Scholar Publishing Group
International Journal of Big Data Intelligent Technology
https://doi.org/10.38007/IJBDIT.2025.060210
ISSN 2790-0932 Vol. 6, Issue 2: 94-103

Cloud－Edge Collaborative Image Recognition Task Offloading: A Federated Learning－Driven Framework for Training－Inference Collaboration and Resource Optimization

Yizhou Meng

DRIVE VRI COGS 1010, Microsoft, Redmond, WA,98052, US

Keywords: Cloud edge collaboration; Task uninstallation; Image recognition; Federated Learning; Sample imbalance.

Abstract: With the deep integration of cloud－edge collaboration and artificial intelligence, image-based intelligent perception has been widely applied in fields such as healthcare and marine science. However, the traditional "cloud-side training—cloud/edge inference" paradigm faces challenges including bandwidth pressure, inference latency, deployment difficulty, and accuracy degradation caused by limited edge computing power and imbalanced sample distribution. To address these issues, this paper proposes a "cloud－edge collaborative image recognition task offloading framework driven by federated learning for training－inference collaboration and resource optimization." The framework focuses on three aspects: integrated training－inference, collaborative inference offloading, and federated aggregation weighting. First, a cloud－edge integrated training－inference task offloading model is constructed, formally characterizing the task－resource－data relationships. Using containerization and YAML orchestration, we implement an end-to-end workflow covering cloud training, image distribution, and edge deployment, and build an experimental platform based on Kubernetes/KubeEdge, which significantly reduces data transmission time and overall execution latency. Second, to tackle the problems of limited edge computing capacity and low confidence of lightweight models, we propose a collaborative inference migration strategy triggered by overload conditions and the upward transmission of low-confidence samples. This strategy is integrated as a plugin into the Kubernetes/KubeEdge/Sedna framework and is validated in pathological image analysis and marine fish recognition scenarios, demonstrating improvements in inference efficiency and recognition accuracy. Finally, to address the imbalance of local data samples in federated learning, we introduce a weighted aggregation method that simultaneously considers local model accuracy, stability, and sample size. This approach increases the contribution of high-quality local models in global aggregation. Experimental results show that the proposed method outperforms FedAvg under multiple imbalanced scenarios, effectively enhancing global model accuracy and robustness. Overall, the proposed framework realizes a closed loop between training and inference with collaborative optimization, providing a scalable engineering solution for large-scale, real-time, and high-accuracy cloud－edge image

recognition applications.

1. Introduction

In recent years, the deep integration of cloud computing, edge computing, and artificial intelligence has driven the widespread adoption of image recognition in fields such as healthcare, marine monitoring, and security. The cloud excels at training and storage, while the edge is suited for low-latency inference; cloud–edge collaboration has thus become a key pathway to ensure both real-time performance and accuracy. However, current research still faces several challenges: training and inference processes remain decoupled, lacking closed-loop management and resulting in low iteration efficiency; edge resources are limited, lightweight models suffer from insufficient accuracy, and computational bottlenecks are evident; collaborative inference lacks dynamic resource awareness and elastic migration mechanisms; and federated learning experiences accuracy degradation under data imbalance, undermining overall performance.

To address these issues, this study aims to build a cloud–edge collaborative framework that integrates closed-loop training–inference, efficient edge inference, and precise federated aggregation, thereby improving the usability and robustness of image recognition systems in distributed environments. We propose a "cloud–edge collaborative task offloading framework driven by federated learning for training–inference collaboration and resource optimization," which realizes an end-to-end automated process from training, image creation, and distribution to inference; introduces a collaborative inference strategy based on a dual-trigger mechanism of load and confidence for dynamic task migration; and designs a federated weighted aggregation method incorporating accuracy, stability, and sample size, enhancing the contribution of high-quality models while mitigating imbalance effects.

