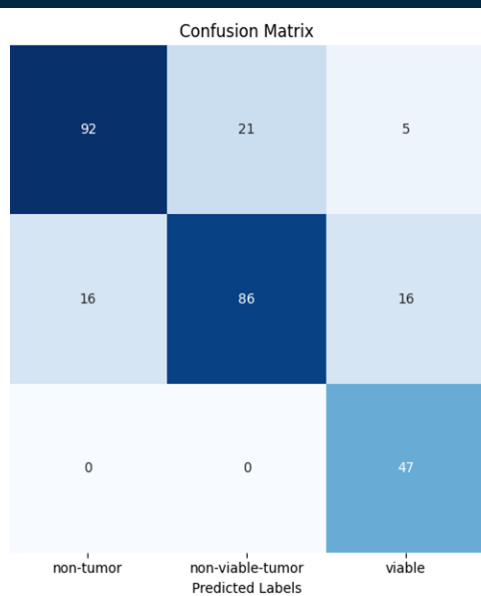
Strategy	Description
FedAvg	Computes a weighted average of the local updates from each client.
FedAvgM	Builds on FedAvg by incorporating momentum parameter, to accelerate convergence by considering past gradients
FedProx	Builds on FedAvg by introducing a proximal term to handle heterogeneity
FedTrimmedAvg	Aggregates model updates by averaging the central 80% of updates

