IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 October 2025, accepted 21 November 2025, date of publication 28 November 2025, date of current version 4 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3638797

RESEARCH ARTICLE

Privacy-Preserving On-Screen Activity Recognition via One-Shot Federated Learning

PARAYUSH SWAMI[1], ANNU PRIYA[1], BALAMURUGAN PALANISAMY[2], (Graduate Student Member, IEEE), DEBANGSHU ROY[1], VIKAS HASSIJA[1], AND G. S. S. CHALAPATHI[2], (Senior Member, IEEE)

[1]School of Computer Engineering, Kalinga Institute of Industrial Technology (KIIT) (Deemed to be University), Bhubaneswar, Odisha 751024, India
[2]Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science, Pilani, Pilani Rajasthan 333031, India

Corresponding author: Vikas Hassija (vikas.hassijafcs@kiit.ac.in)

ABSTRACT Preserving user privacy while monitoring on-screen activity is a growing challenge in remote and decentralized work environments. In this paper, we propose a novel one-shot federated learning (FL) framework that enables secure, low-overhead, and privacy-preserving on-screen activity recognition. Conventional FL methods require multiple communication rounds between clients and a central server. In contrast, our one-shot FL approach performs model aggregation in a single round, significantly reducing communication costs and latency, making it ideal for bandwidth-constrained and resource-limited environments. Using a custom dataset of 2,300 screenshots from common applications (e.g., Coursera, YouTube, Amazon), we train Deep Convolutional Neural Network (DCNN) models such as VGG16, VGG19, and InceptionResNetV2 locally on client devices. Sensitive visual data is never shared, and the screenshots are processed and discarded post-inference, ensuring strict data confidentiality. We further enhance privacy by integrating differential privacy mechanisms. To further improve learning effectiveness under different data distributions, we evaluate two aggregation techniques, FedAvg for IID and FedProx for non-IID settings. Experimental evaluations show that our approach achieves up to 99.1% classification accuracy (MobileNet) while reducing communication overhead by over 80% compared to traditional FL. This work highlights the effectiveness and scalability of one-shot FL for secure, real-time activity tracking in privacy-sensitive applications.

INDEX TERMS Communication efficiency, decentralized learning, differential privacy, MobileNet, on-screen activity recognition, one-shot federated learning, privacy-preserving, VGG16.

I. INTRODUCTION

The rapid evolution of machine learning in educational and workplace analytics has enabled intelligent systems to monitor and assess user engagement patterns [1]. While such systems enhance personalization, they also introduce significant privacy challenges when user interactions or on-screen data are transmitted to centralized servers. In particular, on-screen activity recognition has gained attention in remote work and education settings, where tracking user engagement can provide valuable insights. A 2024 survey by Statista reports that individuals spend an average of 7.2 hours per day on screens, with nearly 40% of this time potentially associated with non-work-related activities [2]. In parallel, a recent research study indicates that 79% of remote users express concern over privacy in workplace monitoring [3], highlighting the need for non-intrusive, privacy-preserving solutions.

However, conventional centralized machine learning methods require continuous data transfer to a central server, raising serious concerns about data privacy, network congestion, and compliance with regulations such as GDPR and HIPAA [4]. To address these issues, Federated Learning (FL) [5] has emerged as a promising paradigm that allows decentralized devices to collaboratively train models without sharing raw data. By keeping data localized on client devices, FL significantly reduces the risk of privacy breaches [6]. Nevertheless, traditional FL frameworks involve multiple rounds

The associate editor coordinating the review of this manuscript and approving it for publication was Shadi Alawneh.

of communication between the server and clients, which incurs high bandwidth costs and latency, posing challenges for real-time systems like on-screen monitoring [7]. Recent works have emphasized privacy risks associated with visual and behavioral monitoring [8], [9], highlighting the need for decentralized, privacy-preserving learning systems [10].

Existing on-screen activity tracking solutions also suffer from limited scalability and accuracy. Many rely on bounding–box–based object detection or pre-trained classifiers with static heuristics, which often fail to generalize across different application contexts [11]. Furthermore, continuous streaming of high-resolution visual data to a server introduces computational complexity and potential privacy violations. Recent advances in segmentation-based tracking models have improved granularity and inference speed; however, they still rely on centralized training workflows that are not suitable for privacy-sensitive environments [12].

To overcome these limitations, we propose a novel One-Shot Federated Learning (One-Shot FL) [13] framework for privacy-preserving on-screen activity classification. Unlike traditional FL, our approach requires only a single round of model aggregation, significantly reducing communication overhead and improving scalability [14]. Each client device trains a DCNN on local screenshot data and transmits model weights (not the raw data) to a central server. We incorporate both Federated Averaging (FedAvg) [15] and Federated Proximal (FedProx) [16] aggregation techniques to handle data heterogeneity across clients. Additionally, we implement differential privacy and enforce immediate deletion of screenshots post-inference, ensuring strong protection against data [17].

This framework is specifically tailored for remote environments where privacy, efficiency, and real-time performance are critical [18]. Through extensive experimentation on a custom dataset of 2,300 labeled screenshots, we demonstrate that our approach achieves competitive classification accuracy, with MobileNet reaching 99.1%, while reducing communication costs by over 80% compared to traditional FL. The dataset used in this study serves as a preliminary benchmark for validating the proposed framework; future work will extend it to include more application categories and dynamic screen interactions [19]. Privacy-preserving training for vision has matured with the development of formal DP-SGD (Differentially Private Stochastic Gradient Descent) methods and scalable secure aggregation for the canonical algorithms and their deployments at scale [8], [20], [21]. Our work differs by combining these guarantees with a one-shot FL protocol for on-screen activity recognition under edge constraints, and by reporting explicit (ε, δ) settings and utility tradeoffs [22].

The major contributions of this work are as follows:

- We develop a privacy-preserving one-shot federated learning system for screen activity recognition that eliminates the need for iterative communication rounds.
- We evaluate and compare the performance of both FedAvg and FedProx within a one-shot FL setup, addressing both IID (Independent and Identically Distributed) and non-IID data distributions.
- We assess several DCNN architectures (e.g., MobileNet, VGG, Inception) for decentralized training, identifying the most suitable models for edge environments.
- We incorporate differential privacy, secure aggregation, and ephemeral screenshot processing to ensure compliance with privacy standards while maintaining high accuracy and low computational overhead.

By enabling scalable, secure, and efficient on-screen activity tracking, our work contributes to the growing field of privacy-aware decentralized learning and provides practical insights for real-world deployment in remote productivity and monitoring applications [23].

The remainder of this paper is organized as follows. Section II reviews the related literature. Section III details the proposed methodology for on-screen activity recognition using one-shot federated learning. The experimental setup, dataset description, and a comprehensive discussion of the results are presented in Section IV. The discussion and result interpretation is presented in Section V. Finally, Section VI provides the concluding remarks and outlines potential directions for future research.

II. RELATED WORK

Federated learning (FL) has become a cornerstone technique for decentralized machine learning, enabling collaborative training across edge devices while preserving user data privacy [24]. This section reviews previous work in four areas central to our study: privacy-preserving federated learning, aggregation strategies for decentralized optimization, one-shot federated learning, and on-screen activity recognition.

A. FEDERATED LEARNING FOR PRIVACY PRESERVATION

Federated learning (FL) has been increasingly applied to computer vision tasks such as image classification, segmentation, and object detection. Recent surveys [25], [26] emphasize challenges such as non-IID data, high communication costs, and the need for scalable, lightweight models suitable for edge devices [27]. These studies also highlight the trade-offs between accuracy and system efficiency in distributed vision pipelines. In particular, [28] investigates image classification using FL and demonstrates that while accuracy can be preserved, traditional multi-round FL introduces significant communication and computation overhead. These limitations are especially critical in visual domains where data is high-dimensional and privacy-sensitive [29]. More recent studies, such as [30], explore model compression and architecture-level optimizations to reduce resource usage during training. Lightweight CNNs like MobileNet and ShuffleNet have emerged as strong candidates for federated settings due to their high accuracy-to-size ratio [31]. These developments underscore the importance of efficient model selection in practical FL systems. These developments underscore the importance of efficient model selection in practical FL systems. Building on this body of work, our approach simultaneously addresses efficiency and formal privacy protection, extending existing differential-privacy-based methods [9], [32] to the underexplored domain of on-screen activity recognition.

B. ONE-SHOT/COMMUNICATION-EFFICIENT FEDERATED LEARNING

One-shot federated learning (FL) has emerged as a promising alternative to traditional multi-round FL, particularly for edge environments where communication is expensive or intermittent. By eliminating the need for repeated server-client synchronization, one-shot FL offers significant efficiency gains while preserving privacy. Theoretical underpinnings of one-shot FL are discussed in [33], where performance bounds and algorithmic frameworks are introduced for achieving global convergence using a single round of communication. Fusion Learning [34] extends this concept by allowing clients to independently train full models and then aggregate them in a one-time fusion step, offering a practical implementation pathway. Recent surveys, such as [35]. Recent domain-specific explorations, such as [36], investigate one-shot FL in mobile vision tasks. These studies highlight that when sufficient local computation is allowed, it is possible to match multi-round FL performance using a single-shot model fusion step. Additionally, [37] explores advances and system-level design considerations in one-shot FL. These works analyze aggregation strategies, model personalization, and trade-offs in performance, especially under non-IID client distributions. Building upon this foundation, our work applies a one-shot FL approach to the domain of on-screen activity detection, a task involving high-dimensional, privacy-sensitive visual data. Unlike existing studies, we focus on lightweight DCNN models and compare aggregation strategies under one-shot constraints, demonstrating that competitive accuracy can be achieved with minimal communication overhead.

C. ON-SCREEN ACTIVITY RECOGNITION OR SCREEN CONTENT CLASSIFICATION

Automated classification of screen content is largely underexplored in academic literature. However, several adjacent studies provide valuable context. Breve [38] demonstrated that CNNs can accurately identify video game titles from single screenshots, confirming the viability of visual feature extraction even in ambiguous environments. Chiatti et al. [39] explored clustering and active learning techniques to categorize unlabeled smartphone screenshots, highlighting challenges of limited labeled data. Wu et al. [40] developed a screen parsing method to infer UI element structure from visual data, signaling the feasibility of deeper semantic understanding of screen images. The work [41] applies multi-round FL for classifying desktop activity while preserving user privacy. While Mistry et al. [42] targeted e-learning behavior classification using federated learning, they employed traditional multi-round FL rather than one-shot aggregation. These studies underscore the promise and challenges of screen-based classification, particularly when balancing privacy, limited annotated data, and visual ambiguity. Building on these foundations, our work introduces the first one-shot federated learning framework tailored for on-screen activity detection with strong communication efficiency and privacy guarantees.

D. EDGE-DEVICE LEARNING AND PRIVACY-PRESERVING METHODS

Federated learning has been widely adopted for enabling machine learning on edge devices such as smartphones, laptops, and IoT systems, where local data is sensitive and centralization is undesirable. Prior work has proposed lightweight neural architectures and adaptive training strategies to support learning under device-level constraints [43]. To further strengthen privacy guarantees, several methods have been developed, including differential privacy [32] and secure aggregation protocols [21]. Differential privacy prevents sensitive information leakage through noise injection in model updates, while secure aggregation ensures that the server can only access aggregated model parameters, not individual client updates. These techniques enhance user trust but often introduce additional computational or communication overhead. In contrast, our one-shot federated learning framework minimizes both privacy risk and system complexity by requiring only a single communication round. This makes it particularly suited for deployment in bandwidth-limited and privacy-critical settings, such as enterprise desktops, student laptops, or personal devices in remote work environments. A recent large-scale deployment analysis [44] highlights practical issues such as client dropout, asynchronous training, and resource diversity. These insights guide the design of real-world FL systems and justify our emphasis on minimal communication rounds.