The main innovations are as follows: proposing a training–inference integrated task offloading model to achieve closed-loop management and automated deployment; designing a dual-trigger collaborative inference mechanism for resource-constrained scenarios to reduce latency and blocking; introducing a federated dynamic aggregation strategy based on multi-dimensional evaluation to improve global model accuracy and robustness; and implementing a prototype system on Kubernetes, KubeEdge, and Sedna. Experiments in medical and marine scenarios demonstrate significant improvements in efficiency, accuracy, and latency, highlighting the framework's scalability and application value.

2. Related Research

In the field of distributed intelligent perception, federated learning has emerged as an important paradigm in recent years, attracting widespread attention for its advantages in data privacy protection and cross-domain modeling. Zhang C [1]systematically expounded the mechanism by which federated learning protects data privacy through local training and model transmission, and comprehensively reviewed its current state and key challenges across five dimensions: algorithms, communication, systems, privacy, and applications. The study highlighted that practical deployment still faces critical issues such as communication overhead, system heterogeneity, and security protection. Further, Chen H [2]Y focused on the conflict between generalization and personalization performance caused by data distribution differences in federated learning, and proposed the Fed-RoD framework, which jointly optimizes global generalization and local personalization objectives, offering a new perspective for addressing the trade-off between adaptability and robustness in distributed scenarios.

Meanwhile, cloud–edge collaboration has rapidly developed as an important technical pathway to address limited computing power, constrained bandwidth, and stringent real-time requirements. Bao G [3]pointed out that the integration of cloud–edge collaboration and federated learning remains at an early stage, particularly with significant gaps in key technologies, practical challenges, and application integration, necessitating systematic exploration. Regarding specific mechanisms, Gu H [4]proposed a cloud–edge–end collaborative network based on deep reinforcement learning, which effectively solves dynamic decision-making problems such as task offloading and resource allocation, demonstrating the potential of intelligent scheduling in complex environments. Furthermore, Wang B [5]emphasized that task offloading in edge cloud computing still faces bottlenecks, and established a classification framework that summarizes various offloading strategies and their applicable scenarios[6], while also indicating that future research should balance system performance optimization with industrial deployment[7].

In summary, federated learning research has provided a solid theoretical and methodological foundation for distributed modeling and privacy protection, while cloud–edge collaboration research has gradually advanced task offloading and dynamic resource management [8]. However, most existing work focuses on optimizing single dimensions, and has yet to form an integrated closed loop covering model training, inference deployment, resource scheduling, and federated aggregation [9]. To bridge this gap [10], this paper proposes a "cloud–edge collaborative image recognition task offloading framework driven by federated learning for training–inference collaboration and resource optimization [11]," which integrates the above research directions and addresses current limitations, thereby enhancing the real-time performance, accuracy, and robustness of large-scale distributed image recognition applications [12].

3. Cloud–Edge Collaborative Image Recognition: Modeling and Integrated Training–Inference Implementation

3.1. Task–Resource–Data Modeling and Optimization Objectives in Cloud–Edge Collaboration

In cloud–edge collaborative image recognition scenarios, tasks, resources, and data constitute the core elements of the overall system. From a unified modeling perspective and in combination with the federated learning paradigm, this study proposes a collaborative optimization framework for integrated training and inference. The goal is to achieve a closed-loop linkage of training, model image construction and distribution, and edge inference, while balancing latency, energy consumption, and accuracy under data privacy protection. This modeling draws on the concept of integrated orchestration in prior work, but extends it in terms of task granularity, optimization objectives, and federated learning constraints.

