**Safeguarding Personal Screens:**

**A Federated Learning Approach**

**to**

**Screenshot Categorization**

**Team 6 :** Federated Learning

**Submitted To :**

Prof. Olin Hartin, Ph.D.

Arizona State University

May 2nd, 2025

**Submitted By :**

Krutyanjay Shinde (1234108189)

Vipul Jethwa (1232525361)

Aditya Pradeep Kogri (1236244206)

Vaishnavi Shridhar Bhat (1235817468)

Ibaad Sarwar (1222624624)

Contents

[1. Introduction 1](#_Toc197107608)

[2. Background Theory 1](#_Toc197107609)

[2.1. Federated Learning Concept 1](#_Toc197107610)

[3. Objective 2](#_Toc197107611)

[4. Dataset and Preprocessing 2](#_Toc197107612)

[5. Model Architecture 2](#_Toc197107613)

[6. System Architecture 3](#_Toc197107614)

[6.1. Client System 3](#_Toc197107615)

[6.2. Server System 3](#_Toc197107616)

[7. Federated Optimization Strategies 4](#_Toc197107617)

[7.1. FedAvg 4](#_Toc197107618)

[7.2. FedYogi 4](#_Toc197107619)

[7.3. FedAdagrad 5](#_Toc197107620)

[7.4. FedAdam 5](#_Toc197107621)

[8. Results 6](#_Toc197107623)

[9. Conclusion 7](#_Toc197107624)

[10. Future Scope 7](#_Toc197107625)

[References 8](#_Toc197107626)

# Introduction

* Screenshots are becoming a more common way for smartphone users to record short-term yet important information, such as recipes, reminders, receipts, learning materials, vacation reservations, or online purchase confirmations. Nevertheless, manual categorization gets taxing as the number of screenshots rises. Users are becoming more reluctant to provide cloud services access to their personal information at the same time that privacy worries are at an all-time high. The core of the issue we seek to address is these two trends: more data and increased privacy demands. Uploading data to cloud servers where machine learning models are developed is the standard method for classifying images.
* Although accurate, this method is in direct opposition to the need for privacy. Therefore, a decentralized approach is required, in which learning can take place locally on the user's device and only knowledge not data is exchanged. This is precisely what Federated Learning (FL) provides: a collaborative training model in which user data stays on the device at all times. For this project, we have created a system that employs federated learning to categorize screenshots into five pre-established groups: real-world scenarios, food, shopping, notes, and entertainment.
* A lightweight convolutional neural network (CNN) is locally trained on each client (device), and the server just receives the model updates. These changes are then combined by the server to enhance a global model. This collaborative learning approach preserves model performance, facilitates distributed training, and protects data privacy.

# Background Theory

## Federated Learning Concept

* Federated learning is particularly useful in real-world applications that involve sensitive user data, such as screenshots taken by an individual. More clients and classes can be added to the model, which also generalizes well to non-IID data. These findings open the door for privacy-preserving AI solutions that are integrated into mobile ecosystems. In the decentralized machine learning paradigm known as "federated learning," several clients work together to train a model without ever exchanging raw data. Only model updates (such weights or gradients) are sent to the central server; the training data is stored locally on each client.
* This method works especially well for sensitive applications where data privacy is important, such as personal finance, healthcare diagnostics, and mobile keyboard recommendations.In our project, FL lets clients learn on their data in secret, with each client having a different collection of user screenshots. The server receives the new model weights once each client finishes its local training cycle. The global model is updated by the server after combining these weights using a federated optimization technique like Federated Averaging (FedAvg). For a predetermined number of communication rounds, this cycle continues. Managing non-IID (non-independent and identically distributed) data between clients is a significant difficulty in FL.
* Federated learning needs to take personalization and data imbalance into consideration because each user may have wildly disparate screenshot distributions and types. FL is still a strong tool for safe, scalable learning in spite of these challenges.



Figure 1: Different devices having different screenshots

# Objective

* This research effectively illustrates that federated learning can be used to classify screenshots while maintaining user privacy. Data exposure is prevented by training on-device and only communicating model updates. When paired with FedYogi and FedAvg optimizers, our CNN-based model produced classification accuracies that were on par with centralized training models.

# Dataset and Preprocessing

* Every client device underwent preprocessing independently. Among the steps were:Eliminating corrupt or illegible images is known as data cleaning.
  + **Normalization:** ImageNet per-channel means and standard deviations are used to standardize pixel intensity values.
  + **Resizing:** To comply with CNN input specifications, all screenshots should be resized to a consistent 224x224 pixel resolution.
  + **Padding:** When necessary, use zero padding while preserving the original aspect ratio.
  + **Splitting:** Making a 70% training set and a 30% testing set out of the local data. The local datasets were made clean, consistent, and prepared for federated training by this approach.

