

로봇 클러스터링 프레임워크 개발을 통한 자율주행로봇에서의 연합 학습 실증

구세완*, 김영재**

Empirical Study of Federated Learning in Autonomous Robots through the Development of a Robot Clustering Framework

Sewan Gu*, Youngjae Kim**

이 논문은 2026년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2024-00346798, 일상생활 공간에서 자율행동체의 복합작업 성공률 향상을 위한 자율행동체 엣지 AI SW 기술 개발)

요 약

본 논문은 다수 로봇을 군집화(Clustering)하여 관리하고 모니터링하기 위한 클러스터링 프레임워크(Robot clustering framework)를 제안하고, RCF기반에서 연합 학습과 추론엔진 배포를 자동화 하는 FedOps를 구현한다. RCF는 쿠버네티스(Kubernetes) 기반 시스템에서 gRPC프로토콜 기반의 다수 로봇을 클러스터링한다. RCF 기반의 원격 관리 기능을 통해 호스트에서 FedOps에 필요한 소프트웨어를 자동 배포하고 업데이트할 수 있다. 실험을 통한 검증 을 위해서 연합 학습에 플라워 프레임워크(Flower framework)를 활용하며, 관련 클라이언트와 서버를 로봇과 엣지 서버에 배포하여 글로벌 모델을 생성하고 업데이트된 추론 모델을 로봇에 배포한다. 제안된 방법을 로봇과 엣지 및 클라우드 서버등으로 구성된 실험 환경에서 검증한다.

Abstract

This paper proposes a Robot Clustering Framework (RCF) for managing and monitoring multiple robots through clustering, and implements FedOps, which automates federated learning and inference engine deployment within the RCF. RCF clusters multiple robots in a Kubernetes-based system using the gRPC protocol. Through RCF's remote management capabilities, software required for FedOps can be automatically deployed and updated from the host. For experimental validation, the Flower Framework is utilized for federated learning, deploying relevant clients and servers on robots and edge servers to create a global model and distribute the updated inference model to the robots. The proposed method is validated in an experimental environment consisting of robots, edge servers, and cloud servers.

Keywords

robot clustering framework, federated learning, flower framework, FedOps, Jenkins, docker, gRPC, yolo

* LG전자 책임연구원
- ORCID: <https://orcid.org/0000-0001-9547-7626>
** LG전자 수석연구위원 상무(교신저자)
- ORCID: <https://orcid.org/0000-0002-9903-5325>

· Received: Jan. 09, 2026, Revised: Mar. 23, 2026, Accepted: Mar. 26, 2026
· Corresponding Author: Youngjae Kim
LG Electronics, 10 Magokjungang 10-ro,, Gangseo-gu, Seoul 07796, Korea
Tel.: +82-2-2033-1114, Email: youngjae.kim@lge.com

1. 서 론

연합 학습(FL, Federated Learning)은 각 로컬 컴퓨터에서 수집된 원시 데이터를 인공지능 모델 학습을 위해 서버로 직접 송부하지 않고, 각 개발 디바이스에서 직접 모델을 학습시킨 후 그 파라미터(Parameter)만을 서버로 전송하여 글로벌 모델을 구축하는 학습 방법으로 B. McMahan et al.[1]이 처음 제안하였다. 이때 서버와 약속된 주기로 학습 중인 모델의 파라미터를 송부하고 중앙의 서버가 수신한 각 로컬 컴퓨터에서 학습된 파라미터들을 수집 및 연산하여, 최종적인 글로벌 모델을 생성한다. 특히 이러한 연합 학습의 고유한 프라이버시 보호 이점은 의료 분야 등에서 두드러진다. 일례로 A. Chaddad et al.[2]의 연구에서는 데이터의 외부 유출이 엄격히 금지되는 규제와 법률적 제약 하에서도, 환자 진단 데이터를 헬스케어 인공지능 플랫폼에 안전하게 활용할 수 있는 연합 학습의 구체적인 적용 사례를 요약하여 제시한 바 있다. 또한, S. Hong and K. Lee[3]는 연합 학습 시스템을 실제 디바이스 환경에 성공적으로 운영하기 위해서는 연합 학습의 전체 수명주기 관리, 코드 버전 관리, 모델 서빙(Model serving) 그리고 디바이스 모니터링 기능을 포괄하는 MLOps 기반의 개발 환경 설계가 필수적임을 주장한다.

한편, 연합 학습을 실제 서비스 환경에 실시간으로 적용하기 위해서는 분산된 연산 노드들을 효율적으로 관리할 수 있는 다음의 조건들이 충족되어야 한다. 첫째, 각 로컬에서 사용하는 인공지능 학습모델, 추론 모델을 원격에서 배포 가능 해야 하고, 둘째, 로컬에서 수집된 데이터를 자동으로 엡지 서버에 송부할 수 있는 기능이 요구된다. 해당 자동화 과정이 부재할 경우, 관리자가 직접 학습 노드에 방문하여 수작업으로 데이터를 이관해야 하는 비효율이 발생한다. 셋째는 데이터 수집, 학습, 추론 모델 배포를 담당하는 소프트웨어 모듈들을 로봇, 엡지 서버, 클라우드 서버 전반에 걸쳐 자동으로 배포 및 관리하는 파이프라인과 이를 지원하는 통합 프레임워크가 필요하다.

