IETF

IETF-125 Hackathon

Interface to In-Network Computing Functions (I2ICF) Project

March 15, 2026, Shenzhen

Champion: Jaehoon (Paul) Jeong

Members: Yoseop Ahn, Mose Gu, Jiwon Suh, Jisuk Chae, **Xudong Wang**, EunJin Hwang

Department of Computer Science and Engineering at SKKU

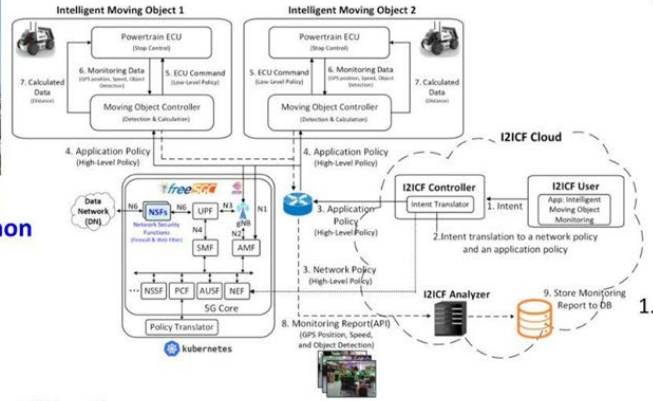
Email: {pauljeong, ahnjs124, rna0415, sjw6136, sue030124, wangxudong28, eunijhwang}@skku.edu

IETF-125 Interface to In-Network Computing Functions (I2ICF)

Champion: Jaehoon Paul Jeong (SKKU)

IETF-125 Interface to In-Network Computing Functions (I2ICF) Hackathon

I2ICF Framework



I2ICF Developing Environment

- OS: Ubuntu 20.04
- Kubernetes: Microk8s v1.32.2
- Object Detection: YOLOv8
- ROS2 Version: Humble
- GitHub Repository: https://github.com/jaehoonpauljeong/I2ICF/tree/main/IETF-125

Professors:

- Jaehoon Paul Jeong (SKKU)
- Yong-Geun Hong (DJU)
- Joo-Sang Youn (DEU)

Researchers:

- Byoungman Robert An (KETI)
- Jung-Soo Park (ETRI)
- Yunchul Choi (ETRI)

Students:

- Mose Gu (SKKU)
- Jiwon Suh (SKKU)
- Jisuk Chae (SKKU)
- Eunjin Hwang (SKKU)
- Xudong Wang (SKKU)
- Kihyun Shim (SKKU)
- Deokhyeon Ryu (DJU)
- Eunsan Seong (DEU)
- Taehyun Kwan (DEU)

Objectives

- To demonstrate an intent-driven, closed-loop safety pipeline in which a moving object receives an intent, this I2ICF framework (i) performs on-board camera-based object detection and camera–LiDAR calibration/fusion for distance estimation, (ii) streams detections to a cloud server, and (iii) executes real-time stop/detour for obstacles and pedestrians.