From the perspective of system composition, cloud nodes undertake high-performance training and model aggregation, while edge nodes focus on real-time inference and local incremental training. Resources cover computing, storage, bandwidth, and energy consumption, which need to be reasonably allocated between cloud and edge. Tasks include training, image construction, distribution, and inference, and are coupled through data flows: raw training data remain on the edge, with only parameters or gradients uploaded; the cloud generates model images and distributes them to the edge; inference results and feedback from the edge are then returned to the cloud to trigger incremental training and model updates. System operation must satisfy multiple constraints. Capacity constraints require that task allocation and resource demands at each node do not exceed its capability. Latency constraints stipulate that end-to-end latency must remain within the threshold, covering training,

image construction and distribution, inference, and transmission. Privacy constraints prohibit direct leakage of edge data, which may only participate in training via federated learning. Model version consistency and deployment reliability must also be ensured to prevent service interruptions caused by frequent updates. For optimization objectives, this study introduces multi-objective functions to balance latency, energy consumption, bandwidth, and accuracy. A weighted-sum approach can minimize latency, energy, and bandwidth overhead while improving recognition accuracy. Alternatively, a hierarchical optimization strategy can first ensure latency and privacy constraints, then further reduce energy and bandwidth consumption within the feasible region, and finally enhance model accuracy. With the introduction of federated learning, accuracy improvement depends not only on single-node performance but also on the extent of participation of edge nodes in training and the efficiency of cloud aggregation.

In practical deployment, the modeling framework naturally aligns with cloud-native architectures. Through containerization and YAML-based orchestration, an automated pipeline for training, model construction and distribution, and inference can be established. Scheduling strategies between cloud and edge are dynamically determined by the system based on resource status and network conditions, thereby achieving true integration of training and inference.

3.2. Integrated Training – Inference Task Offloading Method and Implementation Architecture

In cloud – edge collaborative image recognition scenarios, the key to achieving integrated training – inference lies in constructing a complete closed-loop architecture that encompasses training, image construction and distribution, edge inference, and feedback-driven retraining. This architecture relies on containerization and declarative orchestration technologies to connect each stage, enabling models to transition in a standardized manner from training to deployment and inference, while continuously iterating and optimizing during runtime.

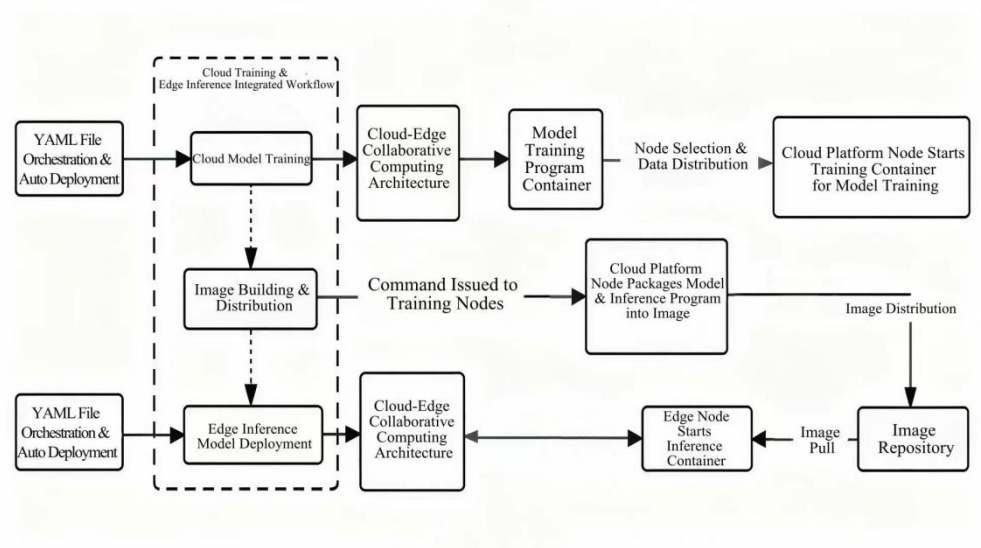

Figure 1. Integrated Workflow of Cloud Training and Edge Inference

As shown in Figure 1, the integrated workflow begins with orchestration instructions defined in YAML files, covering the entire process of cloud training, image creation and distribution, and edge inference. Specifically, on the cloud side, the system first performs centralized training and model aggregation. Training jobs define resource requests, data mounting, and runtime environments through container images, which are then allocated by the scheduler according to node load, thereby

shortening training time and improving resource utilization. Upon completion, model weights are packaged into images and stored in a repository, providing a standardized and versioned foundation for subsequent edge inference. In this way, model artifacts are transformed into directly callable runtime images, reducing deployment risks caused by environmental differences.