이를 위해 본 논문은 원격의 다수 로봇을 하나의

군집 시스템으로 운용할 수 있는 로봇 클러스터링 프레임워크(RCF, Robot Clustering Framework)를 제안한다. 제안하는 RCF 환경에서 다수의 로봇은 연합 학습 클러스터의 노드로서 참여하며, 서비스 추론뿐만 아니라 지속적인 모델 학습을 위한 데이터 수집 역할을 동시에 수행한다. 이 과정에서 본 연구는 Moon 등[4]의 연구에서 제안된 FedOps(Federated Learning Operations) 파이프라인 개념을 차용하여, 전체 연합 학습 과정에서 요구되는 원격 로봇, 엡지 서버, 클라우드 서버의 소프트웨어(SW) 빌드, 배포 및 업데이트 전 과정을 하나의 자동화된 파이프라인으로 구성하였다. 단, 본 플랫폼의 인지 성능 고도화를 위해 로봇으로부터 수집된 데이터를 연합 학습 라벨링에 활용하는 단계는 인간의 개입을 통한 수작업으로 진행한다. 또한, 본 연구의 RCF는 각 로봇이 독립적인 쿠버네티스 기반에서 운영이 되도록 구성되며, 로봇 내부에서 구축된 쿠버네티스를 gRPC프로토콜을 기반으로 클러스터링을 수행한다. gRPC 공식 기술 문서[5]와 S. Ritu et al.[6]에 명시된 바와 같이, gRPC는 HTTP/2 기반의 높은 통신 효율을 지닌 양방향 스트리밍과 이진 직렬화(Binary serialization) 기능을 지원한다. 따라서 이를 활용하면 메모리 및 통신 대역폭이 제한적인 로봇 환경에서도 낮은 지연 시간(Low latency)과 높은 데이터 처리 성능을 보장받을 수 있어 실시간 제어에 유리하다. 이때, 본 논문에서 제안하는 RCF 클러스터가 기존의 마스터-슬레이브 구조의 쿠버네티스 클러스터와 다른 주요 차이점은 ‘독립된 마스터 쿠버네티스들의 연합 클러스터’라는 점이다. 일반적인 쿠버네티스 노드로 구성된 단일 네트워크 클러스터는 대면적을 이동하는 다수 로봇 환경에서 컨테이너 네트워크 인터페이스(CNI, Container Network Interface) 오류가 빈번하게 발생할 수 있으나, 단일 마스터를 묶은 RCF 단위는 이러한 단절 오류를 구조적으로 방지할 수 있다. 이를 운영하기 위해 RCF를 구성하는 노드 중 호스트 노드가 원격으로 로봇과 엡지 서버를 통합 관리하고 소프트웨어 업데이트를 통제한다. 이때 기능별 실행 소프트웨어는 개별 로봇 내 환경에서 원활히 동작하도록 사전에 도커(Docker) 이미지 형태로 준비된다.

마지막으로, 본 연구는 연합 학습의 실행을 위한 오픈소스 프레임워크로 D. J. Beutel et al.[7]이 개발한 플라워 프레임워크(Flower framework)를 활용한다. J. Kim et al.[8]의 연합 학습 플랫폼 비교 연구 및 V. Pais et al.[9]의 적용 사례 연구 등에서 요약 검증된 바와 같이, 플라워 프레임워크는 유연한 클라이언트-서버 구조와 뛰어난 확장성 덕분에 범용적인 연합 학습 구현에 높은 경쟁력을 갖춘 것으로 평가받는다. 본 연구에서는 제안하는 RCF를 통해 플라워 프레임워크의 클라이언트와 서버를 개별 로봇과 엣지 서버에 각각 배포함으로써 연합 학습 환경을 완성한다. 본 논문의 환경에서 로봇은 독자적인 학습을 수행하는 노드가 되며, 엣지 서버는 수집된 파라미터들을 취합하여 글로벌 모델을 생성한다. 성능 실증을 위해 본 논문은 로봇 주행 중 시각 엔진이 미인식 객체로 판단한 물체의 데이터를 연합 학습으로 재학습시킨 후, 갱신된 지능형 추론 모델을 RCF를 통해 즉각적으로 각 로봇에 자동 배포하는 FedOps 시나리오를 구성하고 실험을 통해 그 실효성을 검증한다.

논문의 구성은 다음과 같다. 2장에서 RCF의 세부 설계 및 구현 방법에 대해 기술하고, 연합 학습 적용과 FedOps 구현 방법에 대하여 논한다. 3장에서는 핵심 실험을 통한 검증 결과를 제시하며, 4장에서 결론을 맺는다.