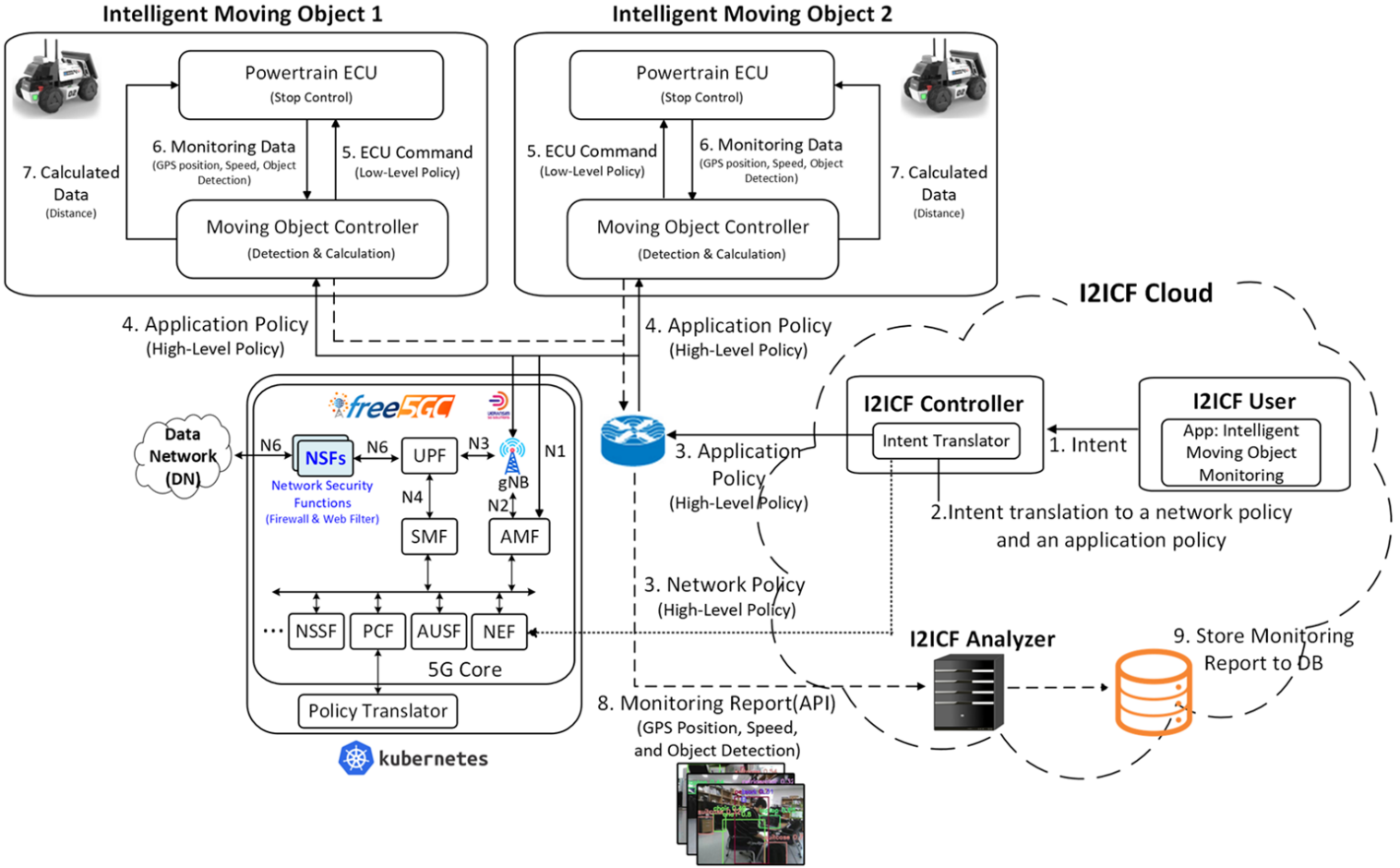
Future Work

- To optimize End-to-End performance (e.g., inference accuracy, and inference latency) and support two robot cars so that they exchange on-board inference data to cooperatively plan collision-free, optimal paths when approaching each other.

Workflow of the I2ICF Testbed

1. Intent Submission: A User provides a safety/perception intent (e.g., avoid obstacles/pedestrians and stop within a configured distance).
2. Intent Application: A Moving Object sets detection/stop parameters from the intent.
3. Calibration: Camera–LiDAR calibration and synchronization for distance estimation.
4. Perception & Fusion: Camera detection and LiDAR range fusion to classify objects and estimate distance.
5. Cloud Streaming: Detection results and motion metadata are sent to a cloud server (I2ICF Analyzer).
6. Safety Action: If a proximity threshold is met, the Moving Object stops to avoid a collision and detours for a safe driving.
7. Monitoring: Logs (e.g., JSON) are stored in the cloud database (DB) for further analysis.