Edge nodes primarily handle real-time inference tasks. Inference containers define image versions, node selectors, resource requests, and runtime parameters through YAML files, and are scheduled to suitable edge nodes via the control plane. As task requests fluctuate, the system can automatically adjust the number of concurrent replicas, achieving elastic scaling to ensure low latency and high availability. When edge node resources are insufficient, inference confidence is low, or network conditions permit, the system offloads part of the tasks to the cloud, invoking high-accuracy models for verification inference and synchronizing the results back to the edge, thus striking a balance between latency and accuracy.

To further enhance long-term model performance, the system introduces a federated learning mechanism. Edge nodes conduct local training without exposing raw data, uploading gradients or weight summaries to the cloud. The cloud aggregates these updates to generate new global model versions. New models are distributed to edge nodes via the image repository, forming a closed loop of "inference–training–release–reinference." Meanwhile, low-confidence samples and misclassification information accumulated at the edge are fed back to the cloud, serving as key data for the next training round, thereby accelerating model evolution and improving adaptability to complex scenarios. Technically, the architecture relies on cloud-native systems such as Kubernetes and KubeEdge as unified orchestration and edge management platforms, leveraging image repositories for cross-region storage and distribution, and adopting rolling upgrades and canary release strategies for smooth iteration. Regarding resource allocation, the system sets lower bounds and priorities separately for training and inference, while dynamically determining task offloading and collaborative triggers through threshold functions that account for edge load, cloud queuing, network availability, and target accuracy. At the deployment and operations level, small-batch iterations and partitioned canary releases mitigate model update risks, while observability metrics provide monitoring and optimization across the entire pipeline.

The proposed integrated training–inference task offloading method and implementation architecture unify training, inference, and resource optimization, forming a dynamic balance and continuous optimization mechanism between cloud and edge. It not only inherits the engineering advantages of containerization and declarative orchestration, but also incorporates federated learning and collaborative strategies to endow the system with adaptive evolution capabilities, thereby offering a scalable, observable, and continuously optimized implementation pathway for cloud–edge collaborative image recognition.

4. Cloud–Edge Collaborative Inference Optimization and Federated Aggregation Methods

4.1. Collaborative Inference Migration Strategy and Implementation for Resource-Constrained Scenarios

The essence of collaborative inference migration lies in deciding when to remain at the edge and when to offload to the cloud. To this end, the system introduces three lightweight decision signals at key stages of request arrival and inference execution: network-side reachability and transmission time, edge-side resource load thresholds, and model-side inference confidence thresholds. On the network side, the system estimates the minimum achievable transmission time for a unit task and compares it with the actual transmission time to determine whether conditions for data uplink are met. Building on this, it then observes whether edge resources such as CPU and memory have exceeded load limits.

If the threshold is exceeded and the network is available, tasks are migrated to the cloud; if resources are constrained but the network is unavailable, the system remains in a listening state until either "resource release," in which case local inference continues, or "network availability," in which case migration is triggered. This two-stage decision process effectively avoids the queuing and rollback costs of blind cloud offloading.

On the model side, after the lightweight edge model produces an initial prediction, the system extracts the maximum class probability of each image as a confidence indicator and compares it with a minimum tolerance threshold. Samples falling below this threshold are regarded as "hard cases" and are automatically sent to the cloud for secondary recognition by a high-capacity model. In this way, while maintaining the overall latency budget, the system significantly reduces misclassifications and missed detections at the edge. The "low-confidence trigger—cloud verification" strategy enables the edge to handle only high-certainty subsets, while the cloud concentrates on complex samples, forming a clearly defined inference pipeline.