II. 로봇 클러스터링 프레임워크(RCF) 설계 및 구현과 연합 학습에 적용

2.1 로봇 클러스터링 프레임워크(RCF) 설계 및 구현

RCF는 다수의 로봇을 체계적으로 관리하고 모니터링하기 위한 기반이 되는 미들웨어 프레임워크로서, 기존 연합학습에서 활용하는 쿠버네티스에서의 CNI 네트워크 오류를 줄이며, 독립적으로 각 클러스터 노드를 관리할 수 있도록 설계된다. 네이티브 쿠버네티스를 사용하는 경우의 한계와 문제 정의의 기술하고, 이후 차례로 RCF 설계 원칙과 구성요소 및 운영상의 이점과 검증 흐름도를 살펴본다.

2.1.1 기존 방법의 한계 및 문제 정의

기존의 로봇 군집 관리 및 배포 방식은 단일 클러스터 내에서 중앙 집중식 쿠버네티스를 활용하거나 단순 배포 스크립트에 의존하는 경우가 많다. 그러나 다수의 로봇이 참여하는 연합 학습(FL) 실증 환경에서는 다음과 같은 기술적 한계가 존재한다:

- 네트워크 불안정성: 로봇이 이동하며 참여하는 환경에서 표준 CNI(Container Network Interface) 기반의 네트워크는 잦은 연결 오류를 발생시키며, 특정 노드의 오류가 전체 클러스터로 전파될 위험이 있다.
- 복잡한 자원 및 배치 관리: 노드별로 이질적인 데이터 수집, 학습 및 추론 모듈을 배포하고 전체 라이프사이클을 관리하기에 기존의 정적인 배포 방식은 유연성이 부족하다.

2.1.2 RCF 설계 원칙 및 구성 요소

RCF(Robot Clustering Framework)는 위와 같은 문제를 해결하기 위해 '독립적 클러스터 노드 관리'와 '고성능 원격 제어'를 핵심 설계 원칙으로 삼는다. RCF의 주요 구성 요소는 다음과 같으며, 이에 대한 내용을 그림 1의 RCF설계 구성도에 나타낸다.

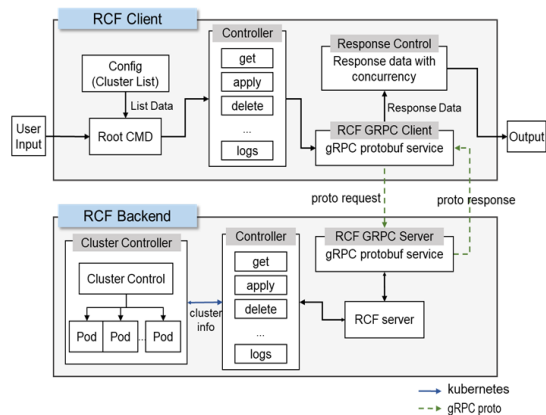


그림 1. RCF설계 구성도
Fig. 1. RCF design block diagram

- RCF Backend(Custom controller): 각 로봇 및 엣지 노드에 독립적으로 설치되어 운영된다. 이는 해

당 노드의 쿠버네티스 리소스를 직접 제어하며, 타 노드의 장애로부터 격리된 안정적인 운영 환경을 보장한다.

- RCF Client(Host node): 호스트 노드에서 여러 대의 분산된 로봇을 gRPC 기반으로 통합 관리한다. 사용자의 요청(모듈 배포, 상태 조회 등)을 백엔드로 효율적으로 중계한다.

- gRPC 기반 통신 레이어: 마이크로서비스 아키텍처에서 고성능을 발휘하는 gRPC와 Protobuf를 적용하여, 대역폭이 제한되거나 불안정한 네트워크 환경에서도 데이터 전송 신뢰성을 확보할 수 있다.

2.1.3 운영상 이점 및 검증 흐름

RCF의 도입은 연합 학습 운영(FedOps) 관점에서 다음과 같은 이점을 제공한다:

- 장애 격리 및 안정성(Robustness): 각 노드가 독립적인 쿠버네티스를 유지하므로, 네트워크 단절이나 특정 노드의 결함이 발생하더라도 전체 시스템의 가용성에 영향을 주지 않는 SPOF(Single Point of Failure) 방지 설계가 구현된다.

- FedOps 효율화: 데이터 수집 에이전트, 연합 학습 클라이언트 및 서버(Flower), 최종 추론 엔진(YOLOv11)으로 이어지는 복잡한 파이프라인을 원격에서 통합 배포함으로써 운영 리소스를 획기적으로 절감한다.

- 기여도 및 검증: 본 프레임워크는 기존 배포 방식이 해결하지 못했던 'FL 전용 운영 복잡성'을 해결함으로써, [데이터 수집 → 로컬 학습 → 글로벌 모델 갱신 → 배포]라는 선순환 구조를 자율주행 로봇 환경에서 실현하는데 핵심적인 기반을 제공한다.