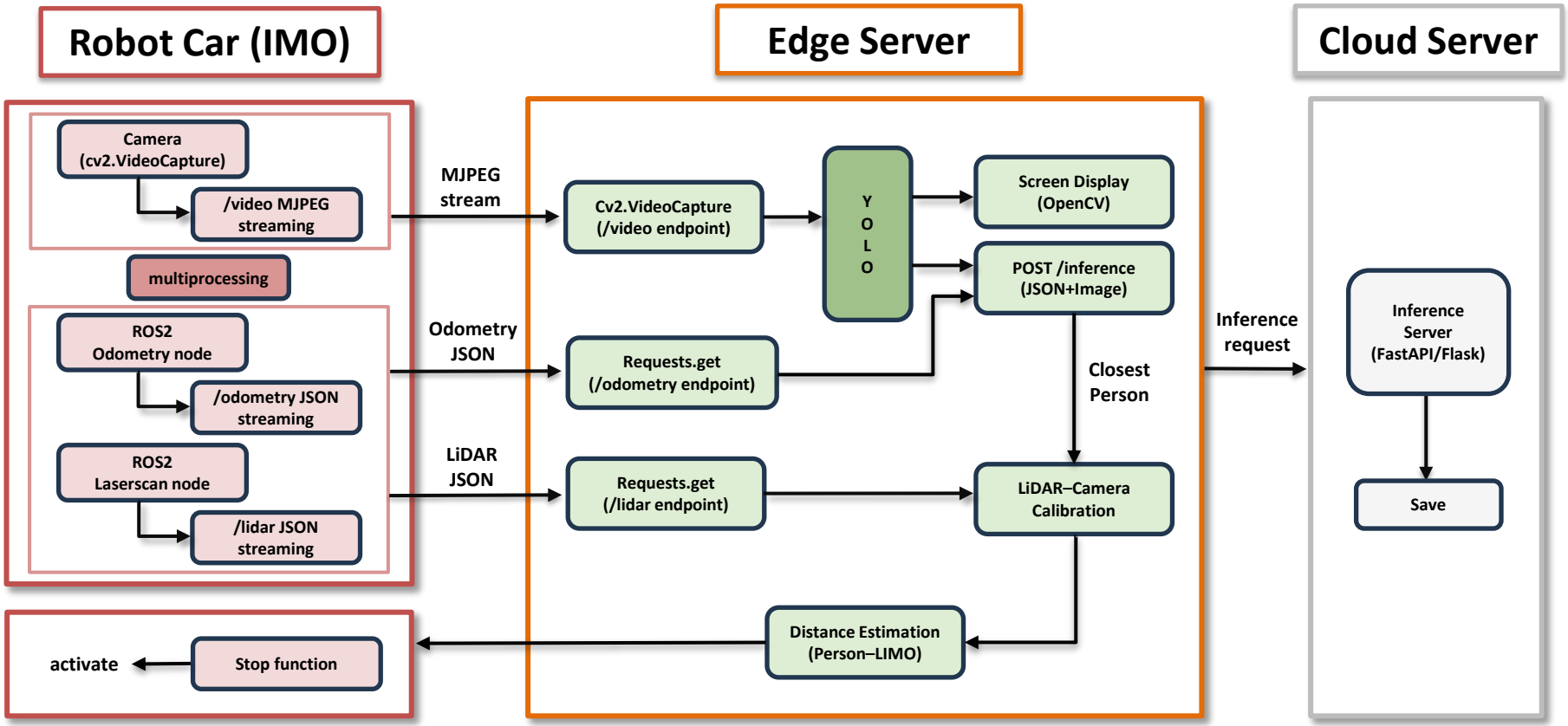
Interface to In-Network Computing Functions (I2ICF) for Mobile Objects