# Model Architecture

* A lightweight CNN model appropriate for mobile training was employed on each client device. Two convolutional layers with max pooling and ReLU activation were part of the architecture.
  + A dense (completely linked) layer comes after a flattening layer.
  + A softmax output layer for classification into five classes.

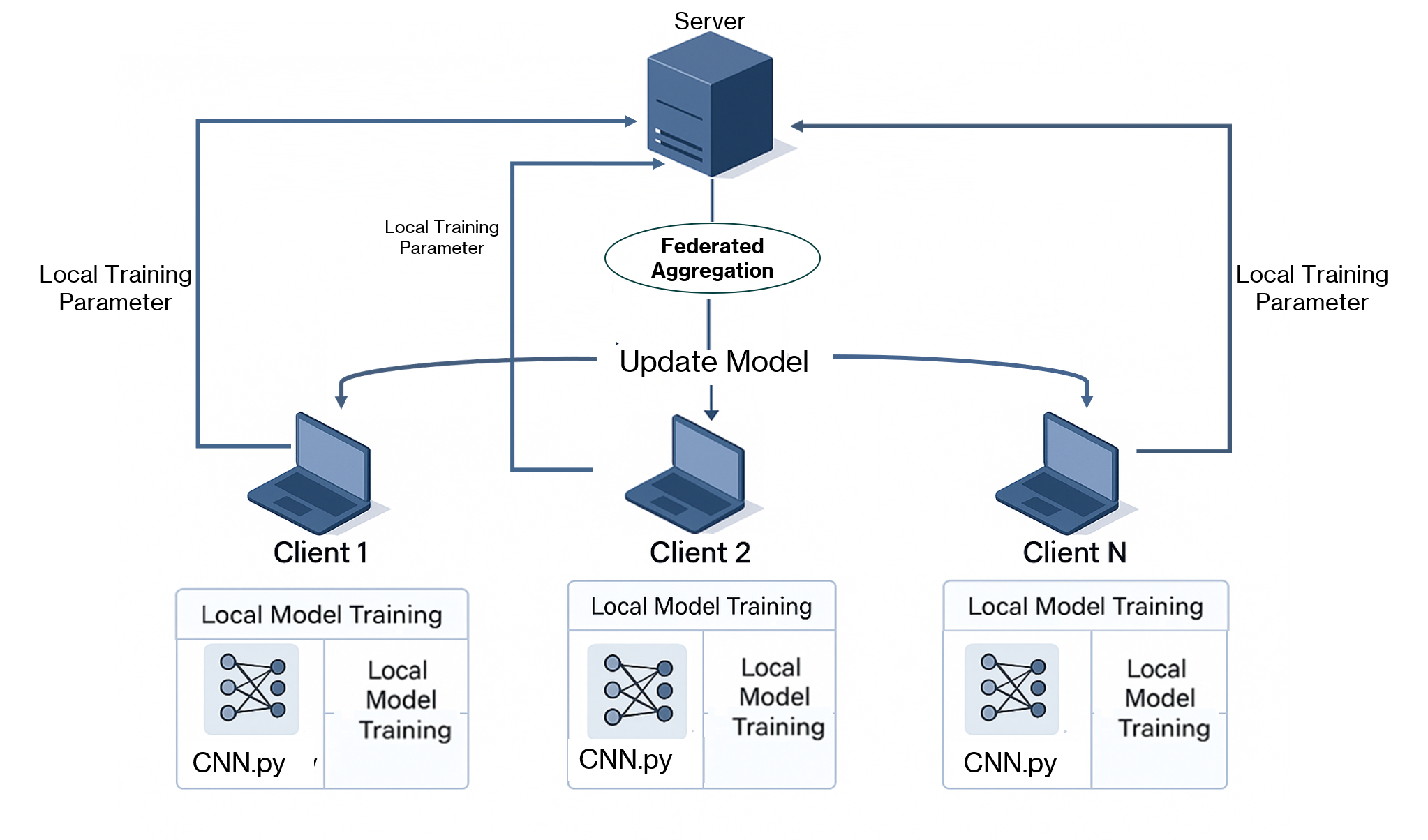


Figure 2: Model architecture

* Stochastic Gradient Descent (SGD) was used to optimize the model; weight changes were communicated to the server after local training over several epochs. Even on devices with low processing power, the CNN's simplicity made guaranteed that training remained effective.

# System Architecture

## Client System

* It is the responsibility of each client to load and preprocess its local dataset.
* Using local photos to train the CNN model.
* Making use of the test split to assess the model.
* Notifying the server of new model weights, accuracy, and loss.
* Complete privacy over the actual image content is ensured by isolating the local training process

from the server.

## Server System

* The server, which was constructed with the Flower (FLWR) federated learning framework, is in charge of:
  + Setting up the global model.
  + Sending the global model to customers.
  + Using a chosen method (such as FedAvg or FedAdam) to aggregate the updated weights after receiving them.
  + Overseeing several training sessions.
  + Using a "SaveBest" approach, the top-performing global model is saved.Because of the server's flexibility, the number of rounds, learning rates, and client count can all be changed via the command line. Additionally, it can withstand delays, dropouts, and inconsistent client contributions.