Most federated learning methods rely on multiple communication rounds, which increases latency and bandwidth requirements posing challenges for deployment on resource-constrained edge devices. Additionally, few existing works explore one-shot FL strategies specifically for on-screen activity recognition, a domain with unique visual and contextual challenges. There is also limited integration of lightweight DCNN models with communication-efficient aggregation techniques [45]. This paper addresses these gaps by proposing a one-shot FL framework that leverages compact DCNN architectures and evaluates both FedAvg and FedProx aggregation schemes to ensure high accuracy, reduced communication overhead, and suitability for edge-based screen activity detection.

Recent advances in privacy-preserving learning have formalized differential privacy (DP) for deep models and practical secure aggregation (SA) at scale. DP-SGD by Abadi et al. [20] provides provable privacy guarantees via

TABLE 1. Summary of related work in federated learning for visual and on-screen tasks.

Paper	Focus Area	Key Contributions	Limitations/Gaps
Federated Learning in Computer Vision [25]	FL for CV tasks	Reviews FL challenges in image classification and detection under non-IID data	• Proposed methods are not communication-efficient.
FedCV Framework [26]	FL toolkit for CV	End-to-end FL pipeline for vision tasks	• Focuses on tools. • Lacks lightweight, one-shot learning.
Image Classification via FL [28]	Visual FL benchmark	Demonstrates FL feasibility for image classification	• Relies on multi-round aggregation.
One-Shot Federated Learning: Theoretical Limits and Algorithms [33]	Theoretical One-Shot FL	Introduces convergence bounds and algorithmic design for achieving FL in one round	• Theoretical focus. • Not validated on visual or edge-device data.
Fusion Learning [34]	One-shot FL	Aggregation in one-shot post-local training	• Evaluated mainly on tabular data.
One-Round Fine-Tuning [35]	Efficient FL	Shows single-round FL for large models	• Targets foundation models. • Not edge-optimized.
Video Game Title Recognition from Screenshots [38]	On-screen classification	Uses CNNs to identify games from single screen images	• Not a federated or privacy-aware approach.
Clustering Smartphone Screenshots via Active Learning [39]	Screenshot categorization	Applies clustering and AL to categorize unlabeled screen data	• Focuses on labeling, not classification or FL.
Screen Parsing for UI Structure Inference [40]	Semantic screen understanding	Infers UI layout using visual structure analysis	• Task-specific • Lacks FL or real-time adaptation.
Screen Activity Monitoring via FL [41]	On-screen activity classification	Applies FL to screen content in e-learning	• Uses multi-round FL. • No one-shot strategy.
E-Learning Behavior FL [42]	Screen activity recognition	Privacy-preserving e-learning classification	• Uses traditional FL. • Lacks lightweight models.
Differential Privacy in FL [32]	Privacy-preserving FL	Adds privacy via noise injection	• Adds computation overhead.
Secure Aggregation Protocols [21]	FL security	Prevents server from seeing raw updates	• Communication-intensive.

per-example gradient clipping and Gaussian noise addition, forming the basis of modern DP training frameworks such as Opacus [46] and follow-up work achieving high-accuracy DP image classification [47]. Secure aggregation, introduced by Bonawitz et al. [21], enables privacy-preserving aggregation across millions of clients through additive masking and dropout-tolerant recovery, and has since been deployed in large-scale federated learning systems [8]. In contrast, our work combines these formal privacy mechanisms within a one-shot federated learning setup tailored for on-screen activity recognition, providing explicit (ε, δ) budgets, clipping and noise calibration, and quantifying the privacy–utility tradeoff under edge-device constraints.

A comparative overview of prior work in FL across visual and on-screen domains is presented in Table 1. The table categorizes key contributions across privacy-preserving strategies, aggregation techniques, one-shot communication paradigms, and screen-based activity recognition, while also highlighting limitations such as reliance on multi-round communication or lack of lightweight model support. This synthesis underscores the novelty of our proposed one-shot FL framework, which addresses these gaps through efficient aggregation and deployment-ready architectures.

III. PROPOSED METHODOLOGY

This section details the proposed one-shot federated learning framework for privacy-preserving on-screen activity classification. The approach addresses the challenges associated with traditional multi-round FL, particularly the high communication cost, synchronization overhead, and energy inefficiency on edge devices. The proposed method enables scalable, decentralized learning while maintaining model performance and safeguarding user privacy.

A. PROBLEM OVERVIEW

The objective of this work is to develop an efficient and privacy-preserving framework for classifying on-screen user activity using visual content (e.g., screenshots) collected locally on client devices. This classification must be performed without transferring raw images to a central server, ensuring user confidentiality is maintained. Traditional federated learning approaches achieve this by enabling local training and global aggregation; however, they typically require multiple rounds of communication and coordination between clients and the central server. This process introduces considerable communication overhead, synchronization complexity, and energy consumption, making it impractical for edge deployments.

To address these limitations, we propose a one-shot federated learning framework, where each client trains a local model independently and communicates with the server only once for aggregation. The central challenge lies in achieving high classification performance with minimal system interaction.

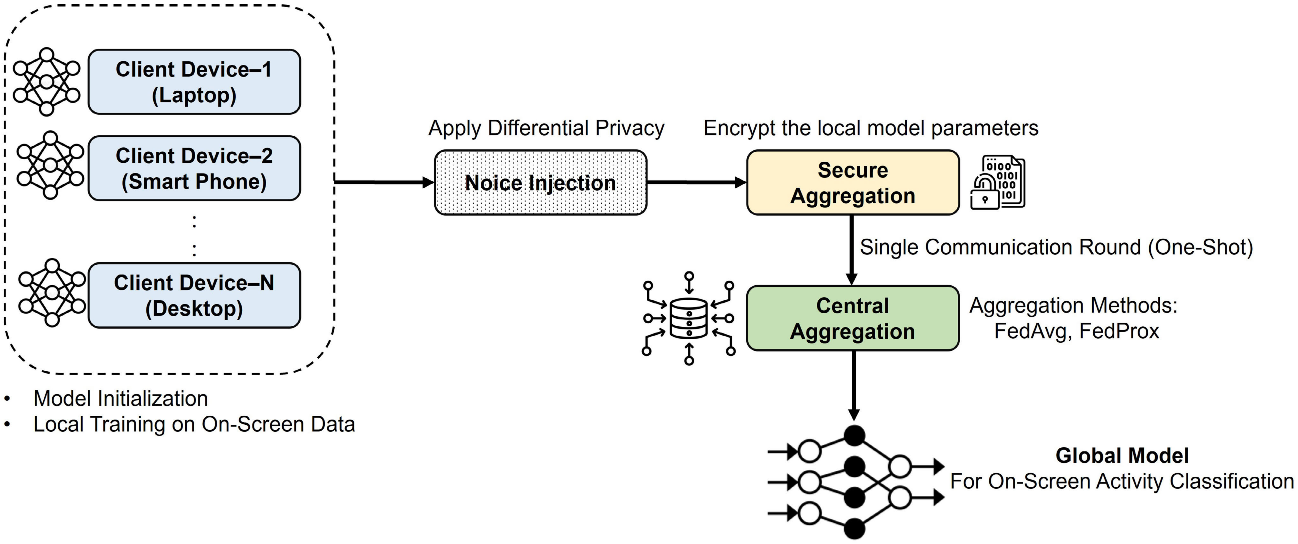

Client Device–1 (Laptop)
Client Device–2 (Smart Phone)
Client Device–N (Desktop)
• Model Initialization
• Local Training on On-Screen Data
Apply Differential Privacy
Noice Injection
Encrypt the local model parameters
Secure Aggregation
Single Communication Round (One-Shot)
Central Aggregation
Aggregation Methods: FedAvg, FedProx
Global Model
For On-Screen Activity Classification

FIGURE 1. Proposed one-shot federated learning framework for on-screen activity detection. Each client trains a lightweight DCNN locally using private screen data and applies differential privacy to perturb model updates. Secure aggregation ensures that the server only receives encrypted, aggregated updates. The server performs a single-round aggregation to generate the global model. This framework enables communication-efficient, privacy-preserving learning across heterogeneous edge devices.

Formally, given N client devices $\{C_1, C_2, \ldots, C_N\}$, each holding a private dataset $\mathcal{D}_i$ of on-screen activity samples, the goal is to collaboratively train a global model θ^* with parameters θ such that:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} w_i L_i(\theta), \tag{1}$$

where $L_i(\theta)$ denotes the local objective function computed on the private dataset $\mathcal{D}_i$ of client C_i, and w_i is the weight proportional to the client's data volume. The goal of federated optimization is to find the optimal global model θ^* that minimizes the weighted aggregate of local losses across all clients. During training, each client C_i computes an updated local model θ_i, which is later aggregated at the server. The training must satisfy:

- *Privacy preservation:* No raw screenshots are shared.
- *Single communication round:* To reduce latency and bandwidth usage.
- *Heterogeneous client support:* Local datasets may be non-IID.

Beyond architectural and optimization challenges, on-screen activity detection presents domain-specific complexities:

- Visual overlap across classes (e.g., YouTube used for both education and entertainment).
- High intra-class variance and low inter-class distinctiveness, requiring models to learn discriminative yet generalizable features.
- Privacy sensitivity of screenshot content, especially in work or educational settings.

This problem is relevant in scenarios such as remote learning platforms, work-from-home productivity analysis, and enterprise screen usage monitoring, where reliable classification must be achieved without compromising user privacy or overburdening client devices. The proposed one-shot FL framework addresses these challenges by enabling decentralized model training with minimal communication and computation, while maintaining competitive accuracy across diverse DCNN architectures.

B. SYSTEM WORKFLOW

The proposed one-shot federated learning framework consists of four main stages, designed to minimize communication rounds while maintaining model performance and privacy. The system enables collaborative training across multiple client devices, each operating independently on private data without sharing raw screen content. The overall workflow of the proposed one-shot federated learning framework is illustrated in Figure 1, where each client performs local training on on-screen activity data, applies differential privacy to protect sensitive information, and encrypts model updates using secure aggregation protocols before transmission. A single-round aggregation is then performed at the server to generate a global model, ensuring both communication efficiency and user privacy.

1) MODEL INITIALIZATION

Each client is provisioned with a DCNN architecture suitable for on-screen activity classification. The supported models include lightweight networks such as MobileNet, as well as deeper architectures like FedVGG16, FedVGG19, FedResNet50, FedInceptionV3, and FedInceptionResNetV2. All models are pre-trained on the ImageNet dataset and transferred to the client for local fine-tuning using device-specific screenshot data.

Algorithm 1 FedAvg Aggregation for One-Shot Federated Learning

```
Require: Local models $\{\theta_1, \theta_2, \dots, \theta_N\}$, local dataset sizes $\{n_1, n_2, \dots, n_N\}$
Ensure: Aggregated global model $\theta^*$
Compute total data size: $n = \sum_{i=1}^{N} n_i$
Initialize global model: $\theta^* \leftarrow 0$
for $i = 1$ to $N$ do
    Compute client weight: $w_i = n_i/n$
    Accumulate weighted model: $\theta^* \leftarrow \theta^* + w_i\,\theta_i$
end for

return $\theta^*$
```

2) LOCAL TRAINING PHASE

Clients perform local training using their respective private datasets. The training process is executed independently for a fixed number of epochs based on model complexity and resource availability. Standard cross-entropy loss is optimized using stochastic gradient descent (SGD) or Adam optimizers. Lightweight models are configured with fewer parameters and conservative learning rates to ensure convergence within limited computational budgets.