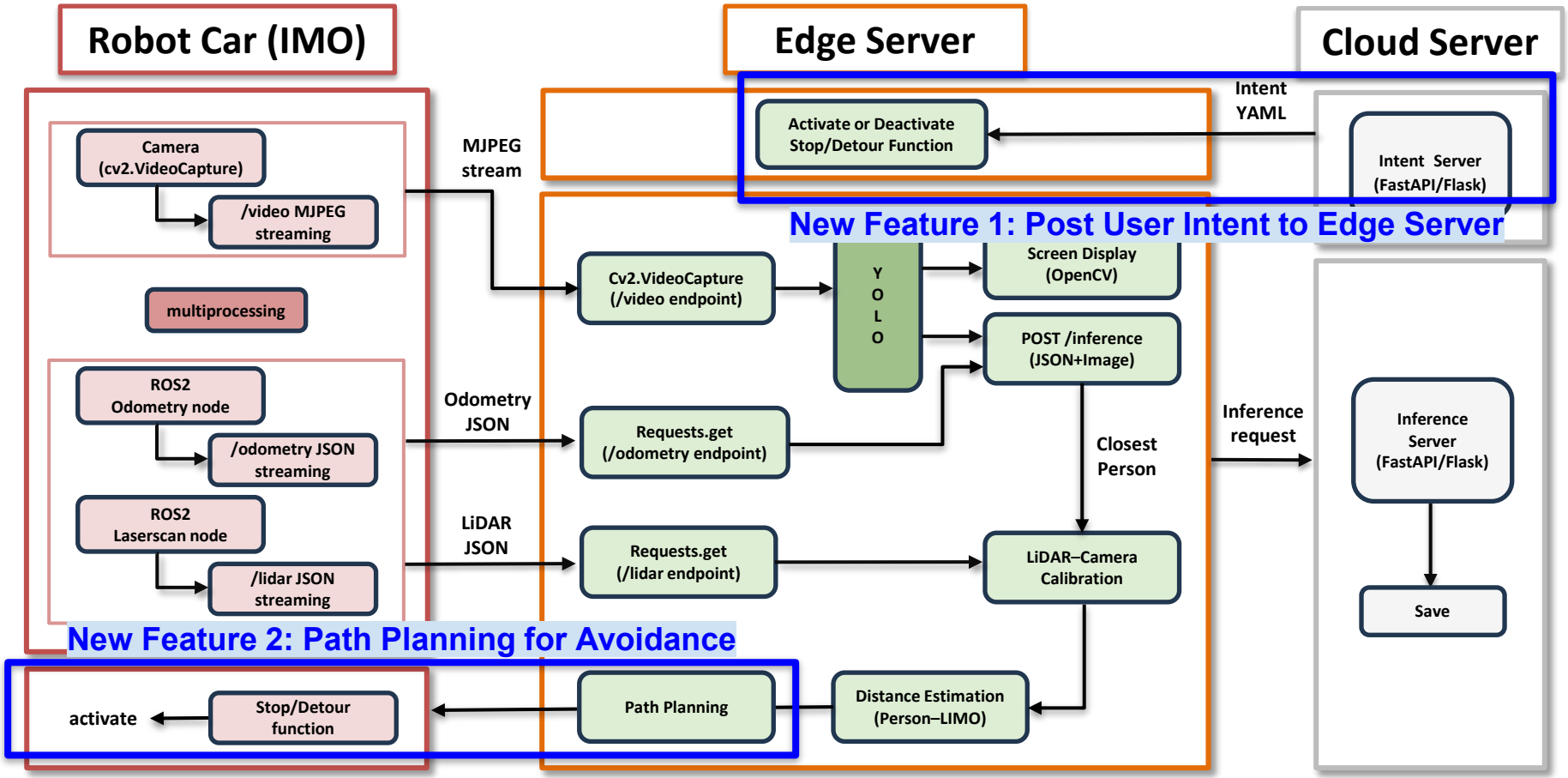
Goal of Hackathon Project

- The goal is to show the feasibility of Configured Moving Objects to Collision Avoidance by self-determining obstacles and pedestrians
  - **I2ICF (Interface to In-Computing Functions) Moving Object Controller**
    - The robot car fuses the sensing data of both camera and LiDAR in real time to measure a distance to an object.
    - The robot car automatically detours for collision avoidance when a detected person is too close to it.
- Internet Drafts for the I2ICF Project
  - https://datatracker.ietf.org/doc/draft-jeong-opsawg-i2icf-problem-statement/
  - https://datatracker.ietf.org/doc/draft-jeong-opsawg-i2icf-framework/
  - https://datatracker.ietf.org/doc/draft-gu-nmrg-intent-translator/

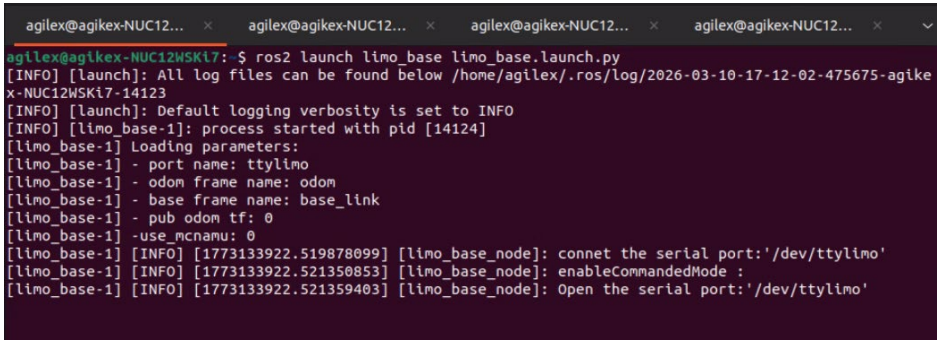
IETF-124 I2ICF Framework (Before)



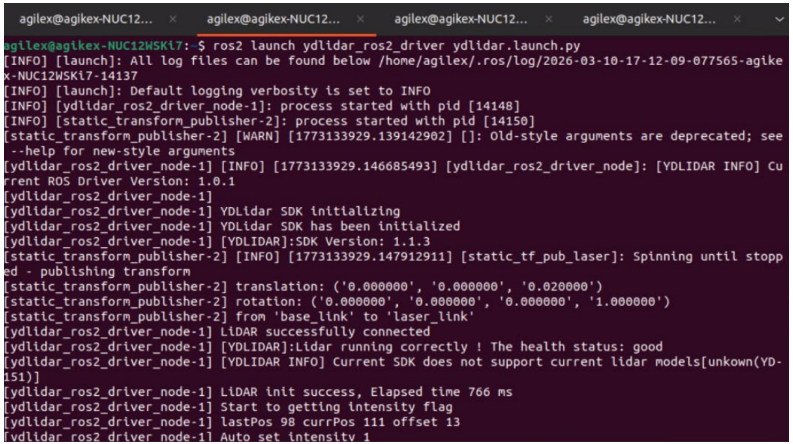
IETF-125 I2ICF Framework (Now)