This three-signal triggering mechanism not only provides criteria for deciding whether to migrate or retain tasks, but also directly shapes runtime resource usage curves and total completion time. Comparative experiments against the default joint inference strategy of Sedna show that the proposed trigger–migration mechanism achieves a better balance in total completion time, edge completion time, and average resource utilization. The results demonstrate that enabling the edge to "do less but faster" while letting the cloud conduct "hard case verification" contributes to overall performance improvement.

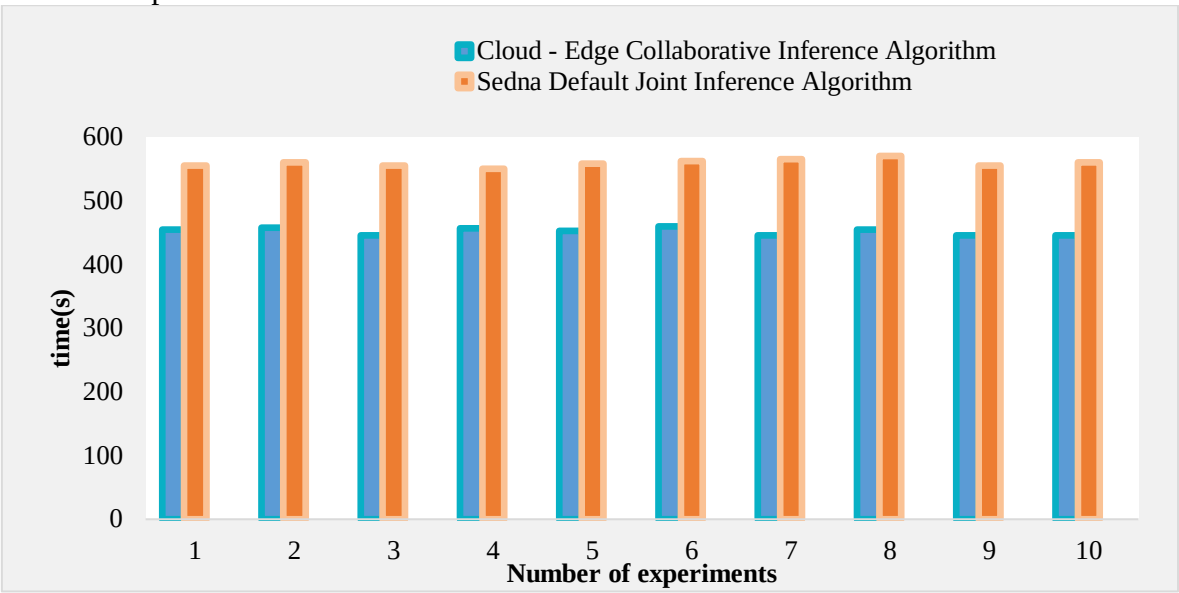

Figure 2. Comparison of Total Inference Task Completion Time

As shown in Figure 2, in the medical pathology image recognition task, the total task completion time of the proposed cloud–edge collaborative inference algorithm is significantly and consistently lower than that of Sedna's default joint inference algorithm. This experimental result directly verifies the above conclusion and demonstrates that the dual-trigger collaborative migration strategy designed in this paper can effectively optimize system resource utilization and greatly reduce overall task execution time.

To implement the strategy as a runnable system, the platform layer provides unified orchestration and observability support. Built on top of Kubernetes, KubeEdge extends edge nodes with authentication, offloading, and load-balancing capabilities, establishing a dedicated communication

channel for cloud–edge information exchange. This enables the cloud scheduler to accurately offload inference containers to designated edge nodes based on nodeSelector/affinity configurations, while lightweight message buses (e.g., MQTT) are used to carry trigger signals and status heartbeats. Combined with Sedna's joint inference and continual learning primitives, the process of "resource threshold—network judgment—confidence triggering—migration execution" can be encapsulated as configurable operators that are orchestrated and reused without modifying business code.