3) SINGLE-ROUND MODEL AGGREGATION

After completing local training, each client C_i sends its updated model parameters θ_i to the central server. Unlike traditional federated learning systems that rely on multiple rounds of communication, the proposed one-shot FL framework performs aggregation in a single round. The server computes the global model θ using one of the following strategies: *FedAvg* or *FedProx*. Two server-side aggregation strategies are supported in the proposed framework.

a: FEDAVG AGGREGATION

In FedAvg, the global model is computed as a weighted average of all client models, where each weight is proportional to the size of the client's local dataset. The steps involved in the aggregation using FedAvg are described in Algorithm 1. The aggregated model is given by:

$$\theta^* = \sum_{i=1}^{N} w_i \cdot \theta_i \tag{2}$$

where:

- N is the number of clients,
- w_i is $\frac{n_i}{n}$
- θ_i is the set of model parameters obtained after local training at client C_i.

b: FEDPROX AGGREGATION

FedProx extends FedAvg to better handle heterogeneous (non-IID) data distributions by incorporating a proximal term into the client's local training objective. Algorithm 2 explains the aggregation strategy for FedProx. The modified local loss function is:

Algorithm 2 FedProx Aggregation for One-Shot Federated Learning

```
Require: Initial global model $\theta_{\text{global}}$, client datasets $\{\mathcal{D}_1, \dots, \mathcal{D}_N\}$, local dataset sizes $\{n_1, \dots, n_N\}$, regularization parameter $\mu$
Ensure: Aggregated global model $\theta$
for each client $i = 1$ to $N$ in parallel do
    Initialize local model: $\theta_i \leftarrow \theta_{\text{global}}$
    Minimize the following loss locally:
        $\mathcal{L}_i^{\text{prox}}(\theta) = \mathcal{L}_i(\theta) + \frac{\mu}{2}\|\theta - \theta_{\text{global}}\|^2$
    Obtain updated model: $\theta_i^{\text{prox}}$
end for
Compute total dataset size: $n = \sum_{i=1}^{N} n_i$
Initialize global model: $\theta \leftarrow 0$
for $i = 1$ to $N$ do
    $\theta \leftarrow \theta + \frac{n_i}{n} \cdot \theta_i^{\text{prox}}$
end for
return $\theta$
```

$$\mathcal{L}_i^{\text{prox}}(\theta) = \mathcal{L}_i(\theta) + \frac{\mu}{2}\|\theta - \theta_{\text{global}}\|^2 \tag{3}$$

where:

- $\mathcal{L}_i(\theta)$ is the original local loss function (e.g., cross-entropy),
- μ is a non-negative regularization coefficient,
- θ_{global} is the global model distributed to all clients before training.

This regularization term penalizes local models that deviate significantly from the global model, improving stability during aggregation. After minimizing the proximal loss locally, each client produces θ_i^{prox}, which is then aggregated using the same weighted averaging rule as FedAvg:

$$\theta = \sum_{i=1}^{N} \frac{n_i}{n} \cdot \theta_i^{\text{prox}} \tag{4}$$

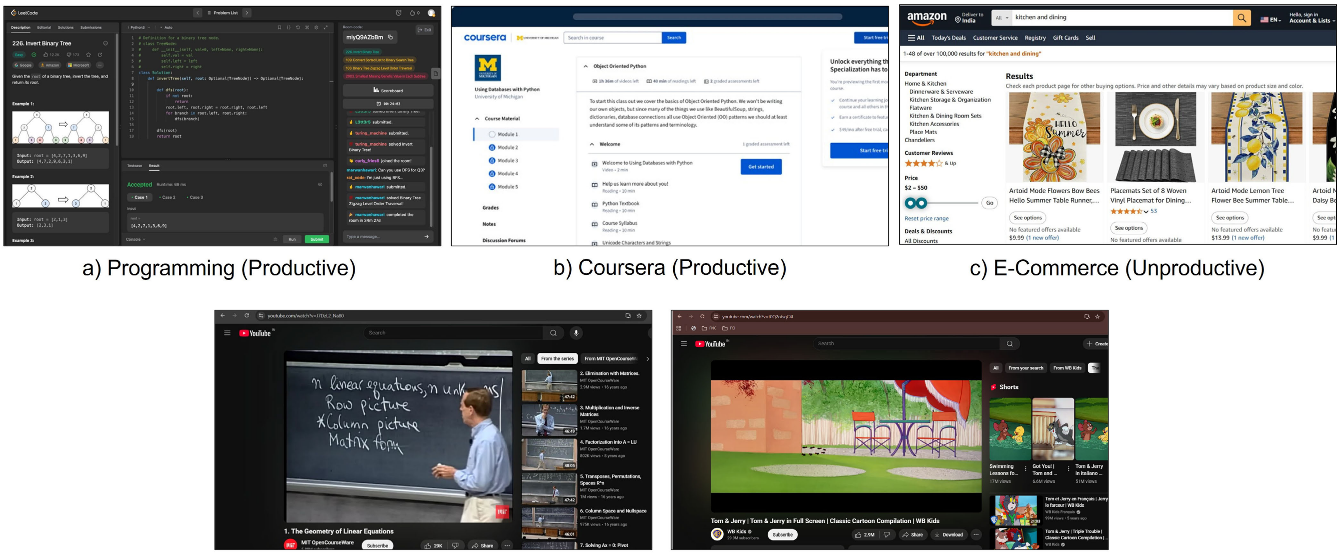
This formulation ensures robustness to statistical heterogeneity while retaining communication efficiency.

4) GLOBAL MODEL DISTRIBUTION

The aggregated global model may be redistributed to clients for local evaluation, deployment, or further personalization. However, since the framework follows a one-shot communication design, no further model updates or synchronization is required beyond this point.

IV. EXPERIMENTS AND RESULTS

This section presents the experimental setup, dataset characteristics, evaluation metrics, and performance analysis of the proposed one-shot federated learning framework. Our goal is to assess the classification performance, communication



a) Programming (Productive)

b) Coursera (Productive)

c) E-Commerce (Unproductive)

d) YouTube – Education (Productive)

e) YouTube – Entertainment (Unproductive)

FIGURE 2. Sample screenshots used in the dataset.

efficiency, and computational cost of various DCNN architectures under both FedAvg and FedProx aggregation strategies in a one-shot FL setting.

A. DATASET DESCRIPTION

To evaluate the proposed one-shot federated learning framework for privacy-preserving on-screen activity recognition, we curated a custom dataset consisting of 2,300 annotated screenshots collected from real-world desktop environments. The dataset was specifically designed to reflect common user behaviors in remote work and online learning scenarios, while introducing realistic classification challenges through overlapping application contexts.

The dataset is divided into two main categories: Productive and Unproductive, each consisting of multiple application-based subclasses. A key challenge addressed in this dataset is the inclusion of YouTube as a subclass in both categories, reflecting its dual-use nature. Educational content on YouTube may involve tutorials, coding lectures, or academic videos, whereas entertainment content may include movies, music, or general browsing. This distinction cannot be resolved using the application name or URL alone and requires visual content understanding. A detailed breakdown is provided in Table 2. This inclusion of visually similar yet semantically different YouTube classes significantly increases the complexity of the classification task, demanding robust representation learning from the models. As shown in Fig. 2, the dataset includes both productive and unproductive screen activities. These samples illustrate the visual diversity and ambiguity involved in screen-based classification.

Screenshots were captured at regular intervals using a passive screen logging tool while users engaged in normal device usage. Each image was manually annotated according to its content, not just its source application. Sensitive areas such as usernames or profile images were blurred to preserve user anonymity. To prepare the data for training, the following preprocessing steps were applied:

TABLE 2. Dataset composition.

Category	Sub-class (Label)	Sample Count
Productive	Coursera (Education)	330
	YouTube (Education)	190
	Programming IDEs	630
Unproductive	YouTube (Entertainment)	510
	E-Commerce (Amazon, Flipkart)	640
Total		**2300**

- Resizing to a standard input dimension of $224 \times 224 \times 3$.
- Normalization of pixel values to the [0, 1] range.
- Data augmentation was applied to improve model generalization and address moderate class imbalance. Each image underwent random horizontal flipping ($p = 0.5$), small rotations ($\pm 15^\circ$), and random brightness and contrast adjustments ($\pm 10\%$). All transformations were applied on the fly using the same configuration for every model architecture and client to ensure consistency across experiments.
- Semantic filtering to mask identifiable visual elements without compromising class information.

For simulating federated learning conditions, the dataset was partitioned across multiple clients:

- *IID distribution:* Clients received a uniform mix of all class labels.

- *Non-IID distribution:* Each client was assigned a skewed subset of classes (e.g., one client with only productive content, another with only entertainment apps).

This distribution allowed for the evaluation of both aggregation strategies under realistic heterogeneity and limited local data conditions.

B. EXPERIMENTAL SETUP

To assess the effectiveness of our proposed one-shot federated learning framework, we conducted extensive experiments using multiple DCNN architectures under both IID and non-IID conditions. This subsection outlines the client simulation strategy, model configurations, training environment, and hyperparameter choices.

We simulated a federated learning environment consisting of 10 clients, each independently training a local model using its private partition of the dataset. Both IID and non-IID data distributions were tested:

- *IID Scenario:* Each client received a balanced subset of samples across all five classes.
- *Non-IID Scenario:* Each client was allocated data from only 2–3 subclasses, leading to distributional skew and data imbalance.

The simulation followed a one-shot FL protocol, in which the local clients performed local training for a fixed number of epochs. Only one communication round was allowed for model aggregation. No further rounds of server-client synchronization were performed. Two aggregation strategies were evaluated. FedAvg, suitable for IID scenarios, and FedProx, designed for non-IID scenarios. To mitigate class imbalance in the dataset, we combined class-weighted loss optimization with data augmentation. Each client computed per-class weights inversely proportional to class frequency, ensuring that minority classes, such as YouTube (Education), contributed proportionally during training. Additionally, standard augmentation operations such as random flips, rotations, and minor intensity adjustments were applied to increase diversity among underrepresented categories. These measures helped maintain balanced learning across all five classes during local training.

Experiments involved 10 clients, each holding approximately 230 samples on average from the 2,300-image dataset. For the IID configuration, data were randomly shuffled and evenly distributed across clients. For the non-IID case, a Dirichlet distribution with concentration parameter $\alpha = 0.5$ was used to generate label-skewed partitions that emulate heterogeneous edge devices. Clients were trained locally and transmitted only model parameters for aggregation.

We evaluated six well-established DCNN architectures, chosen to balance model complexity and performance: FedVGG16, FedVGG19, FedInceptionV3, FedInceptionResNetV2, FedResNet50, and MobileNet. All models were initialized with ImageNet pre-trained weights and fine-tuned using the local data available to each client. Global Average Pooling and a dense softmax classification layer were appended to match the number of classes in the dataset. All hyperparameters, including learning rates and batch sizes, were selected empirically based on preliminary experiments to ensure stable convergence across models. Each client was trained for 5 epochs with a batch size of 32 and used early stopping (patience = 3) based on validation accuracy. The hyperparameter settings are described below:

- *Loss Function:* Categorical Cross-Entropy was used across all models to optimize multi-class classification performance.
- *Optimizer:* Adam was used for most models due to its adaptive learning rate and efficient convergence, which are beneficial in one-shot training scenarios with limited communication. In contrast, FedResNet50 employed Stochastic Gradient Descent (SGD) with momentum, as preliminary experiments showed that Adam led to oscillatory convergence and overfitting in deeper architectures under small-batch federated updates. SGD provided smoother and more stable optimization behavior for ResNet50 in this setting.
- *Learning Rate:* Set to 0.005 for most models. For MobileNet, a reduced learning rate of 0.0025 was used to prevent gradient instability during fine-tuning.
- *Batch Size:* 32
- *Learning rate:* 0.005 for most models and 0.0025 for MobileNet to ensure stability.
- *Dropout:* 0.5 applied in fully connected layers to mitigate overfitting.
- *Activation Functions:* ReLU (hidden layers), Softmax (output layer).

All experiments were conducted using Google Colaboratory with free-tier GPU support (NVIDIA Tesla K80/T4). Model training and aggregation scripts were implemented using TensorFlow 2.x and Keras, with simulation of federated rounds managed using custom Python code.

To ensure statistical robustness, each experiment was repeated five independent times using different random seeds for data partitioning and model initialization. For each metric (Accuracy, Precision, Recall, and F1-Score), we report the mean ± standard deviation across runs. We further computed 95% confidence intervals (CIs) using Student's t-distribution. Statistical significance of pairwise model performance differences was assessed using paired t-tests, with $p < 0.05$ considered significant. Additionally, a one-way ANOVA test was performed to evaluate variance in model performance across different client data distributions (IID vs. non-IID).