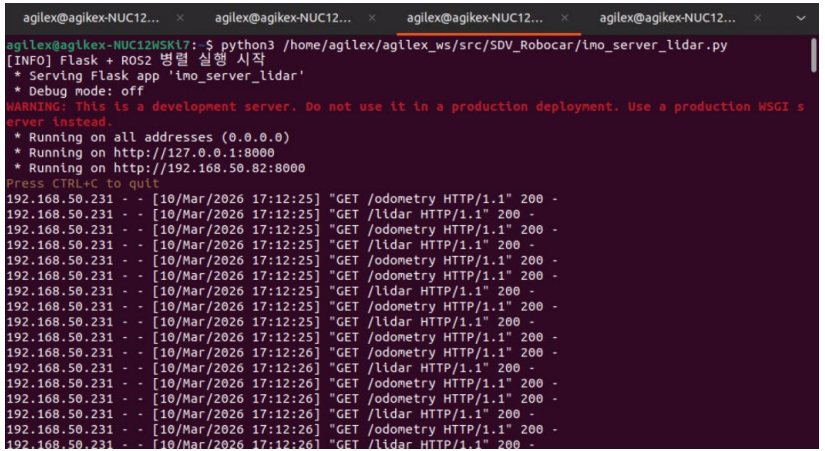
Demonstration (1/4)



1. [LIMO] Run ‘limo_base.launch.py’



2. [LIMO] Run ‘ydlidar.launch.py’



3. [LIMO] Run ‘limo_server_lidar.py’

Demonstration (2/4)

```
(gpu) eunjin@DESKTOP-CJ4J7QB:~/ros2_ws/IETF125-260310/SDV_Robocar$ python k8s_server.py
 * Serving Flask app 'k8s_server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI
server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8080
 * Running on http://115.145.178.171:8080
Press CTRL+C to quit
[2026-03-10_18-12-27] 이미지 저장 완료 → logs/images/2026-03-10_18-12-27.jpg
[2026-03-10_18-12-27] 수신 완료 | 객체 수: 7
127.0.0.1 - - [10/Mar/2026 18:12:27] "POST /inference HTTP/1.1" 200 -
[2026-03-10_18-12-27] 이미지 저장 완료 → logs/images/2026-03-10_18-12-27.jpg
[2026-03-10_18-12-27] 수신 완료 | 객체 수: 6
127.0.0.1 - - [10/Mar/2026 18:12:27] "POST /inference HTTP/1.1" 200 -
[2026-03-10_18-12-28] 이미지 저장 완료 → logs/images/2026-03-10_18-12-28.jpg
[2026-03-10_18-12-28] 수신 완료 | 객체 수: 4
```

4. Run ‘k8s_server.py’

```
(gpu) eunjin@DESKTOP-CJ4J7QB:~/ros2_ws/IETF125-260310/SDV_Robocar$ python intent_server.py
 * Serving Flask app 'intent_server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI
server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://115.145.178.171:5000
Press CTRL+C to quit
```

5. Run ‘intent_server.py’

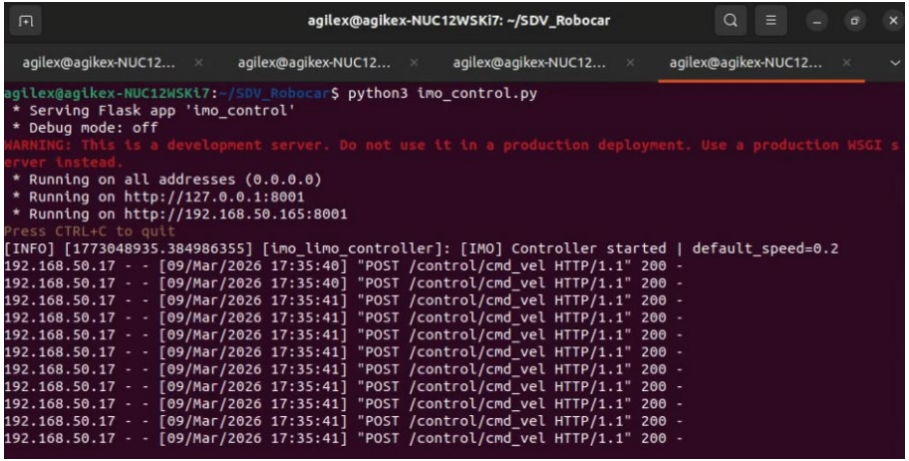
Demonstration (3/4)

```
(gpu) eunjin@DESKTOP-CJ4J7QB:~/ros2_ws/IETF125-260310/SDV_Robocar$ python3 edge_control.py
[Odom] 시작
[LiDAR] 시작
[Infer] 시작
[Sender] 시작
[Capture] 시작
[Sender] 전송 성공: 2026-03-10_18-12-27
[Sender] 전송 성공: 2026-03-10_18-12-27
[Robocar] distance -> 2.07 m 전송 OK
[Robocar] distance -> 1.98 m 전송 OK
[Sender] 전송 성공: 2026-03-10_18-12-28
[Robocar] distance -> 1.45 m 전송 OK
[Robocar] distance -> 1.89 m 전송 OK
[Sender] 전송 성공: 2026-03-10_18-12-39
```