다음 절에서는 실제 RCF 기반으로 연합학습을 실행하기 위한 내용을 기술하며, 이를 통해서 내부적으로 RCF의 동작과 연합 학습을 위한 모듈 배포 과정을 설명한다.

2.2 RCF 기반의 연합 학습

RCF는 연합 학습을 실행하기 위한 모듈들을 배포

하고 관리하기 위한 것으로 다음과 같은 역할을 수행한다. 첫째, 연합 학습에 참여하는 서버, 엣지 서버, 로봇들을 노드로서 하나의 네트워크로 연결하는 역할이며, 이를 클러스터링이라 표현한다. 둘째, 클러스터링내에 노드들에 연합 학습 모듈들을 배포하는 역할을 한다. 뒤에 이어서 설명할 플라워 프레임워크의 클라이언트와 서버 등이 대상이다. 셋째, 연합 학습이 완료된 후 추론 모델을 배포하고, 배포된 로봇의 지능을 원격으로 관리하는 역할을 한다. 넷째는 학습 데이터를 수집하는 수집 모듈을 배포하는 역할을 하는데, 해당 모듈은 각 로봇이 수집한 데이터를 엣지 서버에 송신하여 수작업으로 라벨링이 되도록 한다. 호스트 노드에서 레이블을 하기 위한 프로그램을 각 엣지 서버에 원격으로 배포 한다.

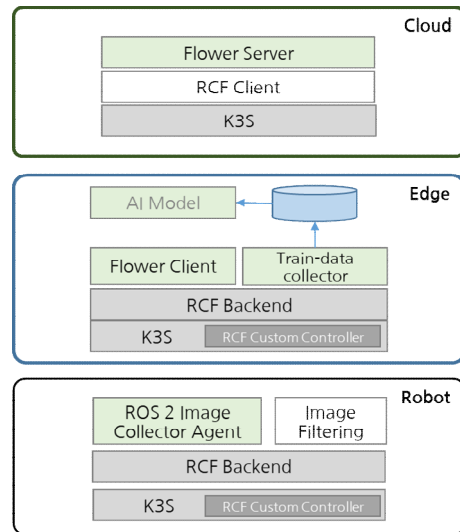


그림 2. RCF기반 연합 학습 구성도
Fig. 2. RCF-based federated learning block diagram

RCF기반 분산 협업 연합 학습 구성도를 그림 2에 나타내고 있는데, 쿠버네티스는 로봇의 하드웨어를 고려하여 경량 버전인 K3S를 설치하며, 앞선 그림 1에서 설명한 Custom Controller를 RCF Custom Controller 라는 이름으로 설치한다. RCF Custom Controller는 각종 모듈과 엔진들이 쿠버네티스 위에서 실행되면서 쿠버네티스에서 실행되는 로봇 관련된 모든 리소스에 대한 조회, 삭제, 갱신등 필요한 요구사항을 수용하기 위한 역할을 한다. 기본적인

쿠버네티스에서 지원하지 않는 리소스들을 관리하는 기능들이 이에 해당한다. 필요한 리소스와 기능에 대한 요청은 사용자가 RCF Client를 통해 요청하게 되고, RCF Backend가 요청을 받아서 RCF Client에 응답을 하게된다. 객체 인식을 위한 모델을 업데이트하기 위해 연합 학습을 수행하는 것을 가정하면, 다음과 같이 표 1의 절차를 통하여 이루어지는데, 이에 대한 것은 다음 절의 플라워 프레임워크를 활용하는 과정에서 함께 살펴본다.

표 1. 연합 학습을 위한 모듈 배포 과정
Table 1. Sequence of module deployment for federation learning

<ol style="list-style-type: none"> 1. Deployment of ROS 2 Image Collector and Image Filtering Engine: Distribute the ROS 2 Image Collector and Image Filtering Engine to each robot (Host Node → Robot). 2. Deployment of Training Data Collector Agent: Deploy the Training Data Collector Agent to each robot to gather recognized image data (Host Node → Robot). 3. Deployment of FL: Deploy the FL (Flower Server and Flower Client) to the Edge and Cloud servers (Host Node → Edge Server / Cloud Server). 4. Deployment of Object Detection Inference Model (YOLOv11): Distribute the YOLOv11 object detection inference model to the Edge server (Host Node → Edge Server).
--

2.3 RCF 기반의 플라워 프레임워크 연합 학습

플라워 프레임워크는 분산 처리를 기반으로 하는 연합 학습에 가장 많이 활용이 되는 오픈소스이며, 그림 3에서 기본적인 플라워 프레임워크 기반 연합 학습 구성도를 통해 설명한다. FL Server는 연합 학습을 주도하는 중앙의 서버로서 연합 학습을 수행할 모델을 각 FL Client에 송부하여 연합 학습에서 사용할 각종 설정값을 지정하고, 연합 학습을 시작하여 중간에 오차를 만족하는 지를 판단하여 추론 모델을 만드는 역할을 한다. 좀 더 자세히 살펴보면 다음과 같다. 서버는 전역 모델 파라미터, 훈련 라운드 정보 등 전역 제어 메시지를 SuperLink를 통해 SuperNode에 전달한다. SuperNode는 이 메시지를 자

신이 관리하는 개별 클라이언트 그룹에게 전파하여 로컬 훈련을 지시하며, 클라이언트들은 로컬 데이터로 모델을 업데이트한 후 업데이트된 로컬 모델 파라미터와 훈련 매트릭(손실, 정확도 등)을 SuperNode에게 다시 전달한다. SuperNode는 이러한 클라이언트 응답들을 1차적으로 취합하여 서버에 전달한다.