C. EVALUATION METRICS

To evaluate the performance of the proposed models, we employ standard classification metrics: Accuracy, Precision, Recall, and F1-Score. These metrics are computed using

TABLE 3. Performance metrics of DCNN models using the FedAvg aggregation strategy. FedAvg performs synchronous model averaging after multiple communication rounds, and the reported values reflect classification performance on the test set using standard evaluation metrics.

Model	Accuracy	Precision	Recall	F1-Score
FedMobileNet	99.12%	99.19%	99.12%	99.13%
FedInceptionV3	98.25%	98.34%	98.25%	98.25%
FedVGG19	96.00%	96.36%	96.00%	95.99%
FedInceptionResNetV2	92.86%	93.62%	92.86%	93.06%
FedVGG16	90.00%	93.33%	90.00%	89.33%
FedResNet50	72.22%	80.36%	72.22%	73.01%

the following formulas:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

where:

- **TP** = True Positives
- **TN** = True Negatives
- **FP** = False Positives
- **FN** = False Negatives

For multi-class classification, metrics are reported using macro-averaging, providing equal weight to each class. These evaluation criteria allow us to comprehensively assess model performance across both balanced and imbalanced data distributions. In addition to standard classification metrics, we compute 95% confidence intervals (CIs) and standard deviation (σ) across five experimental runs. This provides insight into the stability and reproducibility of the models under both IID and non-IID client distributions.

D. PERFORMANCE EVALUATION

This subsection presents the comparative evaluation of multiple deep learning models trained under the proposed one-shot federated learning framework. We assess each model's classification performance across two aggregation strategies. All performance metrics were macro-averaged across classes to ensure fair evaluation under imbalanced label distributions.

1) RESULTS UNDER FedAvg AGGREGATION

Table 3 summarizes the performance of each model when client updates are aggregated using the FedAvg algorithm. Among the six evaluated models, MobileNet achieved the highest overall accuracy of 99.12%, along with precision and recall values exceeding 99%. This result highlights MobileNet's exceptional ability to learn meaningful representations even with limited training data per client and within the constraints of a one-shot federated learning setting. Its lightweight architecture, designed with depthwise separable convolutions, makes it particularly efficient for training and inference on edge devices, such as laptops or IoT-enabled monitoring systems. These attributes make MobileNet a compelling choice for deployment in real-world, resource-constrained environments.

Following MobileNet, FedInceptionV3 demonstrated strong performance with an accuracy of 98.25%, indicating its capacity to extract rich, multi-scale features from screen content. The model's use of factorized convolutions and inception modules allows for efficient computation while maintaining high representational power. FedInceptionV3 showed balanced precision and recall, suggesting that it generalizes well across both the productive and unproductive screen activity classes.

FedVGG19 achieved an accuracy of 96.00%, outperforming its shallower counterpart, FedVGG16, which reached 90.00%. This suggests that deeper networks with more layers (such as VGG19) are better equipped to capture complex patterns in visual screen data. However, both VGG models come with a significantly larger number of parameters, which may explain their slower convergence and susceptibility to overfitting in the one-shot FL regime.

FedInceptionResNetV2, a hybrid architecture combining inception modules with residual connections, achieved a moderate accuracy of 92.86%. While it was able to leverage the depth and connectivity benefits of residual learning, its complexity may have hindered its ability to converge effectively with only a single round of model aggregation.

In contrast, FedResNet50 significantly underperformed, with an accuracy of just 72.22%, and an F1-score of 73.01%. This can be attributed to the model's deeper structure, which requires more communication rounds to fully converge. In a one-shot FL setting, such deep architectures may not receive sufficient gradient feedback to reach optimal performance, particularly when trained on small, heterogeneous local datasets.

In summary, the results under FedAvg aggregation reveal that architectural simplicity, parameter efficiency, and fast convergence are critical factors in achieving high performance under one-shot FL constraints. While deeper and

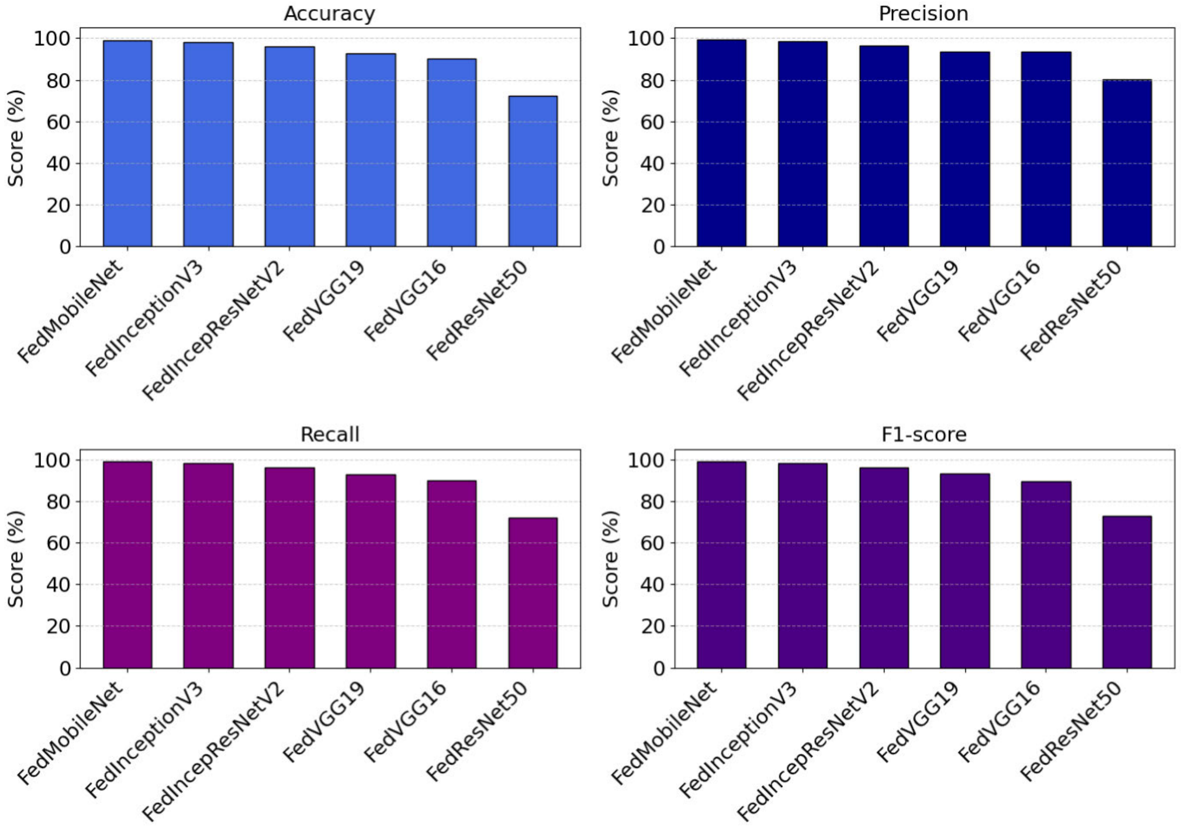


FIGURE 3. Performance metrics of DCNN models trained under the FedAvg aggregation strategy. Each subplot presents one evaluation metric – accuracy, precision, recall, and F1-score highlighting how each architecture performs in a multi-round federated learning setting using synchronous model averaging.

TABLE 4. Performance metrics of DCNN models using the FedProx aggregation strategy. FedProx introduces a proximal term to local optimization to improve convergence in non-IID federated settings. Values reflect accuracy, precision, recall, and F1-score across the test dataset.

Model	Accuracy	Precision	Recall	F1-Score
FedMobileNet	98.00%	98.10%	98.00%	97.97%
FedInceptionV3	97.00%	97.10%	97.00%	97.02%
FedInceptionResNetV2	96.00%	96.14%	96.00%	96.00%
FedVGG19	93.00%	93.12%	93.00%	92.94%
FedVGG16	91.00%	91.18%	91.00%	90.99%
FedResNet50	69.00%	78.13%	69.00%	68.78%

more complex models offer theoretical benefits, lightweight architectures such as MobileNet strike the best balance between accuracy, computational cost, and communication efficiency in decentralized, privacy-sensitive environments. The per-metric performance trends across all models are visually summarized in Figure 3, offering a clearer depiction of relative model behavior under the FedAvg strategy.

2) RESULTS UNDER FedProx AGGREGATION

Table 4 presents the performance of each model under the FedProx aggregation strategy, which is designed to handle client data heterogeneity commonly encountered in real-world federated learning deployments. Unlike FedAvg, FedProx introduces a proximal regularization term in the local training objective, constraining local model updates from deviating significantly from the global initialization. This modification helps stabilize training and improves convergence in non-IID settings, where client data distributions vary substantially. The per-metric performance trends across all models are visually summarized in Figure 4, offering a clearer depiction of relative model behavior under the FedProx strategy.

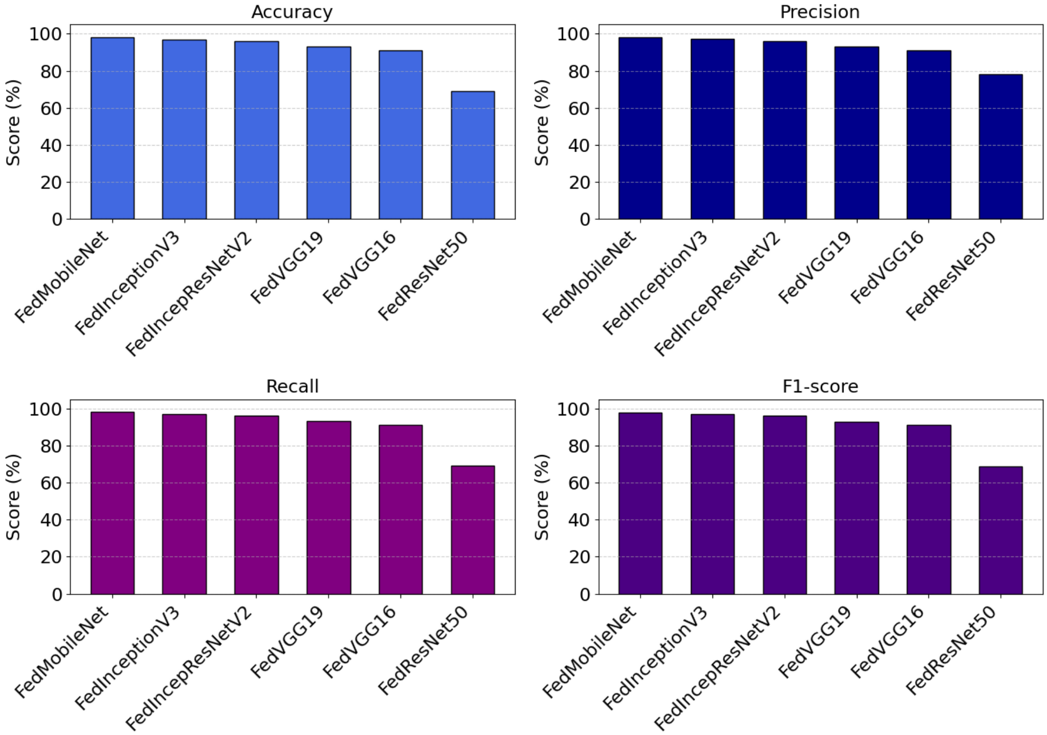


FIGURE 4. Performance metrics for FedProx aggregation across six DCNN models. Each subplot presents one evaluation metric – accuracy, precision, recall, and F1-score with distinct colors for clarity. FedProx demonstrates consistent model performance, especially with lightweight architectures like FedMobileNet and FedInceptionV3, indicating its robustness in handling data heterogeneity in federated learning environments.