Experiments: Confusion Matrices

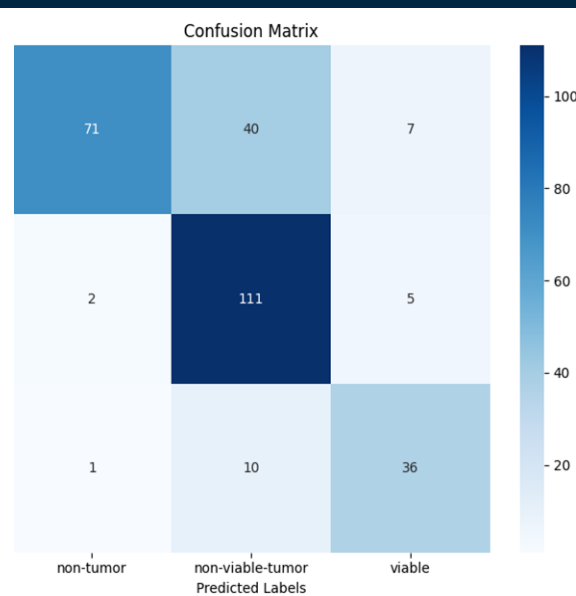
IID



Dirichlet ($\alpha = 0.1$)
More Skewed



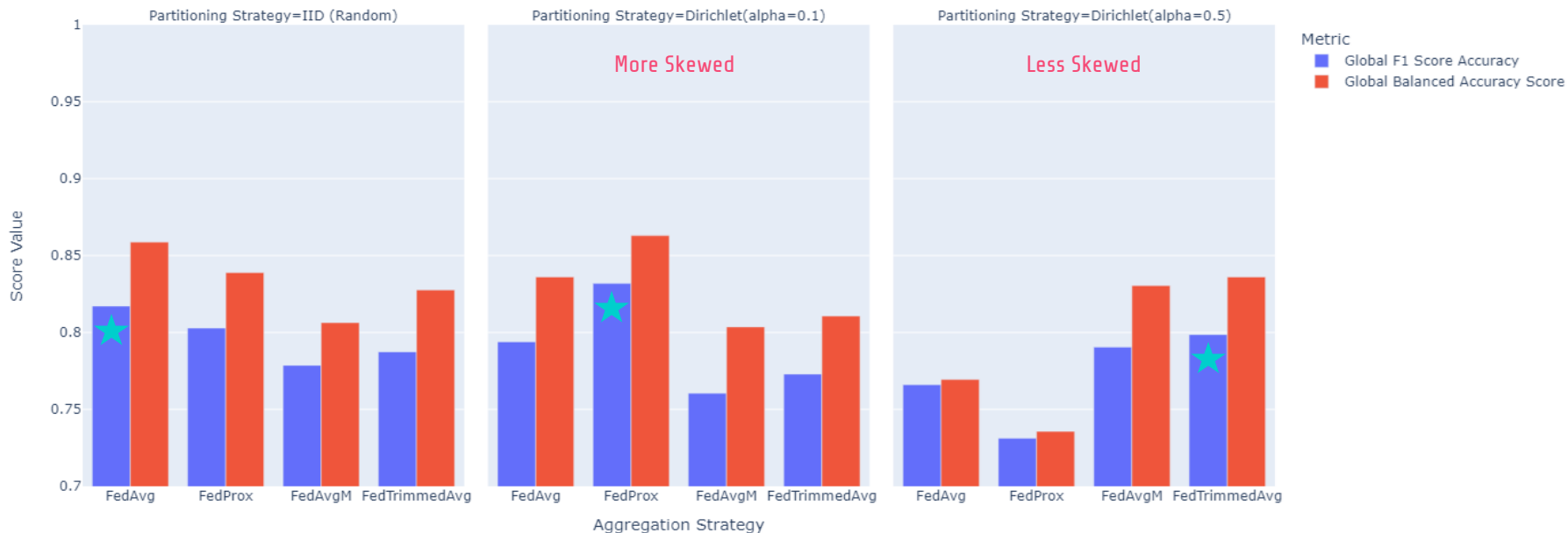
Dirichlet ($\alpha = 0.5$)
Less skewed



Note: FedAvg Results

Experiments: Balanced Accuracy & F1-scores

Comparison of Global F1 Score and Balanced Accuracy across Aggregation Strategies



Experiments: Non-IID Observations

Best Strategy: Dirichlet ($\alpha = 0.1$)

More Skewed

FedAvg

- High variability data
- Focus on balanced accuracy
- General robustness

Balanced Accuracy: 0.84
F1 Score: 0.80
Precision: 0.81
Recall: 0.80

FedProx

- Highly non-IID data
- Regularization focus
- Consistency across metrics

Balanced Accuracy: **0.86**
F1 Score: 0.83
Precision: 0.85
Recall: 0.83

Best Strategy: Dirichlet ($\alpha = 0.5$)

Less Skewed

FedAvgM

- Moderate data imbalance
- High recall requirements
- Balanced performance

Balanced Accuracy: 0.83
F1 Score: 0.79
Precision: 0.82
Recall: 0.80

FedTrimmedAvg

- Moderate non-IID data
- Outlier robustness
- High precision requirements

Balanced Accuracy: **0.86**
F1 Score: 0.84
Precision: 0.85
Recall: 0.83

Design Features

1. Customising Model Parameters: Hydra framework



- Configuration manager
 - To efficiently evaluate across different datasets and parameters
 - Important Parameters:
 - No of clients
 - Rounds of training
 - Path to data directory
 - Partition type
 - Strategy type
- ... many more

Design Features

1. Configurability: Hydra



```
data_dir: "./data/combined"
```

```
batch_size: 4
class_names: ["non-tumor", "non-viable-tumor", "viable"]
input_size: 224
```

```
num_clients_per_round_fit: 4
num_clients_per_round_eval: 4
num_rounds: 5
num_clients: 4
```

```
num_cpus_per_client: 4
gpu_usage_per_client: 0.2
strategy_name: "FedAvg"
```

```
config_fit:
```

```
lr: 0.002
momentum: 0.9
local_epochs: 5
batch_size: 4
decay_rate: 0.1
step_size: 5
gamma: 0.1
```

```
data_partition:
```

```
strategy: "dirichlet"
alpha: 0.1
```