RCF는 이러한 연합 학습 프로세스를 자동화하기 위해 표 1과 같은 모듈 배포 절차를 거친다. 먼저 호스트 노드에서 로봇으로 이미지 수집 및 필터링 엔진을 배포하고(Step 1-2), 이어 엣지 및 클라우드 서버에 FL 서버와 클라이언트를 배치한다(Step 3). 학습이 완료되면 최종적으로 객체 인식용 추론 모델(YOLOv11)을 배포하여 실무에 적용한다(Step 4).

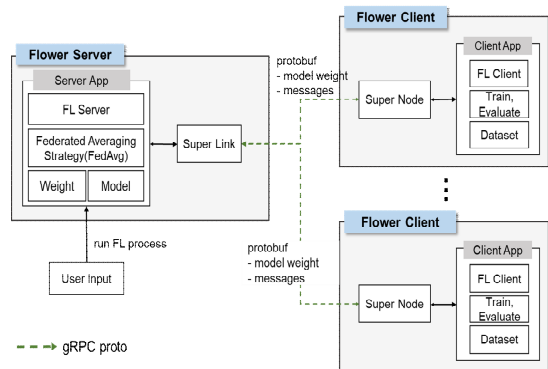


그림 3. 플라워 프레임워크 기반 연합 학습 구성도
Fig. 3. FL-based federated learning block diagram

연합 학습 서버에서 각 클라이언트로부터 수신한 모델 파라미터들을 통합하여 글로벌 모델 파라미터를 만들게 되며, 이때 사용되는 핵심 요소가 통합 전략(Strategy)이다. 플라워 프레임워크는 상황에 맞추어 선택할 수 있도록 FedAvg, Fault Tolerant FedAvg, FedProx, QFedAvg, FedOptim 등 다양한 알고리즘을 지원하며[10], 데이터의 특성, 학습 모델 특성, 하드웨어 성능을 고려하여 최적의 전략을 선택적으로 적용할 수 있다. 본 논문에서는 일반적으로 가장 단순하고 많이 사용하는 FedAvg를 적용한다.

RCF기반의 플라워 프레임워크를 설계한 구성도를 그림 4에 나타낸다.

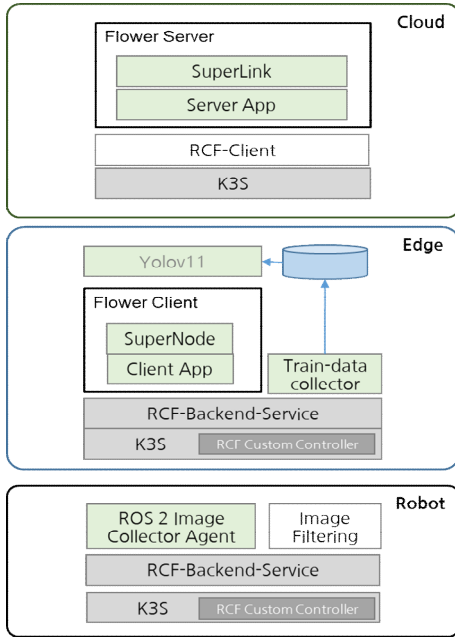


그림 4. RCF기반의 플라워 프레임워크 연합 학습 구성도
 Fig. 4. Block diagram for federated learning of FL-based on RCF

2.4 연합 학습을 위한 FedOps 구성

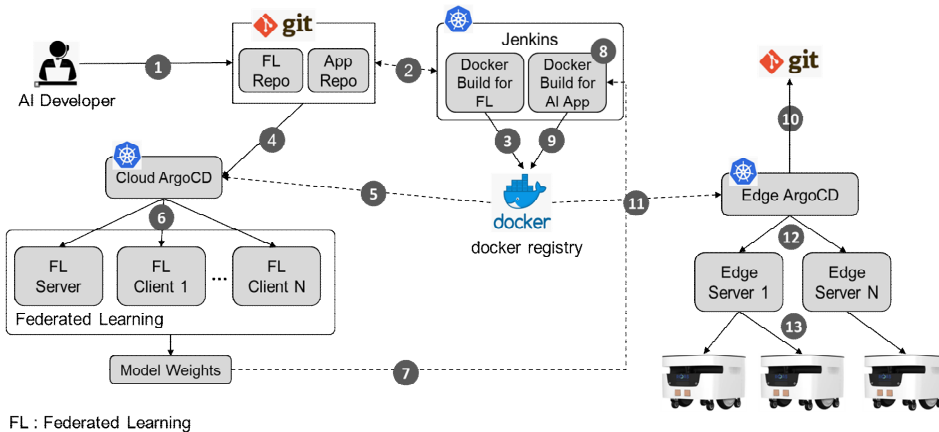
연합 학습을 통해 글로벌 모델을 제작하고, 이를 필요한 클라이언트에 자동으로 배포하는 과정을 개발부터 배포까지의 과정을 연속적으로 실행하기 위한 자동화 과정이 FedOps이다. 본 연구에서는 도커(Docker)와 쿠버네티스(Kubernetes) 기술을 결합하여 그림 5와 같은 FedOps 파이프라인을 구축하며, 해