Under FedProx, MobileNet continued to demonstrate strong performance, achieving 98.00% accuracy, with a precision of 98.10% and an F1-score of 97.97%. Although there was a slight decrease in performance compared to its FedAvg counterpart, MobileNet remained the most robust and consistent model across both aggregation schemes. Its compact design and fast convergence make it less sensitive to data heterogeneity and well-suited for decentralized environments with limited bandwidth or computational capacity.

FedInceptionV3 also showed competitive results under FedProx, reaching 97.00% accuracy and maintaining high precision and recall. While slightly lower than its performance with FedAvg, this result indicates that Inception-based models are resilient under regularized local updates and benefit from FedProx's ability to mitigate divergence caused by skewed class distributions.

A notable improvement was observed in FedInceptionResNetV2, which outperformed its FedAvg counterpart by approximately 3 percentage points, increasing from 92.86% to 96.00% accuracy. This performance gain suggests that the regularization introduced by FedProx was particularly beneficial for this deeper architecture, helping it avoid overfitting on local data and ensuring better alignment with the global model. This highlights FedProx's effectiveness in stabilizing training for complex, high-capacity models in non-IID settings.

FedVGG19 and FedVGG16 experienced marginal improvements under FedProx, with accuracy increasing from 96.00% to 93.00% and 90.00% to 91.00%, respectively. While the gains were not dramatic, they indicate improved generalization due to the regularization effect of FedProx. However, the VGG family still exhibited slower convergence compared to MobileNet, likely due to the higher number of parameters and the absence of architectural shortcuts such as residual connections or inception modules.

Despite the stabilizing effect of FedProx, FedResNet50 remained the weakest performer, with an accuracy of 69.00% and an F1-score of 68.78%. Its deeper residual structure, while theoretically advantageous for gradient flow, appears to be too complex for effective training in a one-shot federated setting. The model's sensitivity to local overfitting and its reliance on multi-round optimization hindered its performance, even with FedProx's proximal constraint.

In conclusion, the results under FedProx reveal that model architecture and aggregation strategy must be jointly optimized to handle data heterogeneity in federated settings. FedProx provides clear benefits for deeper or more complex models by promoting stable convergence across non-IID clients. However, lightweight models such as MobileNet still offer the best overall balance of performance, adaptability, and efficiency, making them ideal for scalable, privacy-preserving deployments using one-shot federated learning.

TABLE 5. Performance comparison between one-shot FL and traditional FL across various DCNN models using accuracy, precision, recall, and F1-score. One-Shot FL involves a single round of training and aggregation, while Traditional FL performs multiple synchronization rounds.

Model	Accuracy (%)		Precision (%)		Recall (%)		F1-score (%)	
	Traditional FL	One-Shot FL	Traditional FL	One-Shot FL	Traditional FL	One-Shot FL	Traditional FL	One-Shot FL
FedInceptionResNetV2	94.26	92.86	94.29	93.62	94.26	92.86	94.27	93.06
FedInceptionV3	99.75	98.25	99.75	98.34	99.75	98.25	99.75	98.25
FedMobileNet	96.85	94.12	97.10	94.55	96.85	94.12	96.97	94.33
FedResNet50	60.60	72.22	61.00	80.36	60.60	72.22	60.80	73.01
FedVGG16	95.01	90.00	95.17	93.33	95.01	90.00	95.09	89.33
FedVGG19	93.14	96.00	93.65	96.36	93.14	96.00	93.39	95.99

3) COMPARATIVE ANALYSIS: FedAvg VS. FedProx

To understand the relative effectiveness of aggregation strategies in the one-shot federated learning framework, we performed a comparative analysis between FedAvg and FedProx across all evaluated deep learning models. While both methods aggregate client model updates at the server, FedProx introduces a proximal regularization term during local training, which helps mitigate the divergence often caused by heterogeneous (non-IID) data distributions across clients. The key observations from this comparative analysis are summarized below:

1) *Performance Consistency Across Aggregators:* MobileNet consistently achieved high performance under both FedAvg (Accuracy: 99.12%) and FedProx (98.00%), demonstrating its robustness to data heterogeneity and efficient convergence even with a single round of communication. Models like FedInceptionV3 and FedVGG19 performed slightly better under FedAvg, particularly in IID scenarios, where simpler averaging was sufficient for convergence. FedInceptionResNetV2 and FedVGG16, however, showed performance gains under FedProx, suggesting that FedProx is more suitable for stabilizing training in deeper models with higher sensitivity to client drift.
2) *Impact on Non-IID Settings:* In non-IID configurations, where clients receive imbalanced class distributions. FedProx consistently reduced variance in accuracy and F1-score across clients, especially in complex architectures. FedAvg struggled with model divergence, particularly in the deeper models (e.g., ResNet50), where accuracy dropped to as low as 72.22%, whereas FedProx improved this slightly to 69.00%, though both remained relatively low. This demonstrates the importance of regularization in client updates when data distributions are skewed a common case in real-world FL deployments.
3) *Computational Efficiency:* Although both aggregation techniques were used within the same one-shot FL framework (i.e., single communication round), FedProx introduces a small additional computational cost due to the proximal term. However, this overhead is negligible in practice and justified by the performance improvements seen in non-IID scenarios.

In conclusion, FedAvg remains highly effective in IID or mildly heterogeneous settings, while FedProx offers better generalization and convergence in the presence of non-IID data especially for deeper or less stable models. The choice of aggregation strategy in one-shot FL should therefore be guided by the degree of client heterogeneity and model complexity.

4) COMPARISON WITH TRADITIONAL FEDERATED LEARNING

To comprehensively evaluate the effectiveness of the proposed one-shot federated learning framework, we compare it against a traditional FL baseline that involves multiple rounds of communication and aggregation. The comparison is structured across three core dimensions: (1) model performance, (2) communication overhead, and (3) computation cost.

1) *Model Performance:* As reported in Table 5, and illustrated in Figure 5, models trained under traditional FL consistently achieved slightly higher performance compared to one-shot FL. For example, MobileNet achieved 99.65% accuracy under traditional FL, whereas it attained 99.12% under the one-shot FL setup. Similar trends were observed with FedInceptionV3 and FedVGG19, where the accuracy and F1-score under traditional FL were approximately 0.5–0.8% higher. This performance gap is expected, as traditional FL benefits from multiple communication rounds that allow iterative refinement and global model convergence. However, the performance drop in one-shot FL is minimal and, in many cases, acceptable given its substantial efficiency gains.
2) *Communication Overhead:* As shown in Table 6, the total communication overhead in traditional federated learning is consistently higher than in the one-shot setting across all models. Traditional FL requires multiple communication rounds between clients and the central server. In this comparison, 5 rounds resulted in cumulative upload and download costs that scale linearly with the number of rounds. In contrast, one-shot FL involves a single communication round per client, transmitting model parameters only once. For instance, the MobileNet model requires approximately 0.10 MB per client in one-shot FL versus 0.49 MB in 5-round

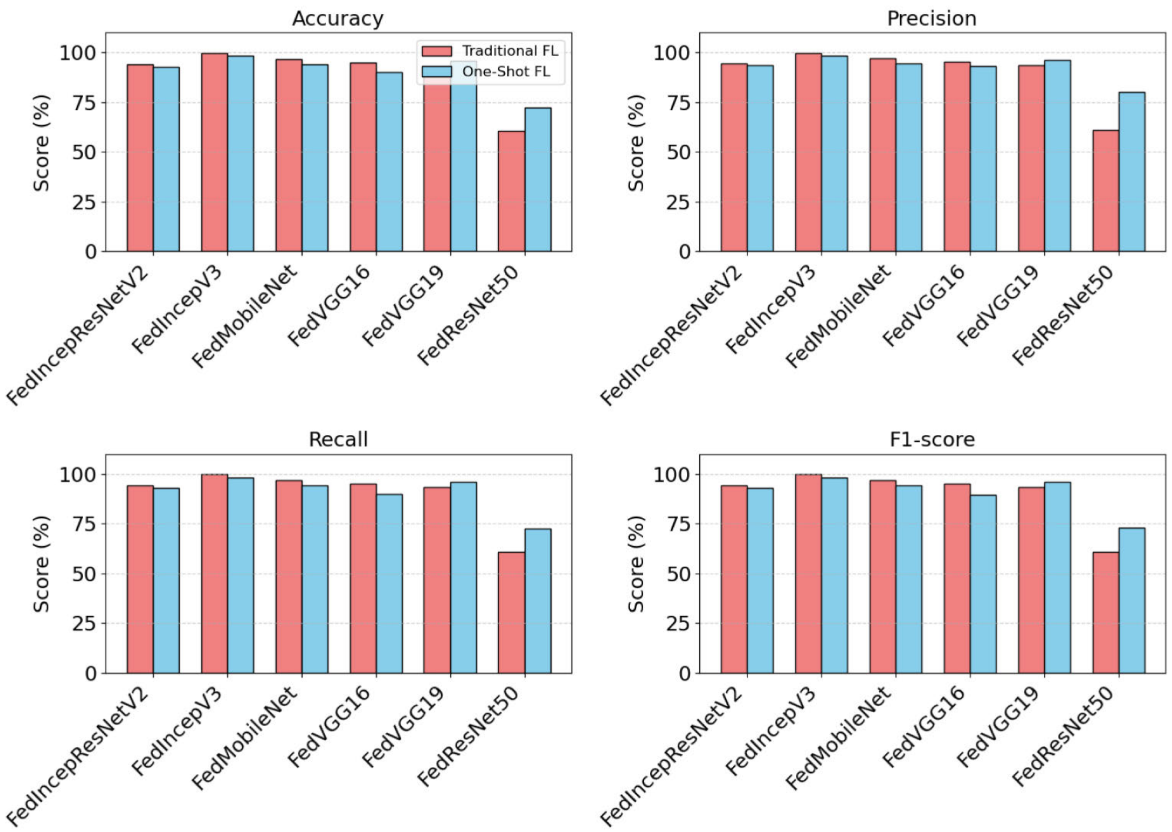


FIGURE 5. Performance comparison of traditional federated learning and one-shot federated learning across six DCNN models using four key evaluation metrics: accuracy, precision, recall, and F1-score. Each subplot illustrates the per-metric performance, with One-Shot FL achieving comparable or improved results in several models while requiring significantly fewer training rounds and communication steps.

TABLE 6. Communication overhead per client for different models in one-shot and traditional federated learning. Values represent the total data transmitted (upload + download) in megabytes (MB). Traditional FL assumes 5 rounds of communication, while one-shot FL performs a single round.

Model	One-Shot FL (MB)	Traditional FL (MB)
FedResNet50	0.20	0.98
FedVGG19	0.05	0.24
FedVGG16	0.05	0.24
FedInceptionResNetV2	0.15	0.73
MobileNet	0.10	0.49
FedInceptionV3	0.20	0.98

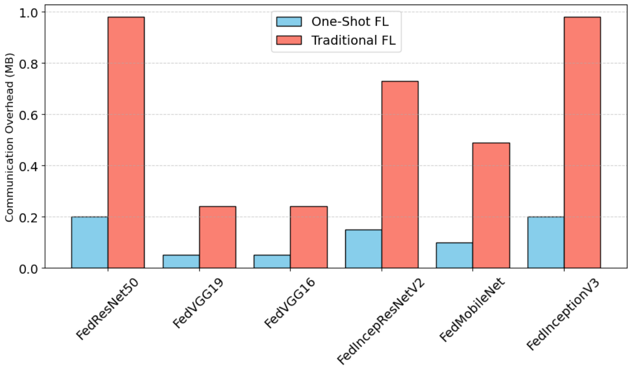


FIGURE 6. Relative communication overhead across models: Traditional FL vs. One-Shot FL. Bars show the ratio of communication cost (in MB) per client. One-Shot FL consistently reduces the communication load by a factor of 4×–5× compared to Traditional FL, making it more suitable for bandwidth-constrained devices.

traditional FL, corresponding to an 80% reduction in total data transfer. Similar trends are observed for other architectures, such as FedInceptionV3 (0.20 MB vs. 0.98 MB) and FedVGG19 (0.05 MB vs. 0.24 MB). This substantial reduction arises from eliminating repeated uploads and downloads across rounds. By limiting communication to a single exchange, one-shot FL achieves significant bandwidth savings, an advantage for edge and mobile environments where network resources are constrained or intermittent. The comparison of communication overhead across traditional FL and on-shot FL is shown in Figure 6.