6-1. Run ‘edge_control.py’: pass

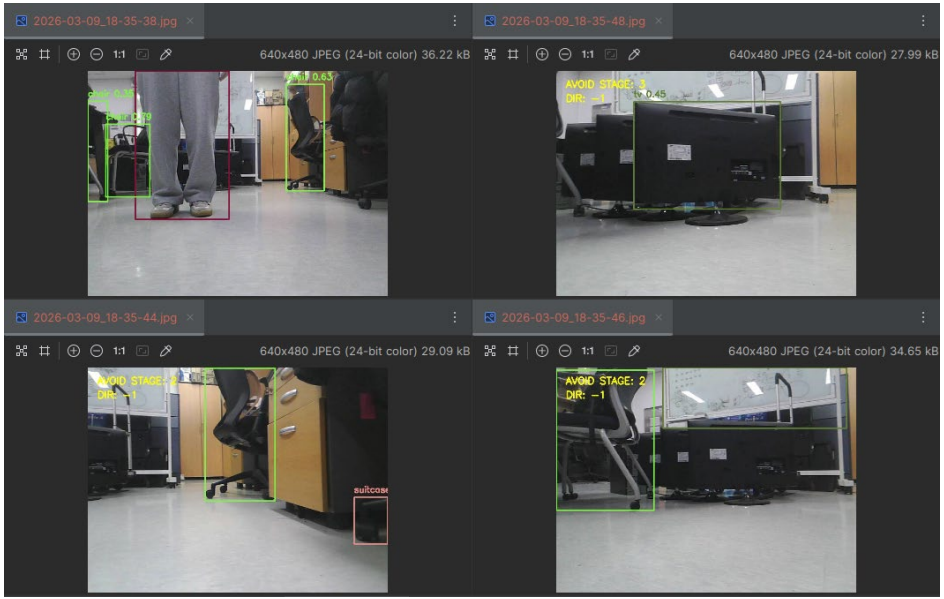
```
(gpu) eunjin@DESKTOP-CJ4J7QB:~/ros2_ws/IETF125-260310/SDV_Robocar$ python3 edge_control.py
[Odom] 시작
[LiDAR] 시작
[Infer] 시작
[Sender] 시작
[Capture] 시작
[Policy] ingress-action=drop -> distance 전송 차단 (1.74 m)
[Policy] ingress-action=drop -> distance 전송 차단 (2.12 m)
[Sender] 전송 성공: 2026-03-10_18-09-12
[Policy] ingress-action=drop -> distance 전송 차단 (1.84 m)
[Policy] ingress-action=drop -> distance 전송 차단 (2.12 m)
[Policy] ingress-action=drop -> distance 전송 차단 (2.11 m)
[Policy] ingress-action=drop -> distance 전송 차단 (1.84 m)
[Policy] ingress-action=drop -> distance 전송 차단 (1.84 m)
[Policy] ingress-action=drop -> distance 전송 차단 (1.84 m)
[Policy] ingress-action=drop -> distance 전송 차단 (2.11 m)
```

6-2. Run ‘edge_control.py’: drop

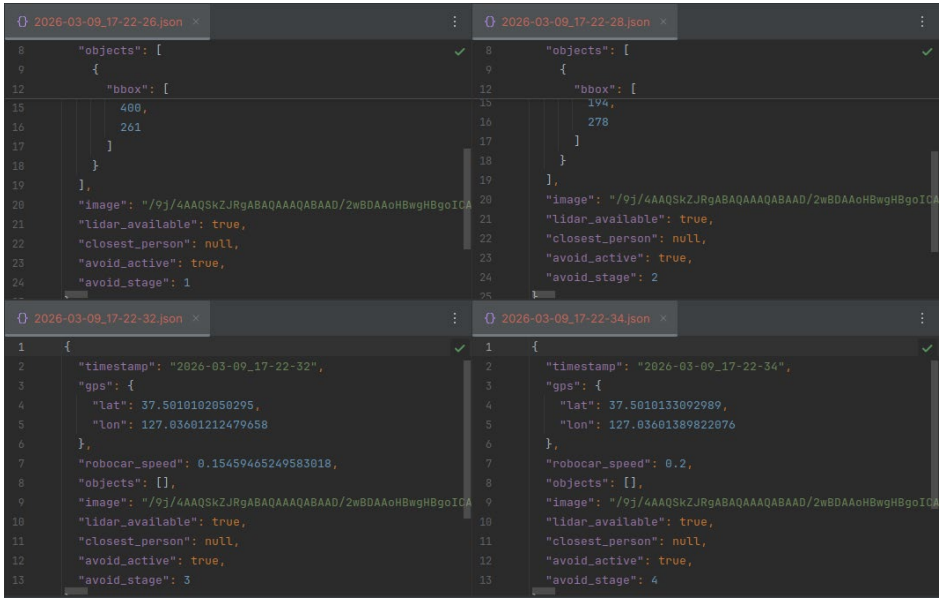


7. Run ‘limo_control.py’

Demonstration (4/4)