In integration with federated aggregation, the collaborative inference stage automatically builds a "hard-case cache," including features and inference metadata of low-confidence samples. These records, without containing original private data, are abstracted and used for sampling and weighting in the next round of aggregation, giving priority to improving the fitting degree in heterogeneous and sparse edge distributions. After aggregation, the cloud releases updated models in versioned image form, which are smoothly replaced at the edge using rolling or canary strategies. In this way, the "hard-case gains from migration" are closed-looped back into "improved inference performance in the next round." This approach is consistent with prior studies, which show that adjusting the aggregation contribution of local models can enhance global accuracy under imbalanced data, while collaborative inference triggering increases the probability that such samples are selected.

In summary, the proposed collaborative inference migration strategy for resource-constrained scenarios builds on threefold triggers—network judgment, resource constraints, and confidence gating—uses cloud–edge native platforms as the execution base, and federated aggregation as the long-term evolution mechanism. This forms a dynamic optimal behavior of "offload to the cloud when feasible, remain at the edge when sufficient." Within an engineering system that ensures observability and rollback, the strategy not only guarantees latency targets at the edge but also leverages cloud verification and federated updates to safeguard accuracy and robustness in complex long-tail scenarios.

4.2. Federated Learning Aggregation Optimization Algorithm for Imbalanced Samples and Experiments

To address aggregation bias caused by heterogeneous data distributions among participants, this section designs a "quality-and-scale collaborative weighting" aggregation method within the "training–inference collaboration" framework. While retaining the influence of sample size, the method introduces model stability and feature recognition capability to evaluate the contribution of each local model, thereby generating dynamic aggregation weights to mitigate the impact of imbalanced distributions on the global model. This approach targets the limitation of traditional FedAvg, which tends to favor nodes with larger datasets under imbalance, and emphasizes a fair aggregation that balances both quantity and quality.

On the cloud side, the algorithm uses a small class-balanced public validation set to evaluate the recognition performance of each edge node's local model across different feature categories, producing a per-class accuracy list. From this, the average accuracy and standard deviation across features are calculated, where the standard deviation reflects stability. Together with the local training sample size (reflecting scale), these factors are normalized and combined to form the contribution weight of each node. This process avoids transmitting raw data and can be completed without altering the local training strategies of participants.

In weight construction, let the stability weight be W_i(higher when more stable), the maximum-class accuracy weight be U_i (reflecting the model's best recognition ability for key features), and the sample-size weight be B_i. After range normalization, the three factors are combined using a

power-normalization scheme: $V_i = \mathrm{norm}(W_i^{\alpha} \cdot U_i^{\beta} \cdot B_i^{\gamma})$, where α, β, γ are platform-defined hyperparameters set according to the scenario to balance fairness and convergence speed. The resulting V_i serves as the aggregation coefficient, applied to the weighted summation of local models and parameters to obtain the new global model and parameters: $G = \sum_i \tilde{V}_i M_i, P = \sum_i \tilde{V}_i p_i$. . The updated global model is then distributed back to edge nodes for continued local training until convergence is achieved.

From an implementation perspective, the algorithm is integrated into the cloud–edge collaboration platform in a plugin-based manner. Within Sedna/Kubernetes, three atomic steps—public-set evaluation, weight generation, and weighted aggregation—are inserted into the existing training–aggregation–distribution loop. This enables seamless integration with task offloading, resource orchestration, and inference services, ensuring the integrity and maintainability of the training–inference pipeline. The aggregation plugin automatically triggers evaluation and weight updates after each training round, allowing the aggregation strategy to adapt dynamically to evolving data and models.

Experiments are designed using pathological image recognition as a case study, with multiple control groups. Several sample allocation schemes with different imbalance ratios and heterogeneous distributions are constructed. A unified initial model and training program are distributed from the cloud, after which participants perform independent local training. On the cloud side, two aggregation processes are executed: the proposed optimized aggregation and Sedna's default FedAvg. Evaluation metrics include global average accuracy, macro-average F1 score, and recall of the weakest class, thereby assessing both overall performance and inter-class fairness. Weight ablation studies are also conducted to examine the effect of α, β, γ. Data allocation and comparison procedures follow the same basic settings as prior studies, covering multi-level imbalance and cross-participant heterogeneity scenarios.