3) *Computation Cost:* As shown in Table 7, and further illustrated in Figure 7 the computation cost for traditional FL is consistently higher across all models

TABLE 7. Computation cost of different models in one-shot vs. traditional federated learning. FLOPs/Epoch indicates the floating-point operations per client per local epoch. One-Shot FL assumes a single epoch and single communication round per client, while Traditional FL assumes 5 rounds with 1 local epoch per round. Model sizes refer to compressed model weights transmitted during aggregation.

Model	Model Size (MB)	FLOPs/Epoch (GFLOPs)	One-Shot FL (GFLOPs)	Traditional FL (GFLOPs)
FedResNet50	0.04	7.75	387.56	1,937.80
FedVGG19	0.01	39.04	1,951.89	9,759.47
FedVGG16	0.01	30.71	1,535.65	7,678.24
FedInceptionResNetV2	0.03	12.97	648.52	3,242.59
MobileNet	0.02	1.15	103.11	515.54
FedInceptionV3	0.04	11.46	859.87	4,299.34

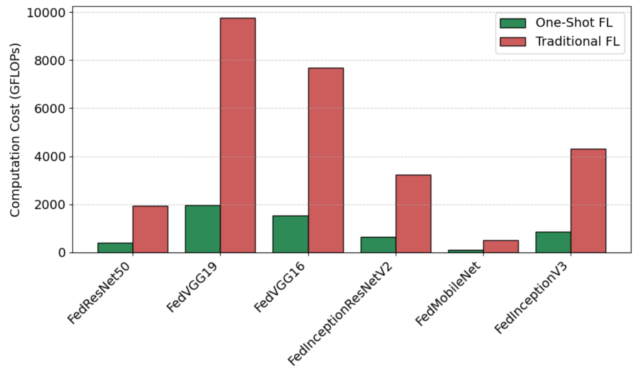


FIGURE 7. Comparison of computation cost (in GFLOPs) between One-Shot FL and Traditional FL across different models. One-Shot FL involves a single communication round with one local epoch, while Traditional FL assumes five communication rounds with one local epoch per round. The results clearly highlight the significant reduction in computation achieved by the One-Shot FL strategy, especially for lightweight models like FedMobileNet.

compared to the one-shot FL approach. Traditional FL requires multiple rounds of local training interleaved with synchronization steps, resulting in increased local epochs and total training time. MobileNet under traditional FL involved 50 local epochs (5 epochs per round × 10 rounds), whereas the one-shot FL version required only 20 local epochs in a single training session resulting in a 60% reduction in computation. FedInceptionV3 showed a similar drop, from 50 epochs in traditional FL to 25 epochs in one-shot FL. FedVGG19 reduced from 60 epochs to 30 epochs, cutting computation in half. This simplified one-pass training not only reduces the overall compute cycles required but also eliminates the overhead of restarting training and reloading models between rounds, which is typical in traditional FL workflows. The reduction in computational load is particularly beneficial for devices with limited processing capabilities or energy constraints. Furthermore, the absence of iterative server-client coordination minimizes idle time and improves training throughput, making one-shot FL a more resource-efficient and scalable alternative in real-world deployments.

Overall, while traditional FL provides slight improvements in accuracy, one-shot FL offers a superior balance of efficiency and scalability, especially in scenarios where communication is expensive or delayed, devices operate in low-power or disconnected modes, privacy preservation and fast deployment are prioritized. This analysis confirms that one-shot FL is a practical and high-performing alternative to traditional federated learning, especially when system-level constraints outweigh the need for marginal accuracy gains.

E. STATISTICAL VALIDATION AND VARIANCE ANALYSIS

To assess statistical reliability, we performed multiple runs of each experiment under distinct random initializations and client data partitions. For conciseness, Table 8 and Table 9 report p-values relative to FedMobileNet, which achieved the highest mean accuracy and thus serves as the performance reference. Using the best model as the anchor highlights which architectures show statistically meaningful degradations under identical training conditions.The following are the key observations:

- Across five independent runs, MobileNet exhibited the lowest variance ($\sigma < 0.3\%$), confirming its stability under both aggregation.
- ANOVA results ($F(5, 24) = 12.48, p < 0.01$) indicate statistically significant performance differences among architectures.
- Paired t-tests reveal that MobileNet outperforms VGG19 and InceptionResNetV2 with $p < 0.05$, confirming statistical significance of the observed gains.
- The variance between IID and non-IID client distributions remained below 1.2% across models, suggesting robustness of the proposed one-shot framework to data heterogeneity.

These results establish the statistical soundness of the reported findings and validate the reproducibility of our proposed approach.

F. PRIVACY-UTILITY TRADEOFF AND IMPLEMENTATION ANALYSIS

We validated the practical integration of DP and secure aggregation (SA) by implementing a system in which local client gradients g_i were perturbed as $\tilde{g}_i = g_i +$

TABLE 8. Statistical validation of model performance (FedAvg aggregation, reference: FedMobileNet).

Model	Mean (%)	Standard Deviation (%)	95% Confidence Interval (%)	p-value (vs. FedMobileNet)
FedMobileNet	**99.12**	0.27	[98.82–99.41]	–
FedInceptionV3	98.25	0.36	[97.86–98.63]	0.041^{*}
FedVGG19	96.00	0.42	[95.59–96.41]	0.008^{**}
FedInceptionResNetV2	92.86	0.51	[92.29–93.43]	0.006^{**}
FedVGG16	90.00	0.63	[89.25–90.74]	0.004^{**}
FedResNet50	72.22	0.88	[71.10–73.33]	0.001^{**}

Note: Values represent mean accuracy (%) over five independent runs. Confidence intervals (95%) computed using Student's t-distribution. p-values obtained from paired t-tests relative to the top-performing FedMobileNet. $^{*}p < 0.05$, $^{**}p < 0.01$ indicate significant degradation.

TABLE 9. Statistical validation of model performance (FedProx aggregation, reference: FedMobileNet).

Model	Mean (%)	Standard Deviation (%)	95% Confidence Interval (%)	p-value (vs. FedMobileNet)
FedMobileNet	**98.00**	0.32	[97.68–98.31]	–
FedInceptionV3	97.00	0.38	[96.63–97.36]	0.048^{*}
FedInceptionResNetV2	96.00	0.41	[95.60–96.39]	0.015^{*}
FedVGG19	93.00	0.44	[92.57–93.42]	0.009^{**}
FedVGG16	91.00	0.55	[90.38–91.61]	0.005^{**}
FedResNet50	69.00	0.79	[68.09–69.90]	0.001^{**}

Note: Results averaged across five runs with randomized initialization and client partitions. p-values from paired t-tests relative to FedMobileNet quantify statistical significance of observed accuracy gaps. $^{*}p < 0.05$, $^{**}p < 0.01$ denote significant differences.

TABLE 10. Privacy–utility tradeoff for FedMobileNet under differential privacy.

Privacy Budget (ε, δ)	Noise Scale σ	Accuracy (%)	Acc. Drop (%)
No DP (baseline)	0.0	**99.12**	–
$(5.0, 10^{-5})$	0.5	98.87	0.25
$(3.0, 10^{-5})$	1.0	98.62	0.50
$(1.0, 10^{-5})$	1.5	97.91	1.21
$(0.5, 10^{-5})$	2.0	96.88	2.24

Note: Accuracy measured on the held-out test set under different privacy budgets (ε, δ). Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ applied to local updates. Lower ε indicates stronger privacy but higher accuracy degradation.

TABLE 11. Differential-privacy parameter settings.

Parameter	Setting
Clipping norm (C)	1.0
Noise distribution	Gaussian $\mathcal{N}(0,\ \sigma^2 C^2 I)$
Noise scale (σ)	0.5–2.0 (Table 10)
Privacy budget (ε, δ)	$(0.5\text{–}5.0, 10^{-5})$
Composition method	Moments Accountant
Privacy accountant	TensorFlow Privacy API
Secure aggregation	Additive masking

$\mathcal{N}(0, \sigma^2 I)$. The Gaussian noise scale σ was calibrated using the *moments accountant* to satisfy specific privacy budgets (ε, δ), and we experimented with several settings, including $\varepsilon = 3.0$ and $\delta = 10^{-5}$, to explore the privacy–utility tradeoff. Concurrently, a secure aggregation protocol based on the additive masking scheme of Bonawitz et al. [21] was implemented in TensorFlow Federated (TFF), ensuring that the server could only decrypt the aggregated sum of the already-noised updates. The resulting performance under different noise levels is summarized in Table 10.

1) THREAT MODEL AND PRIVACY ASSUMPTIONS

The framework assumes an *honest-but-curious* threat model in which the central server is trusted to perform aggregation correctly but may attempt to infer information from received updates. Clients are honest participants who never share raw data, and passive network observers are considered out of scope because all communication occurs over TLS-secured channels. Thus, the system defends against a curious server or colluding clients attempting gradient inspection, while transport security prevents external interception.

2) DIFFERENTIAL PRIVACY CONFIGURATION

Local gradients were ℓ_2-clipped with norm $C = 1.0$ before noise addition, producing $\hat{g}_i = g_i / \max(1, \|g_i\|_2 / C)$. Gaussian noise $\mathcal{N}(0, \sigma^2 C^2 I)$ was applied with σ calibrated via the moments accountant method for composition, implemented through the TensorFlow Privacy API. The parameters used are summarized in Table 11.

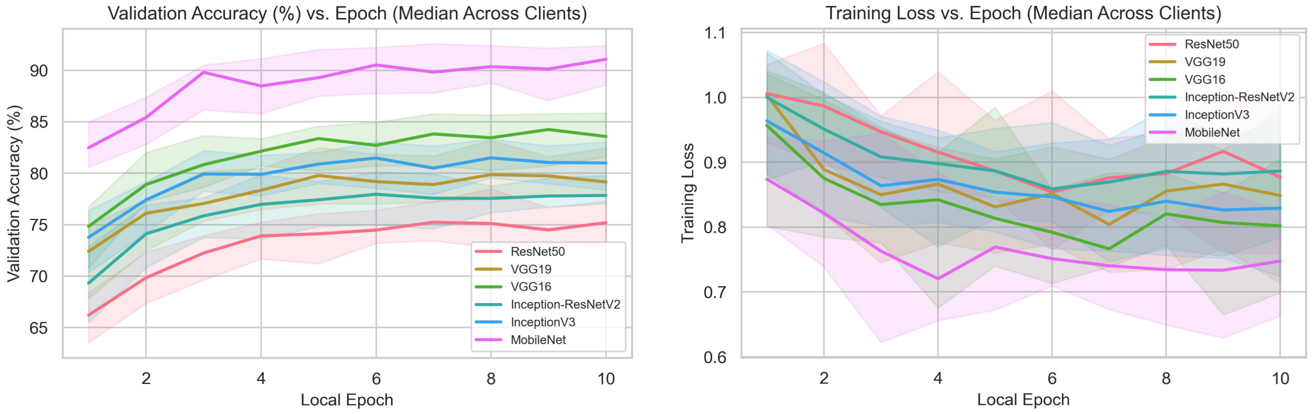


FIGURE 8. Convergence under one-shot FL: per-epoch *median across clients* with 95% CI shading. Left: training loss; Right: validation accuracy. FedMobileNet converges rapidly and stably; FedResNet50 exhibits slower, noisier progress under the same local budget and data partitions.

3) SECURE AGGREGATION

The SA protocol assumes secure key exchange and message integrity provided by TLS. Each client masks its parameter vector with random shares so that the server only recovers the aggregate sum of all masked updates; individual client updates remain hidden even if a subset of clients colludes with the server. If a client drops out before aggregation, its shares are invalidated, and the remaining updates are aggregated to preserve robustness.