당 과정에 대한 상세한 내용을 표 2에 순서대로 설명하는데, 학습 및 생성 단계 (①~⑦) 와 연속 배포 단계 (⑧~⑬)로 크게 구분할 수 있다.

표 2. FedOps 실행 과정

Table 2. Sequence of federated learning operations

- ① (Developer) Modify source code or model parameters.
- ② (Jenkins) Detect source code changes in Git and build Docker images for the FL program (FL Server App and Client App).
- ③ (Jenkins) Push the built Docker images to Docker Hub and update the image versions in the deployment.yaml file.
- ④ (Cloud ArgoCD) Monitor and detect updates in the deployment.yaml file from the Git repository.
- ⑤ (Cloud ArgoCD) Pull the latest FL program Docker images upon detecting changes.
- ⑥ (Cloud ArgoCD) Deploy the FL program Docker images (including the training model) to the FL cluster.
- ⑦ (FL Server) Update global model weights upon completion of the Federated Learning process.
- ⑧ (Jenkins) Verify the weight updates and build a Docker image for the AI Engine.
- ⑨ (Jenkins) Push the AI Engine Docker image to Docker Hub and update the AI Engine image version in the deployment.yaml file.
- ⑩ (Edge ArgoCD) Identify version changes in the deployment.yaml file via Git.
- ⑪ (Edge ArgoCD) Pull the updated AI Engine Docker image following version verification.
- ⑫ (Edge ArgoCD) Deploy the AI Engine Docker image to the Edge Server cluster.
- ⑬ (Edge ArgoCD) Distribute and deploy the AI Engine Docker image to the robot clusters managed by the edge servers.



FL : Federated Learning

그림 5. FedOps와 연속 배포 자동화 순서도
 Fig. 5. Sequential diagram for federated learning operations and continuous delivery

III. 실험

실험을 위한 선행 준비 작업을 위해 표 3과 같이 하드웨어와 기반 소프트웨어 설치를 미리 진행한다. 특히 쿠버네티스는 전체 기능을 사용하지 않기 때문에 K8s 대비 매우 가벼운 K3s를 사용한다. 또한,

쿠버네티스 상에서 모든 소프트웨어 모듈은 컨테이너로 실행해야 하므로 기본적인 모든 로봇 엔진과 애플리케이션은 도커 이미지를 활용하거나 만들어서 실험을 한다. 참고문헌 [11]은 플라워 프레임워크를 도커로 활용하는 방법에 대해 자세히 설명한다.

표 3. 실험에 사용된 노드의 기본 사양
Table 3. Specifications of each node for experiments

	Robot	Edge server	Cloud server
Hardware	nVidia Jetson Orin	i7-1165G7 2.80GHz (RAM: 32GB)	i7-8700K CPU 3.70GHz (RAM: 64GB)
Operating system	Ubuntu 22.04	Ubuntu 22.04	Ubuntu 22.04
Kubernetes version	K3s Standalone	K3s Standalone	K3s Standalone

RCF 기반의 연합 학습 실험을 위한 정적인 구성도를 그림 6에 간략히 표현하고 있다. 클라우드 서버와 엣지 서버는 연합 학습을 위한 RCF기반의 클러스터링을 구성하고 있으며, 엣지 서버와 로봇은 데이터를 수집하고, 나중에 추론 모델을 배포하기 위해 RCF기반의 클러스터링으로 구성된다. 로봇의 배터리만 허용한다면 엣지 서버와 로봇은 하나의 노드로 구성이 될 것이나, 본 논문의 실험에서는 이동 로봇의 배터리 용량 문제로 인해 엣지 서버와 로봇을 하드웨어적으로 분리하여 실험을 하였으며, 실제 RCF 동작을 실증하는데는 문제가 없음을 확인하였다. 실험에서 사용된 미인식 물체는 복도에 놓여진 “분리수거용 쓰레기통”과 “소화전”이며, 이는 Yolo11n에서 사전학습이 안 된 물체들이다. 미인식 물체가 있음을 확인한 후에 연합 학습을 수행하기 위한 FedOps 실험 단계는 표 2의 단계를 따르나,