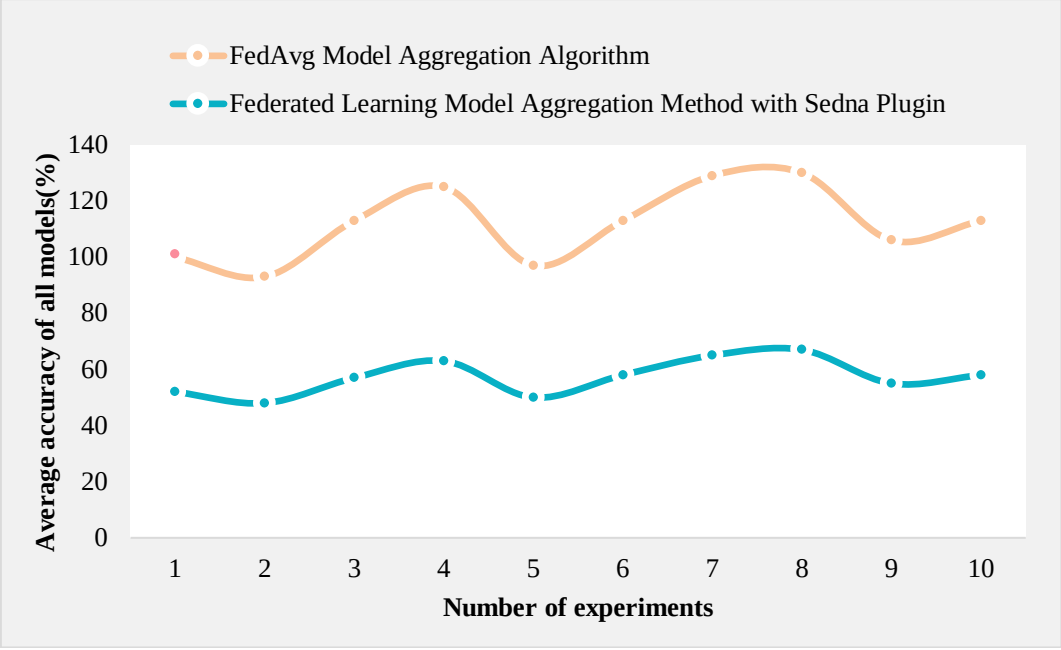

Figure 3: Accuracy Comparison between Optimized Aggregation Method and FedAvg

Results show that under various imbalance settings, the proposed optimized aggregation method significantly increases the effective contribution of high-quality local models, reduces the negative

impact of sample heterogeneity on aggregation, and achieves higher global accuracy than default FedAvg, as illustrated in Figure 3.This trend remains consistent across different experimental groups. Compared with the platform’s default aggregation, the proposed method takes training sample scale into account while placing greater emphasis on model quality and stability, thereby demonstrating stronger robustness in macro-average metrics and tail classes.

In summary, based on the “stability–capability–scale” collaborative weighting principle, the aggregation optimization mechanism proposed in this section can adaptively correct imbalance through lightweight cloud-side evaluation and weighting, without altering the training and inference processes at the edge. Implemented in a pluginized manner within the “training–inference collaboration” framework, it has been stably deployed, and comparative experiments verify its superiority over default FedAvg in aggregation performance.

5. Conclusion and Outlook

This paper proposes a “cloud–edge collaborative image recognition task offloading framework driven by federated learning for training–inference collaboration and resource optimization.” Targeting the problems of transmission latency, limited computing power, and sample imbalance in traditional paradigms, we conducted research and implementation along three dimensions: integrated training–inference, collaborative inference, and federated aggregation. Specifically, a task–resource–data relationship model for cloud–edge collaboration was constructed, and a closed-loop workflow of training, image distribution, and edge inference was realized through containerization and declarative orchestration. A dual-trigger collaborative inference migration strategy based on load and confidence was designed to effectively balance latency and accuracy. Furthermore, a dynamic weighted aggregation method for imbalanced samples was proposed to enhance the contribution of high-quality local models in the global model. Experimental results show that the framework significantly reduces latency and improves recognition accuracy in both medical and marine image recognition scenarios.