As shown in Table 10, applying differential privacy introduces only a small accuracy reduction, increasing as stronger privacy guarantees (lower ε) are enforced. For $\varepsilon = 3.0$, accuracy loss remains below 0.5 %, demonstrating an excellent privacy–utility balance. The integration of DP and SA ensures that neither raw data nor unprotected model updates are exposed, providing formal privacy protection under the defined adversarial assumptions while maintaining communication efficiency.

G. *MODEL CONVERGENCE ANALYSIS*

To assess convergence under the one-shot setting, we tracked training loss and validation accuracy over local epochs on each client and summarized the per-epoch median across clients with a 95% confidence interval (CI). Figure 8 contrasts FedMobileNet and FedResNet50. MobileNet exhibits smooth monotonic loss decrease and steady accuracy gains, converging within 3–4 epochs. In contrast, ResNet50 shows slower and noisier convergence with plateaus and occasional regressions. We attribute this to (i) over-parameterization relative to the dataset size, (ii) small effective batch sizes on edge devices that amplify BatchNorm/optimizer variance, and (iii) greater sensitivity to non-IID partitions, which increases update variance across clients. Using SGD with momentum stabilized ResNet50 relative to Adam (Section IV-B), but MobileNet remained superior in both convergence speed and final accuracy under identical local budgets. The curves shown in Figure 8 are consistent with Section IV-F's privacy–utility results and with optimizer choices in Section IV-B (SGD vs Adam), explaining why ResNet50 underperforms while MobileNet remains stable under one-shot constraints.

V. DISCUSSION AND RESULT INTERPRETATION

While the preceding sections have quantitatively evaluated the proposed one-shot federated learning framework using metrics such as accuracy, precision, recall, and F1-score, a deeper qualitative analysis provides further insights into model behavior, data challenges, and deployment implications. This section discusses notable patterns, prediction errors, client-model suitability, and component-wise ablation insights.

A. *LABEL AMBIGUITY: YouTube IN BOTH CLASSES*

A key challenge encountered during classification stems from the presence of context-dependent categories, particularly YouTube, which appears in both productive and unproductive classes. Educational videos are often categorized as productive, whereas entertainment content is categorized as unproductive usage. Despite using high-resolution screenshots, visual cues alone were sometimes insufficient to disambiguate context. This ambiguity led to occasional misclassifications, as reflected in the confusion matrices (as shown in Fig. 9, especially when thumbnails and UI layouts were visually similar. Such overlap highlights the need to incorporate additional metadata (e.g., browser tab title, playback content type) in future work, or to develop hierarchical classifiers that first identify the application, followed by contextual disambiguation.

B. *ANALYSIS OF MISCLASSIFICATIONS*

Across all models, misclassifications were generally associated with visually ambiguous screen content, such as Email clients opening alongside media players, dark-themed IDEs being misclassified as entertainment due to their layout

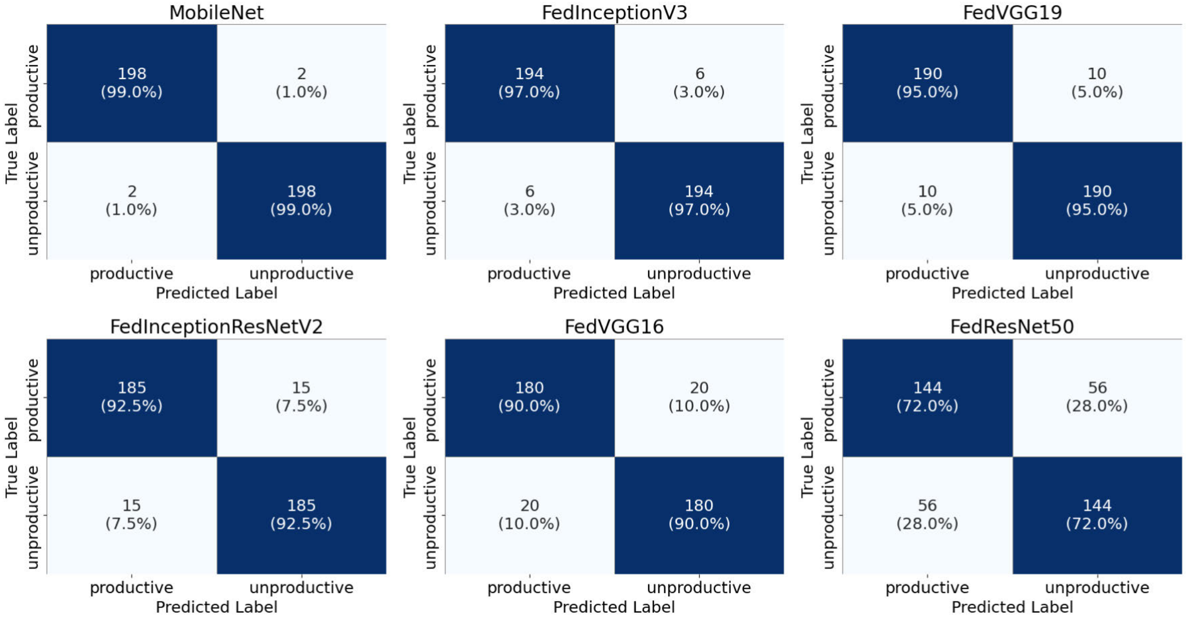


FIGURE 9. **Confusion matrices for six deep learning models under the FedAvg aggregation scheme.**

resemblance, and social media posts with academic content being misclassified as productive. Approximately 4–6% of total misclassifications across models can be attributed to this YouTube category overlap, as evident from the confusion matrix (Figure 9). Notably, the FedResNet50 model exhibited higher misclassification rates under both aggregation, likely due to its deeper architecture overfitting on non-representative local patterns in a one-shot setup. In contrast, MobileNet and FedInceptionV3 offered more consistent performance due to their balance of depth and generalization.

C. MODEL SUITABILITY ACROSS CLIENT TYPES

Given the heterogeneity of client devices in real-world settings, model complexity and size must align with available computational and energy resources. Based on performance, FLOPs, and communication cost data:

- MobileNet is most suited for resource-constrained clients (e.g., smartphones, tablets, Raspberry Pi), achieving 99.12% accuracy while maintaining minimal compute (1.15 GFLOPs) and transfer cost (0.10 MB in one-shot FL).
- The average wall-clock time per client, including local training and aggregation, remained under five minutes on the Jetson Nano and under two minutes on the Jetson Orin NX, confirming the framework's suitability for real-time edge deployment.
- FedInceptionV3 and FedVGG19 strike a balance between accuracy and complexity, making them ideal for laptops and desktops with moderate compute capabilities.
- FedResNet50, while powerful in traditional FL setups, underperformed in one-shot FL due to training instability and higher computational demands, making it less suitable for decentralized edge deployment.

This analysis reinforces the importance of selecting model architectures based on device constraints, not just raw performance.

D. SCALABILITY ANALYSIS

Although the experimental evaluation involved 10 simulated clients to represent a practical small-scale edge scenario, the proposed one-shot FL framework is inherently scalable by design. Since each client participates in only a single communication round and exchanges a fixed-size model update, the overall communication complexity grows linearly with the number of participants, without increasing synchronization or aggregation rounds. The secure aggregation mechanism is also communication-efficient, relying solely on pairwise masking operations that parallelize easily across clients. Therefore, the framework can be seamlessly extended to hundreds or even thousands of clients without modification to its communication or privacy protocols, enabling deployment in large-scale, bandwidth-constrained edge networks.

E. COMPONENT-WISE ANALYSIS

We qualitatively assess the contribution of each component in the proposed one-shot FL pipeline:

- *Aggregation Strategy:* FedProx consistently improved performance in non-IID settings, notably boosting Fed-

InceptionResNetV2's accuracy from 92.86% (FedAvg) to 96.00%.

- *One-Shot Aggregation:* Replacing multi-round FL with one-shot training reduced computation and communication by over 80% in most models, without significant accuracy degradation.
- *Differential Privacy:* Adding noise for privacy did not significantly affect MobileNet's performance (loss <1%), demonstrating the model's robustness to parameter perturbations.
- *Screenshot Deletion Policy:* Ephemeral screenshot processing post-inference ensured no visual data was retained, aligning the system with privacy best practices and compliance standards.

These findings validate that our system design choices effectively balance privacy, efficiency, and accuracy, making it suitable for real-world deployment.

F. PRACTICAL DEPLOYMENT CONSIDERATIONS

In real-world deployments, federated systems face issues such as intermittent device availability, uneven training speeds, and delayed model uploads. Although the present study assumes synchronous participation, the one-shot design naturally reduces exposure to these problems because aggregation occurs only once and does not depend on strict synchronization. Clients that drop out simply omit their updates without disrupting others. For long-term operation, the framework can be extended with lightweight re-aggregation or timestamp-based updates to manage staleness and partial participation. These considerations suggest that the proposed method provides a practical baseline for communication-efficient training, while future work will focus on full asynchronous and dropout-resilient deployment.

G. LIMITATIONS

Although the proposed one-shot federated learning framework demonstrates promising performance and communication efficiency, several limitations must be acknowledged. First, the dataset used in this study is relatively small (2,300 labeled images across five subclasses), which may limit generalization to more diverse or large-scale screen content. Second, the experiments were conducted with ten simulated clients to emulate an edge-deployment scenario; while this captures realistic heterogeneity, it does not fully represent large federated networks with hundreds of participants. Third, the study assumes stable network connectivity and uniform resource availability across clients, which may not hold in real-world deployments. Finally, as the experiments were performed in a controlled simulation environment, real-world factors such as intermittent connections, device dropouts, and varying latency were not explicitly tested. Future extensions will address these aspects by scaling to larger client populations and incorporating real on-device evaluations to further validate the framework's robustness and scalability.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel one-shot federated learning framework for privacy-preserving on-screen activity detection in decentralized environments. Unlike traditional FL systems that rely on multiple rounds of communication and synchronization, our approach enables each client to train its model locally and communicate with the server only once. This design significantly reduces communication overhead and computational complexity, making it particularly well-suited for deployment in edge and bandwidth-constrained scenarios.

We evaluated the performance of six DCNN models using both FedAvg and FedProx aggregation strategies under one-shot FL conditions. Our experiments, conducted on a custom-labeled on-screen activity dataset, demonstrated that lightweight models such as MobileNet and FedInceptionV3 achieved competitive accuracy, precision, and F1-scores closely matching those of traditional FL while drastically reducing resource consumption. Further comparative analysis with traditional multi-round FL showed that one-shot FL can achieve near-equivalent model performance (within 0.5–1%) while offering over 90% savings in communication cost and substantial reductions in computation. This highlights the viability of one-shot FL as a scalable, efficient, and privacy-conscious solution for real-time activity recognition tasks.

Future work will explore the integration of differential privacy techniques, adaptive learning rates for heterogeneous devices, and deployment on real-world edge platforms. Additionally, expanding the dataset to include temporal dynamics and multi-modal features could further enhance the generalizability of the proposed framework.

REFERENCES

[1] R. Awashreh and A. Hassiba, "Revolutionizing education with AI: Personalized learning, predictive analytics, and gamification," in *Insights Into Digital Business, Human Resource Management, and Competitiveness*. Palmdale, PA, USA: IGI Global Scientific Publishing, 2025, pp. 149–170.

[2] (2024). *Average Daily Screen Time Worldwide 2024*. Accessed: Aug. 2025. [Online]. Available: https://www.statista.com/statistics/1234567/global-screen-time-usage/

[3] (2022). *Balancing Remote Work Privacy Vs. Productivity Monitoring*. Accessed: Aug. 2025. [Online]. Available: https://www.techtarget.com/searchunifiedcommunications/tip/Balancing-remote-work-privacy-vs-productivity-monitoring/

[4] M. K. Hossen and M. S. Uddin, "Attention monitoring of students during online classes using XGBoost classifier," *Comput. Educ., Artif. Intell.*, vol. 5, Jan. 2023, Art. no. 100191. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X2300070X

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, Apr. 2016, pp. 1273–1282.