Start Simulation: python main.py

```
2024-06-04 21:45:32,092 [flwr][INFO] Starting Flower simulation, config: num_rounds=5, no_round_timeout
2024-06-04 21:45:37,647 [flwr][INFO] Flower VCE: Ray initialized with resources: {'object_store_memory': 16165516492.0, 'memory': 32331032987.0,
'node_internal_head': 1.0, 'GPU': 1.0, 'node:172.28.0.12': 1.0, 'CPU': 8.0}
2024-06-04 21:45:37,651 [flwr][INFO] Optimize your simulation with Flower VCE: https://flower.ai/docs/framework/how-to-run-simulations.html
2024-06-04 21:45:37,654 [flwr][INFO] Flower VCE: Resources for each Virtual Client: {'num_cpus': 4, 'num_gpus': 0.2}
2024-06-04 21:45:37,691 [flwr][INFO] Flower VCE: Creating VirtualClientEngineActorPool with 2 actors
2024-06-04 21:45:37,675 [flwr][INFO] [INIT]
2024-06-04 21:45:37,679 [flwr][INFO] - Requesting initial parameters from one random client
2024-06-04 21:45:46,428 [flwr][INFO] Received initial parameters from one random client
2024-06-04 21:45:46,431 [flwr][INFO] - Evaluating initial global parameters
2024-06-04 21:45:49,939 [flwr][INFO] initial parameters (loss, other metrics): 1.263537882919043, {'test_bal_accuracy': 0.394993989662219, 'test_f1':
0.25517252023721754, 'test_precision': 0.6158703968694331, 'test_recall': 0.3286219081272085}
2024-06-04 21:45:49,942 [flwr][INFO] -
2024-06-04 21:45:49,946 [flwr][INFO] - [ROUND 1]
2024-06-04 21:45:49,949 [flwr][INFO] configure_fit: strategy sampled 4 clients (out of 4)
2024-06-04 21:46:31,147 [flwr][INFO] aggregate_fit: received 4 results and 0 failures
2024-06-04 21:46:33,089 [flwr][WARNING] - No fit_metrics.aggregation_fn provided
2024-06-04 21:46:36,289 [flwr][INFO] fit progress: (1, 1.9769039609417005, {'test_bal_accuracy': 0.39716312056737585, 'test_f1': 0.3044097497656307,
'test_precision': 0.345644040806727, 'test_recall': 0.4487632508833225}, 46.3470354399996)
2024-06-04 21:46:36,292 [flwr][INFO] configure_evaluate: no clients selected, skipping evaluation
2024-06-04 21:46:36,295 [flwr][INFO] -
2024-06-04 21:46:36,298 [flwr][INFO] - [ROUND 2]
2024-06-04 21:46:36,301 [flwr][INFO] configure_fit: strategy sampled 4 clients (out of 4)
2024-06-04 21:47:17,407 [flwr][INFO] aggregate_fit: received 4 results and 0 failures
2024-06-04 21:47:22,543 [flwr][INFO] fit progress: (2, 2.188792923518192, {'test_bal_accuracy': 0.6638418079096046, 'test_f1': 0.4902542299442604,
'test_precision': 0.7598527912401012, 'test_recall': 0.5795053003535699}, 92.60115222599084)
2024-06-04 21:47:22,546 [flwr][INFO] configure_evaluate: no clients selected, skipping evaluation
2024-06-04 21:47:22,549 [flwr][INFO] -
2024-06-04 21:47:22,552 [flwr][INFO] - [ROUND 3]
2024-06-04 21:47:22,556 [flwr][INFO] configure_fit: strategy sampled 4 clients (out of 4)
2024-06-04 21:48:00,008 [flwr][INFO] aggregate_fit: received 4 results and 0 failures
2024-06-04 21:48:05,160 [flwr][INFO] fit progress: (3, 5.701338005366954, {'test_bal_accuracy': 0.3333333333333333, 'test_f1': 0.24539358317985954,
'test_precision': 0.17385658454968847, 'test_recall': 0.4169611307420495}, 135.2184447489999)
2024-06-04 21:48:05,164 [flwr][INFO] configure_evaluate: no clients selected, skipping evaluation
2024-06-04 21:48:05,168 [flwr][INFO] -
2024-06-04 21:48:05,172 [flwr][INFO] - [ROUND 4]
2024-06-04 21:48:05,176 [flwr][INFO] configure_fit: strategy sampled 4 clients (out of 4)
2024-06-04 21:48:45,489 [flwr][INFO] aggregate_fit: received 4 results and 0 failures
2024-06-04 21:48:50,568 [flwr][INFO] fit progress: (4, 1.70318077618444, {'test_bal_accuracy': 0.7188965019834114, 'test_f1': 0.6016802052909247,
'test_precision': 0.7822973272040019, 'test_recall': 0.6537102473498233}, 180.62626229900002)
2024-06-04 21:48:50,571 [flwr][INFO] configure_evaluate: no clients selected, skipping evaluation
2024-06-04 21:48:50,575 [flwr][INFO] -
2024-06-04 21:48:50,578 [flwr][INFO] - [ROUND 5]
2024-06-04 21:48:50,582 [flwr][INFO] configure_fit: strategy sampled 4 clients (out of 4)
2024-06-04 21:49:30,664 [flwr][INFO] aggregate_fit: received 4 results and 0 failures
2024-06-04 21:49:36,162 [flwr][INFO] fit progress: (5, 0.47380182282716266, {'test_bal_accuracy': 0.8559322033898304, 'test_f1': 0.8197952885501935,
'test_precision': 0.8339563319566574, 'test_recall': 0.819789586572444}, 226.22070998499998)
2024-06-04 21:49:36,166 [flwr][INFO] configure_evaluate: no clients selected, skipping evaluation
```