Looking ahead, several directions remain to be further explored. At the task offloading level, multi-objective constraints such as energy consumption, cost, and security can be introduced. In collaborative inference, intelligent prediction and adaptive scheduling can be incorporated to better suit large-scale dynamic environments. In federated aggregation, personalized aggregation and model distillation can be explored to enhance convergence and generalization across heterogeneous nodes. In terms of engineering applications, mechanisms for cross-domain collaboration and privacy protection should be developed to meet real-world business requirements. Overall, the proposed framework provides a scalable pathway for efficient image recognition in cloud–edge environments and is expected to evolve into multimodal and cross-industry intelligent solutions with the advancement of infrastructure and algorithms.

References

[1] *Zhang C, Xie Y, Bai H, et al. A survey on federated learning[J]. Knowledge-Based Systems, 2021, 216: 106775.*

[2] *Li, W. (2025). Discussion on Using Blockchain Technology to Improve Audit Efficiency and Financial Transparency. Economics and Management Innovation, 2(4), 72-79.*

[3] *Chen H Y, Chao W L. On bridging generic and personalized federated learning for image classification[J]. arXiv preprint arXiv:2107.00778, 2021.*

[4] Tang X, Wu X, Bao W. Intelligent Prediction-Inventory-Scheduling Closed-Loop Nearshore Supply Chain Decision System[J]. Advances in Management and Intelligent Technologies, 2025, 1(4).

[5] Yang D, Liu X. Collaborative Algorithm for User Trust and Data Security Based on Blockchain and Machine Learning[J]. Procedia Computer Science, 2025, 262: 757-765.

[6] Xu, H. (2025). Optimization of Packaging Procurement and Supplier Strategy in Global Supply Chain. European Journal of Business, Economics & Management, 1(3), 111-117.

[7] Huang, J. (2025). Balance Model of Resource Management and Customer Service Availability in Cloud Computing Platform. Economics and Management Innovation, 2(4), 39-45.

[8] Gao Y. Research on Risk Identification in Legal Due Diligence and Response Strategies in Cross border Mergers and Acquisitions Transactions [J]. Socio-Economic Statistics Research, 2025, 6(2): 71-78.

[9] Li W. Building a Credit Risk Data Management and Analysis System for Financial Markets Based on Blockchain Data Storage and Encryption Technology[C]//2025 3rd International Conference on Data Science and Network Security (ICDSNS). IEEE, 2025: 1-7.

[10] Jing, X. (2025). Research on the Application of Machine Learning in the Pricing of Cash Deposit Products. European Journal of Business, Economics & Management, 1(2), 150-157.

[11] Huang, J. (2025). Research on Resource Prediction and Load Balancing Strategies Based on Big Data in Cloud Computing Platform. Artificial Intelligence and Digital Technology, 2(1), 49-55.

[12] Truong T. The Research on the Application of Blockchain Technology in the Security of Digital Healthcare Data [J]. International Journal of Health and Pharmaceutical Medicine, 2025, 5(1): 32-42.

[13] Xu Q. Design and Future Trends of Intelligent Notification Systems in Enterprise-Level Applications[J]. Economics and Management Innovation, 2025, 2(3): 88-94.

[14] Wu X, Bao W. Research on the Design of a Blockchain Logistics Information Platform Based on Reputation Proof Consensus Algorithm[J]. Procedia Computer Science, 2025, 262: 973-981.

[15] Wu, H. (2025). The Commercialization Path of Large Language Models in Start-Ups. European Journal of Business, Economics & Management, 1(3), 38-44.