Result 1: YOLO Detection & Distance Measurement



Result 2: Logs

What we learned

- We implemented a Real-time Perception-to-Avoidance Pipeline in the I2ICF Framework.
  - Sending a User Intent to Activate or Deactivate Stop/Detour Function in a Robot Car.
  - Measuring Distance with the Robot Car's Camera and LiDAR.
  - Auto-Avoiding an Obstacle for Safety.

- Intent-Based Driving
  - The robot car automatically plans a moving path when an obstacle is nearby.

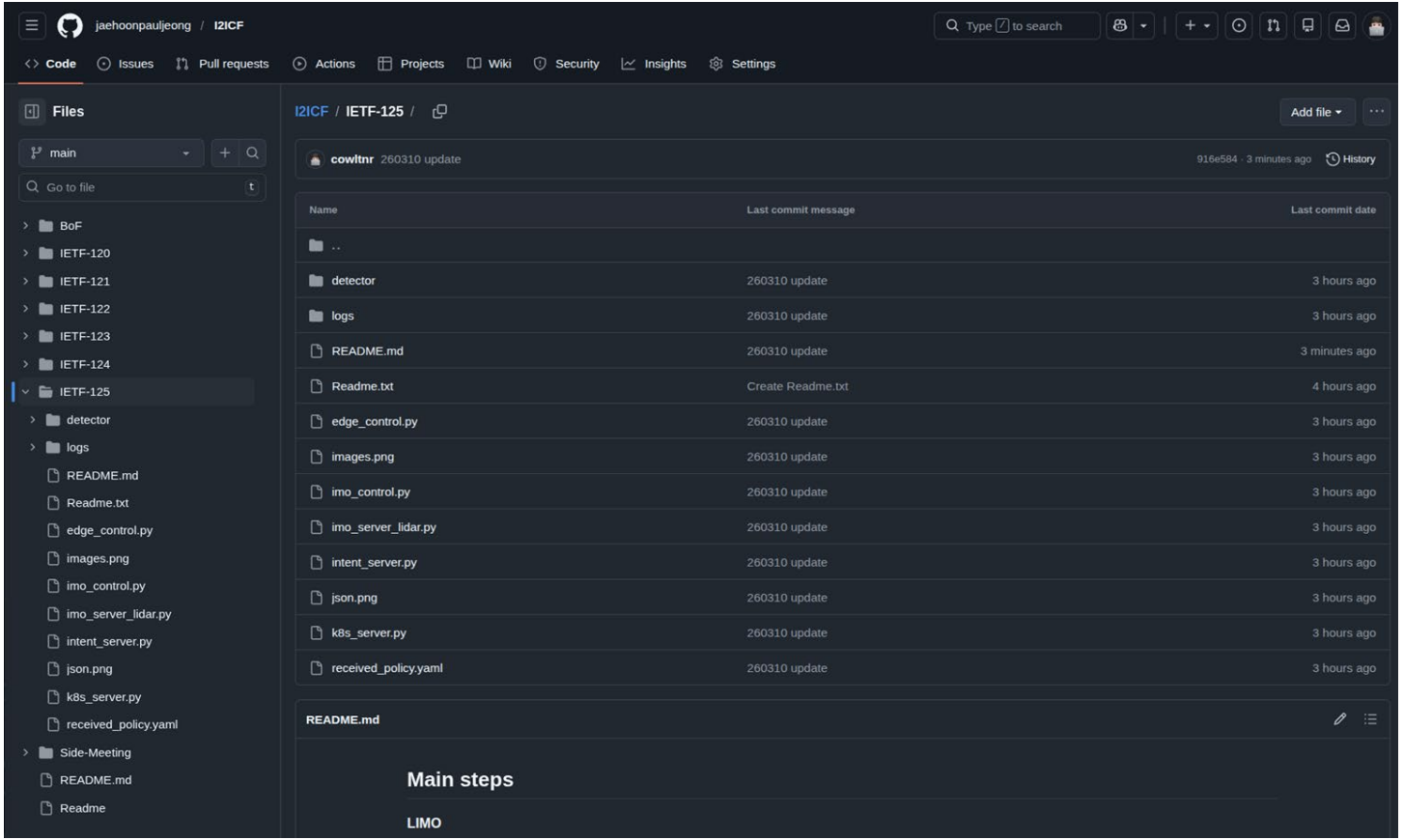
Next Step

- **Design & Implementation of Intent Translator and Policy Translator for Intelligent Moving Objects (e.g., Robot Car and Humanoid).**