FedOps 프로세스에 추가로 인간의 개입에 의한 라벨링이 필요한 경우에 대한 실험을 좀 더 간략히 정리하면 다음과 같다. 첫 번째, 먼저 연합 학습을 실행하기 위한 각 모듈들이 배포하는 단계인데, 이 단계에서는 FL Client와 기본 Yolo11n 학습 모델[12]과 라벨링 도구(Label studio)를 엣지 서버에 배포한다. 두 번째, 로봇에는 데이터 수집 모듈 (Img-collector-agent), 데이터 송신 모듈(Ros-streamer)과 Yolo11n 추론 모델을 배포한다. 세 번째, 로봇 주행을 통한 미인지 물체 이미지를 엣지 서버에 송부하고, 네 번째 단계에서는 수동으로 각 엣지 서버에 접속하여 이미 배포된 라벨링 도구를 이용하여 라벨링을 수행 한다. 다섯 번째는 실제 연합 학습을 수행하는 과정으로 라벨링된 데이터를 엣지 서버들에서 학습하고, 정해진 주기로 업데이트된 모델 파라미터를 서버에 전송한다. 서버는 각 엣지 서버에서 수신된 파라미터를 정해진 알고리즘(FedAvg)에 따라서 글로벌 모델의 파라미터를 계산하여 이를 각 엣지 서버에 전송하여 다음 학습 루프에서 사용하도록 한다. 다섯 번째 단계에서 최종 모델을 완성 이 되고 추론 모델이 만들어진다. 여섯 번째에서 Yolo11n 추론 모델을 로봇에 배포한다. 새롭게 업데이트된 추론 엔진에 의해 로봇이 이전에 인식하지 못했던 물체를 인식하는 것을 볼 수 있다.

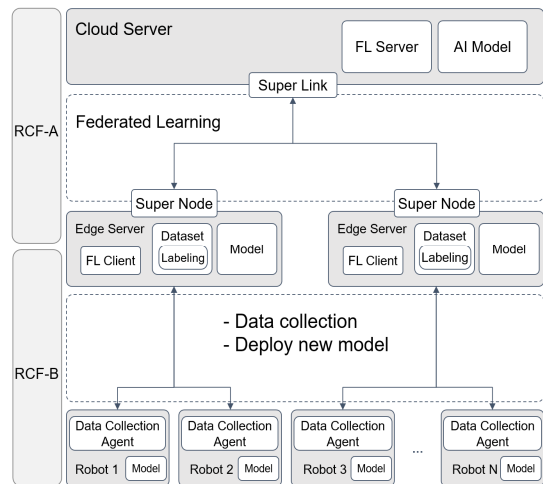


그림 6. RCF 기반의 연합 학습 실험을 위한 구성도
Fig. 6. Block diagram for RCF-based federated learning experimental

실험 결과를 그림 7에서 확인할 수 있는데, 그림 7의 (a) 및 (c)는 연합 학습 전에 물체인식이 되지 않는 상황이며, 그림 7의 (b) 및 (d)를 통해서는 연합 학습 이후 각 로봇에서 촬영된 이미지의 물체가 인식됨을 확인할 수 있다.

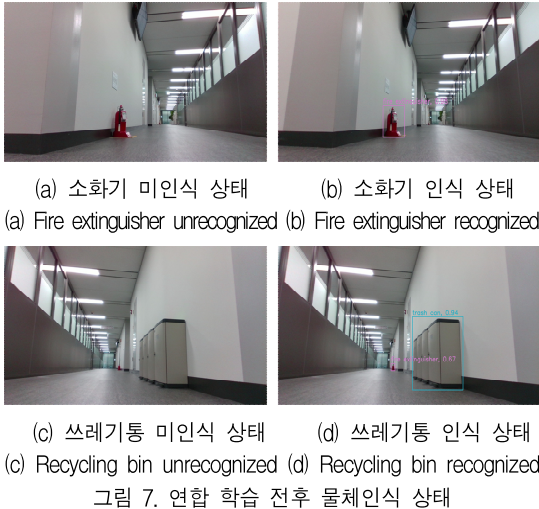


그림 7. 연합 학습 전후 물체인식 상태

Fig. 7. Object recognition status before and after federated learning

IV. 결 론

본 논문은 다수 로봇이 연합 학습에 노드로서 참여하여 지속적으로 인공지능 모델을 학습하고 자동 배포하는 FedOps 파이프라인 구성을 위한 RCF를 제안하였다. RCF는 각 로봇이 독립적인 쿠버네티스 기반에서 운영이 되도록 로봇들을 클러스터화하고, RCF의 호스트 노드에서 원격으로 로봇과 엣지 서버를 관리를 위해 모니터링하고, 이를 통해 FedOps에 필요한 로봇, 엣지 서버, 클라우드 서버의 소프트웨어 모듈을 업데이트할 수 있음을 실험하였다. 연합 학습을 수행하기 위한 프레임워크로 플라워 프레임워크를 사용하고, 플라워 프레임워크의 서버와 클라이언트를 각 엣지 서버와 클라우드 서버에 배포하여 서버에서 글로벌 모델을 만들어서 각 엣지에 새로운 모델을 배포하였다. 이때 사용한 모델은 Yolol1n 모델이었으며, 이를 실증하기 위해 로봇 주행 중에 미인식 물체로 판단되는 물체를 연합 학습을 통해 학습하고, 추가 학습된 Yolol1n 추론