# Federated Optimization Strategies

* Throughout this study, several federated optimization techniques were assessed:
  + It is the responsibility of each client to load and preprocess its local data.
  + Using local photos to train the CNN model.
  + Making use of the test split to assess the model.
  + Notifying the server of new model weights, accuracy, and loss metrics.
  + Complete privacy over the actual image content is ensured by isolating the local training process from the server.

## FedAvg

* The baseline approach, which averages client updates based on weight. It operated well and with consistent accuracy of 88%.

A diagram of a program

AI-generated content may be incorrect.

Figure 3: Flow of FedAvg

## FedYogi

* With an accuracy of 89% in round 31, this adaptive method produced the best overall results. Throughout rounds, it remained stable and generalized.

A diagram of a program

AI-generated content may be incorrect.

Figure 4: Flow of FedYogi

## FedAdagrad

* This approach performed poorly, reaching a peak accuracy of only 24%, and had trouble with unstable gradients.

A diagram of a training program

AI-generated content may be incorrect.

Figure 5: Flow of FedAdagrad

## FedAdam

* LR = 0.1 NaN losses and instability caused an early failure.
* LR is equal to 0.001. performed admirably, training in 1,492 seconds and attaining 82% accuracy after 40 rounds.

## 

## A diagram of a computer system

Figure 6: Flow of FedAdam

# Results

* Using four client devices, we ran our tests across 40 communication rounds. The following significant findings were noted:

A graph showing the results of a loss

AI-generated content may be incorrect.

Figure 7: Loss of different clients

A graph showing the number of learning tests

AI-generated content may be incorrect.

Figure 8: Overall accuracy with different models

* By round 31, FedYogi had achieved the greatest test accuracy of 89%. FedAvg followed closely, convergently, at 88% by round 29. Over 1,492 seconds, FedAdam with a lower learning rate (0.001) demonstrated 82% accuracy and 29 misclassifications. The FedAdagrad and high-LR FedAdam models were judged inappropriate due to their low performance.

A green line graph with numbers

AI-generated content may be incorrect.

Figure 9: FedAdam accuracy with different Learning Rates

* Performance metrics like accuracy, loss trends, and model stability were visualized using Matplotlib, confirming that FedYogi offers the best trade-off between speed and performance. FedYogi provides the optimum trade-off between speed and performance, as demonstrated by the visualization of performance indicators such as accuracy, loss trends, and model stability using Matplotlib.

# Conclusion

* This research effectively illustrates that federated learning can be used to classify screenshots while maintaining user privacy. Data exposure is prevented by training on-device and only communicating model updates. When paired with FedYogi and FedAvg optimizers, our CNN-based model produced classification accuracies that were on par with centralized training models.

# Future Scope

* **Smart Gallery Integration:** Use the model to automatically group screenshots into folders in phone gallery apps.
* **Interactive Learning:** Let users fix incorrectly classified photos and help the model learn from comments.
* Investigate edge-based federated training with unpredictable connectivity and asynchronous clients for real-time training.
* More Classes: To increase usefulness, add more categories including social media, travel, education, and finance.
* Differential Privacy: Include strategies to obscure model updates and guarantee that client data cannot be reverse-engineered.

# References

1. H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017. [Online]. Available: <https://arxiv.org/abs/1602.05629>
2. K. Bonawitz et al. Towards Federated Learning at Scale: System Design. In SysML Confer- ence, 2019. [Online]. Available: <https://arxiv.org/abs/1902.01046>
3. Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Vir- ginia Smith. Federated Optimization in Heterogeneous Networks. In MLSys, 2020. [Online].
4. Available: <https://arxiv.org/abs/1812.06127>
5. Sashank J. Reddi et al. Adaptive Federated Optimization. In International Conference on Learning Representations (ICLR), 2021. [Online]. Available: [https://arxiv.org/abs/](https://arxiv.org/abs/2003.00295) [2003.00295](https://arxiv.org/abs/2003.00295)
6. Daniel Beutel, Taner Topal, Akhil Mathur et al. Flower: A Friendly Federated Learning Framework. [Online]. Available: <https://flower.dev/>
7. A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems (NeurIPS), 2012. [Online]. Available: [https://papers.nips.cc/paper\_files/paper/](https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html) [2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html](https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html)
8. J. Zhang, Y. Chen, and X. Wang. A Mobile Image Gallery App Based on AI and Privacy. In Mobile Software Engineering Conference, 2021. [Online]. Available: [https://ieeexplore.](https://ieeexplore.ieee.org/document/9449618) [ieee.org/document/9449618](https://ieeexplore.ieee.org/document/9449618)