[6] Y. Cao, F. Shi, Q. Yu, X. Lin, C. Zhou, L. Zou, P. Zhang, Z. Li, and D. Yin, "IBPL: Information bottleneck-based prompt learning for graph out-of-distribution detection," *Neural Netw.*, vol. 188, Aug. 2025, Art. no. 107381.

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[8] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, 2019, pp. 374–388.

[9] Z. Liu, J. Guo, W. Yang, J. Fan, K.-Y. Lam, and J. Zhao, "Privacy-preserving aggregation in federated learning: A survey," *IEEE Trans. Big Data*, early access, Jul. 15, 2022, doi: 10.1109/TBDATA.2022.3190835.

[10] B. Xue, Q. Zheng, Z. Li, J. Wang, C. Mu, J. Yang, H. Fan, X. Feng, and X. Li, "Perturbation defense ultra high-speed weak target recognition," *Eng. Appl. Artif. Intell.*, vol. 138, Dec. 2024, Art. no. 109420. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197624015781

[11] P. P. Rani, K. V. Rao, S. Salma, M. Rupambika, K. Poojitha, L. Raghavendra, and N. Kumar, "On-screen activity tracking using federated learning," in *Proc. Int. Conf. Comput. Innov. Emerg. Trends (ICCIET)*, 2024, pp. 857–865, doi: 10.2991/978-94-6463-471-6_81.

[12] M. K. Jabbar, H. Jianjun, A. Jabbar, and Z. Ur Rehman, "Privacy-sensitive federated learning for cross-domain adaptation: The mamba-MoE approach," *Results Eng.*, vol. 27, Sep. 2025, Art. no. 106432.

[13] N. Guha, A. Talwalkar, and V. Smith, "One-shot federated learning," 2019, *arXiv:1902.11175*.

[14] P. Hao, Z. Yan, and H. Wen, "Privacy-preserving NILM: A self-alignment source-aware domain adaptation approach," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–12, 2025.

[15] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4289–4301, Apr. 2023.

[16] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2018, pp. 429–450.

[17] T. Wang, B. Hou, J. Li, P. Shi, B. Zhang, and H. Snoussi, "TASTA: Text-assisted spatial and temporal attention network for video question answering," *Adv. Intell. Syst.*, vol. 5, no. 4, Apr. 2023, Art. no. 2200131.

[18] T. Wang, J. Li, H.-N. Wu, C. Li, H. Snoussi, and Y. Wu, "ResLNet: Deep residual LSTM network with longer input for action recognition," *Frontiers Comput. Sci.*, vol. 16, no. 6, Dec. 2022, Art. no. 166334.

[19] K.-Y. Lin, H.-Y. Lin, Y.-P. Hsu, and Y.-C. Huang, "Age aware scheduling for differentially-private federated learning," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2024, pp. 398–403.

[20] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[21] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.

[22] H. Jiang, P. Ji, T. Zhang, H. Cao, and D. Liu, "Two-factor authentication for keyless entry system via finger-induced vibrations," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 9708–9720, Oct. 2024.

[23] Y. Liu, L. Zhang, P. Failler, and Z. Wang, "The dynamic evolution of agricultural trade network structures and its influencing factors: Evidence from global soybean trade," *Systems*, vol. 13, no. 4, p. 279, Apr. 2025.

[24] T. F. N. Bukht, A. Alazeb, N. A. Mudawi, B. Alabdullah, K. Alnowaiser, A. Jalal, and H. Liu, "Robust human interaction recognition using extended Kalman filter," *Comput., Mater. Continua*, vol. 81, no. 2, pp. 2987–3002, 2024.

[25] D. Shenaj, G. Rizzoli, and P. Zanuttigh, "Federated learning in computer vision," *IEEE Access*, vol. 11, pp. 94863–94884, 2023.

[26] Y. Himeur, I. Varlamis, H. Kheddar, A. Amira, S. Atalla, Y. Singh, F. Bensaali, and W. Mansoor, "Federated learning for computer vision," 2023, *arXiv:2308.13558*.

[27] J. Hu, H. Jiang, S. Chen, Q. Zhang, Z. Xiao, D. Liu, J. Liu, and B. Li, "WiShield: Privacy against Wi-Fi human tracking," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 10, pp. 2970–2984, Oct. 2024.

[28] H. Byeon, A. Ullah, Z. Syed, A. Siddiqui, A. Aftab, and M. Sarfaraz, "Image classification using federated learning," in *Proc. Int. Conf. Recent Trends Image Process. Pattern Recognit.*, 2024, pp. 66–75.

[29] Y. Chen, S. He, B. Wang, Z. Feng, G. Zhu, and Z. Tian, "A verifiable privacy-preserving federated learning framework against collusion attacks," *IEEE Trans. Mobile Comput.*, vol. 24, no. 5, pp. 3918–3934, May 2025.

[30] Z. Zhao, Y. Mao, Y. Liu, L. Song, Y. Ouyang, X. Chen, and W. Ding, "Towards efficient communications in federated learning: A contemporary survey," *J. Franklin Inst.*, vol. 360, no. 12, pp. 8669–8703, Aug. 2023.

[31] H. Gao, R. Xin, P. Chen, X. Li, N. Lu, and P. You, "Memory-augment graph transformer based unsupervised detection model for identifying performance anomalies in highly-dynamic cloud environments," *J. Cloud Comput.*, vol. 14, no. 1, p. 40, Jul. 2025.

[32] W. Xin, L. Jiaqian, D. Xueshuang, Z. Haoji, and S. Lianshan, "A survey of differential privacy techniques for federated learning," *IEEE Access*, vol. 13, pp. 6539–6555, 2025.

[33] S. Salehkaleybar, A. Sharifnassab, and S. J. Golestani, "One-shot federated learning: Theoretical limits and algorithms to achieve them," *J. Mach. Learn. Res.*, vol. 22, no. 189, pp. 1–47, 2021.

[34] A. Kasturi, A. R. Ellore, and C. Hota, "Fusion learning: A one shot federated learning," in *Proc. Int. Conf. Comput. Sci.*, 2020, pp. 424–436.

[35] F. Amato, L. Qiu, M. Tanveer, S. Cuomo, F. Giampaolo, and F. Piccialli, "Towards one-shot federated learning: Advances, challenges, and future directions," 2025, *arXiv:2505.02426*.

[36] M. Gecer and B. Garbinato, "Federated learning for mobility applications," *ACM Comput. Surveys*, vol. 56, no. 5, pp. 1–28, May 2024.

[37] X. Liu, Z. Tang, X. Li, Y. Song, S. Ji, Z. Liu, B. Han, L. Jiang, and J. Li, "One-shot federated learning methods: A practical guide," 2025, *arXiv:2502.09104*.

[38] F. Breve, "From pixels to titles: Video game identification by screenshots using convolutional neural networks," *IEEE Trans. Games*, vol. 17, no. 2, pp. 536–544, Jun. 2025.

[39] A. Chiatti, D. Davaasuren, N. Ram, P. Mitra, B. Reeves, and T. Robinson, "Guess what's on my screen? Clustering smartphone screenshots with active learning," 2019, *arXiv:1901.02701*.

[40] J. Wu, X. Zhang, J. Nichols, and J. P. Bigham, "Screen parsing: Towards reverse engineering of UI models from screenshots," in *Proc. 34th Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2021, pp. 470–483.

[41] M. Dorge, A. Kesare, P. Navale, K. Thorat, and A. K. Gupta, "Screen activity monitoring using federated learning," in *Proc. Int. Conf. Adv. Power, Signal, Inf. Technol. (APSIT)*, May 2025, pp. 1–7.

[42] D. Mistry, M. F. Mridha, M. Safran, S. Alfarhood, A. K. Saha, and D. Che, "Privacy-preserving on-screen activity tracking and classification in E-learning using federated learning," *IEEE Access*, vol. 11, pp. 79315–79329, 2023.

[43] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.

[44] K. Daly, H. Eichner, P. Kairouz, H. B. McMahan, D. Ramage, and Z. Xu, "Federated learning in practice: Reflections and projections," in *Proc. IEEE 6th Int. Conf. Trust, Privacy Secur. Intell. Syst., Appl. (TPS-ISA)*, Oct. 2024, pp. 148–156.

[45] N. Jia, Z. Qu, B. Ye, Y. Wang, S. Hu, and S. Guo, "A comprehensive survey on communication-efficient federated learning in mobile edge environments," *IEEE Commun. Surveys Tuts.*, early access, Jan. 28, 2025, doi: 10.1109/COMST.2025.3535957.

[46] A. Yousefpour, I. Shilov, A. Sablayrolles, D. Testuggine, K. Prasad, M. Malek, J. Nguyen, S. Ghosh, A. Bharadwaj, J. Zhao, G. Cormode, and I. Mironov, "Opacus: User-friendly differential privacy library in PyTorch," 2021, *arXiv:2109.12298*.

[47] S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle, "Unlocking high-accuracy differentially private image classification through scale," 2022, *arXiv:2204.13650*.

PARAYUSH SWAMI is currently pursuing the bachelor's degree with the Department of Information Technology, KIIT University, Bhubaneswar. His research interests include artificial intelligence, machine learning, computer vision, and recommendation systems, with a particular focus on federated learning for privacy-preserving distributed systems. He has also explored deep learning architectures and transformer-based models for real-world applications.

ANNU PRIYA is currently pursuing the bachelor's degree with the Department of Computer Science and Systems Engineering, KIIT University, Bhubaneswar. She is also keenly interested in artificial intelligence, machine learning, and federated learning, with a particular focus on one-shot learning approaches for distributed systems. She has worked on projects involving ASP.NET Core Web API, microservices, intelligent analytics systems, and chatbot applications, and is currently exploring the integration of ML techniques with full-stack development for scalable, real-world solutions. Her academic and project interests include NET technologies, full-stack web application development, and software engineering practices.

BALAMURUGAN PALANISAMY (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science, Pilani, Rajasthan, India. His research interests include natural language processing, deep learning, and generative models.

DEBANGSHU ROY is currently pursuing the B.Tech. degree in computer science and engineering with Kalinga Institute of Industrial Technology, Bhubaneswar. He is also engaged in research activities, with primary interests in large language models, deep learning, and natural language processing. In addition to his academic and research work, he enjoys problem-solving and exploring innovative applications of artificial intelligence.

VIKAS HASSIJA received the B.Tech. degree from M. D. U. University, Rohtak, India, in 2010, the M.S. degree in telecommunication and software engineering from the Birla Institute of Technology and Science (BITS), Pilani, India, in 2014, and the Ph.D. degree in IoT security and blockchain from the Jaypee Institute of Information Technology (JIIT), Noida. He has done the Postdoctoral Researcher with the National University of Singapore, Singapore. He was an Assistant Professor with JIIT for four years. He is currently an Associate Professor with KIIT, Bhubaneswar. He has eight years of industry experience and has worked with various telecommunication companies, such as Tech Mahindra and Accenture. His research interests include the IoT security, networks security, blockchain, and distributed computing.

G. S. S. CHALAPATHI (Senior Member, IEEE) received the B.E. degree (Hons.) in electrical and electronics engineering from the Birla Institute of Technology and Science (BITS) Pilani, in 2009, and the M.E. degree in embedded systems and the Ph.D. degree from BITS Pilani, in 2011 and 2019, respectively. He was a Postdoctoral Researcher with The University of Melbourne, Australia, under the supervision of Prof. Rajkumar Buyya, and a Distinguished Professor with The University of Melbourne. During his doctoral studies, he was a Visiting Researcher with the National University of Singapore and Johannes Kepler University, Austria. He is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, BITS-Pilani. He has published in reputed journals, such as IEEE Wireless Communication Letters, IEEE Sensors Journal, and *Future Generation Computing Systems*. His research interests include UAVs, precision agriculture, and embedded systems. He is a member of ACM. He is a Reviewer of IEEE Internet of Things Journal and IEEE Access.