모델을 자동으로 로봇에 배포하여 미인식 물체를 인식하는 전체 파이프라인을 실험으로 보였다. 제안한 프레임워크 기반의 FedOps에서 사람이 개입해야 하는 부분은 라벨링 작업인데, 이를 FedOps내에 포함하여 자동화 하는 작업은 추가 연구개발 과제가 될 것이다.

References

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", Proc. of the 20th International Conference on Artificial Intelligence and Statistics, Lauderdale, FL, USA, pp. 1273-1282, Apr. 2017. <https://doi.org/10.48550/arXiv.1602.05629>.
- [2] A. Chaddad, Y. Wu, and C. Desrosiers, "Federated Learning for Healthcare Applications", IEEE Internet of Things Journal, Vol. 11, No. 5, pp. 7339-7358, Mar. 2024. <https://doi.org/10.1109/IJOT.2023.3325822>.
- [3] S. Hong and K. Lee, "The Study on the Implementation Approach of MLOps on Federated Learning System", Journal of Internet Computing and Services, Vol. 23, No. 3, pp. 97-110, Jun. 2022. <https://doi.org/10.7472/jksii.2022.23.3.97>.
- [4] J. Moon, S. Yang, and K. Lee, "FedOps: A Platform of Federated Learning Operations With Heterogeneity Management", IEEE Access, Vol. 12, pp. 4301-4314, 2024. <https://doi.org/10.1109/ACCESS.2024.3349691>.
- [5] gRPC, <https://grpc.io/docs/what-is-grpc/introduction>. [accessed: Feb. 03, 2026]
- [6] Ritu, S. Arora, A. Bhardwaj, A. Kukkar, and S. Kaur, "A Comparative Analysis of Communication Efficiency: REST vs. gRPC in Microservice-Based Ecosystems", 2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP), Sonipat, India, pp. 621-626, May 2024. <https://doi.org/10.1109/INNOCOMP63224.2024.00107>.

- [7] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmao, and N. D. Lane, "Flower: A Friendly Federated Learning Research Framework", arXiv:2007.14390, Jul. 2020. <https://doi.org/10.48550/arXiv.2007.14390>.
- [8] J. S. Kim, S. M. Yang, K. Y. Lee, and K. K. Lee, "Advances and Issues in Federated Learning Open Platforms: A Systematic Comparison and Analysis", Journal of Internet Computing and Services, Vol. 24, No. 4, pp. 1-13, Aug. 2023. <http://dx.doi.org/10.7472/jksii.2023.24.4.1>.
- [9] V. Pais, S. Rao, B. Muniyal, and S. Yun, "Federated learning using flower framework for enhanced medication safety in intensive care units", Int. Journal of Data Science and Analytics, Vol. 20, pp. 7039-7053, Dec. 2025. <https://doi.org/10.1007/s41060-025-00877-x>.
- [10] Federated Learning Strategy of Flower Framework, <https://flower.ai/docs/framework/ref-api/flwr.server.strategy.html>. [accessed: Jan. 07, 2026]
- [11] Run Flower using Docker, <https://flower.ai/docs/framework/docker/index.html>. [accessed: Sep. 15, 2025]
- [12] Ultralytics YOLO11, <https://docs.ultralytics.com/ko/models/yolo11>. [accessed: Jan. 07, 2026]

김 영 재 (Youngjae Kim)



1999년 2월 : 서울대학교
전기공학부(학사)
2003년 5월 : Stanford Univ.
전기공학과(석사)
2007년 1월 : Stanford Univ.
전기공학과(박사)
2009년 10월 ~ 2017년 9월 : Apple
Inc. 시니어SW엔지니어
2017년 10월 ~ 2018년 11월 : Velodyne LiDAR,
수석엔지니어
2019년 1월 ~ 현재 : LG전자, HS로봇플랫폼선행연구Task
리더, 수석연구위원
관심분야 : Cloud Robotic, Robot Operating System

저자소개

구 세 완 (Sewan Gu)



1994년 2월 : 한양대학교
전자통신공학과(학사)
1996년 2월 : 한양대학교
전자통신공학과(석사)
2023년 8월 : 한양대학교
전자컴퓨터통신공학과(박사)
1997년 3월 ~2000년 3월 :

한국철도기술연구원 직책
2000년 3월 ~ 현재 : LG전자,
HS로봇플랫폼선행연구Task, 책임연구원
관심분야 : Cloud Robotic, Robot Platform,
Language-Driven-Robotics