Design Features

2. Parallel Programming: Ray



- Enhances scalability
- Manages client model training concurrently
- Resource Optimization

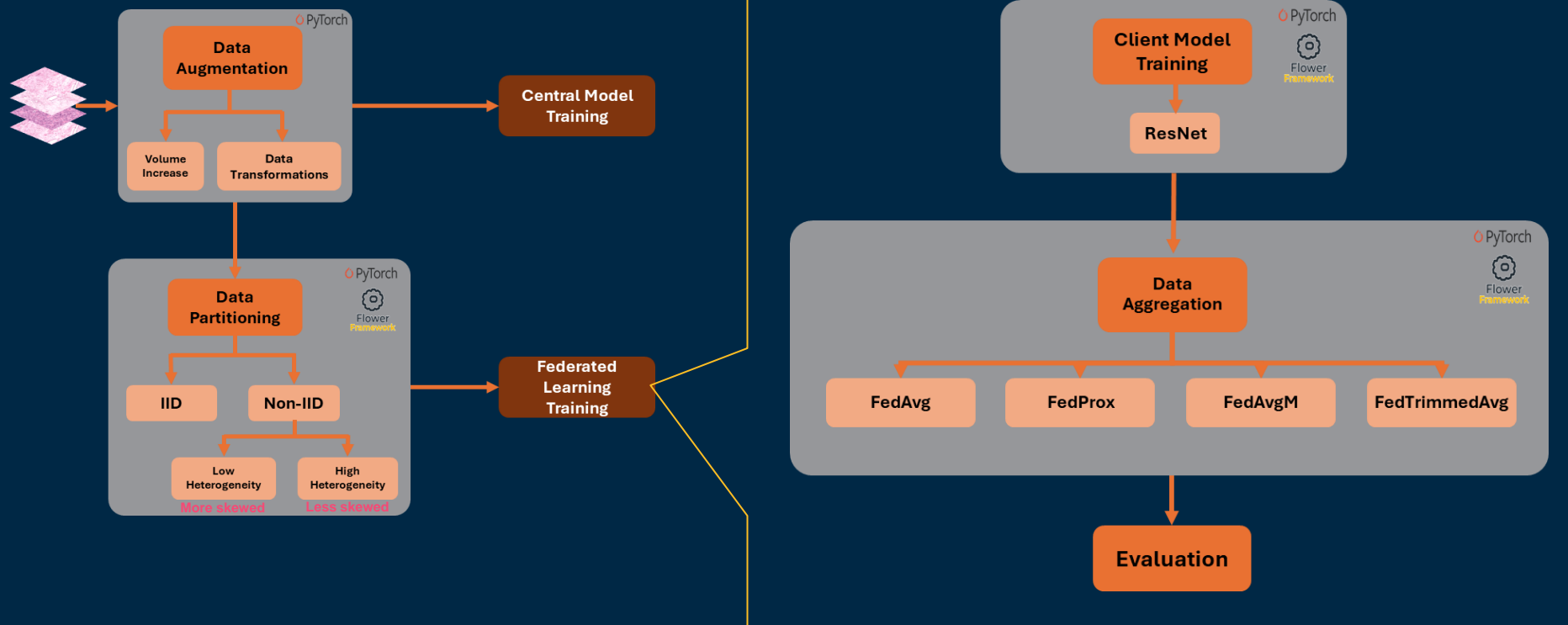
3. Addressing Class Imbalance in Federated Models

- Addition of a decaying learning rate to address overfitting
- Running 5 rounds of training for each client (within each round 10 epochs per local model are run)
- Data Augmentation techniques to increase robustness of the local models

Limitations of current work

01	Limited Data	<ul style="list-style-type: none">• ~2k images after augmentation• Limit on the maximum no of clients to test without increasing overfitting
02	Data Quality and Consistency	<ul style="list-style-type: none">• Variability in data quality and heterogeneity across clients impacts model performance
03	Resource Constraints at Client level	<ul style="list-style-type: none">• Decentralized training can be resource-intensive, and some clients may not have all available across all nodes

Final Product: Big Picture!



Q&A?

Thank you!