- **IETF-126 Hackathon Project**
  - Extension of I2ICF Testbed
    - Collision Avoidance between Robot Cards using Vehicle-to-Vehicle (V2V) Communication
  - YANG Data Models for I2ICF Moving Object Management
    - https://datatracker.ietf.org/doc/draft-jeong-opsawg-i2icf-framework/

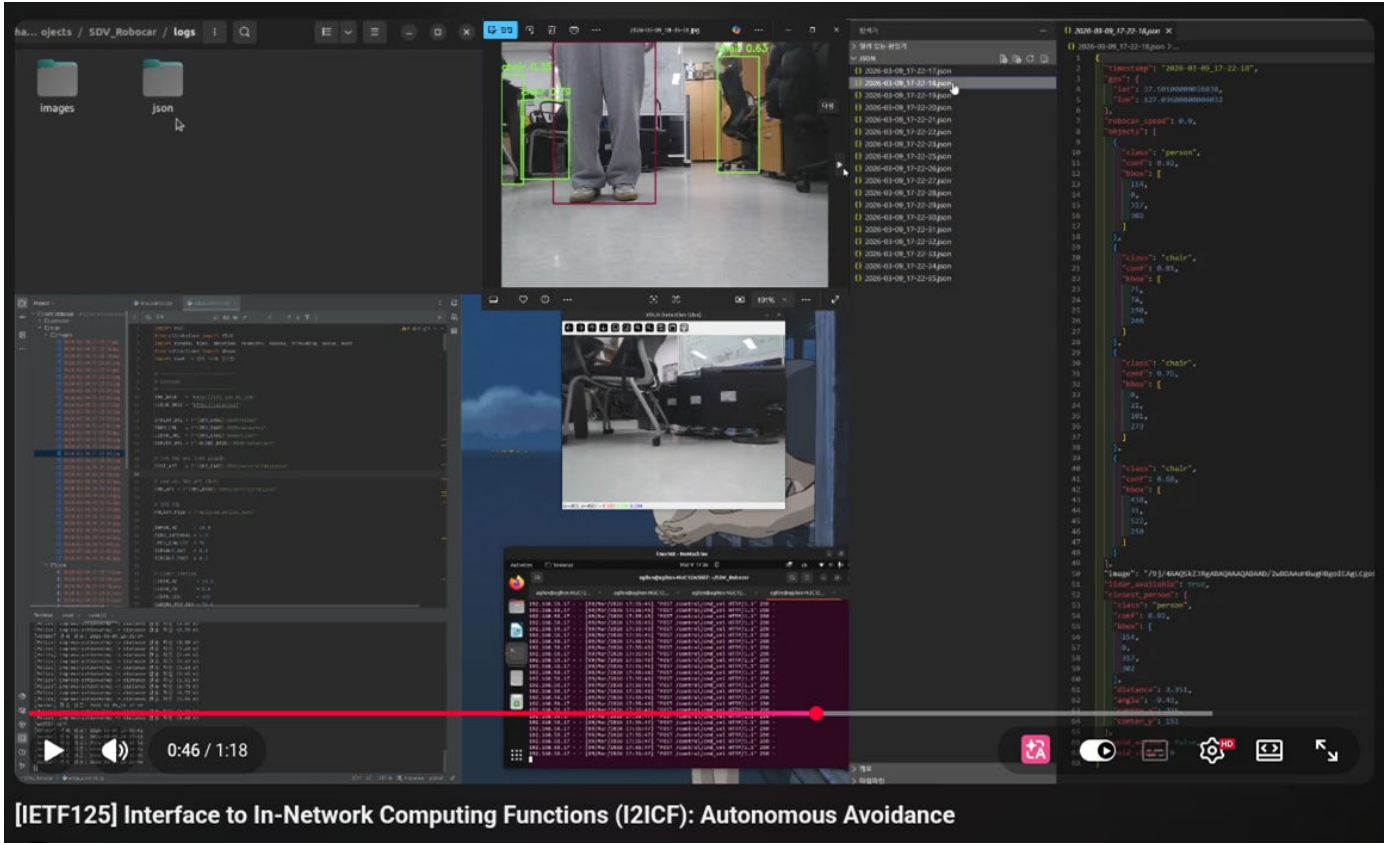
Open-Source Project for I2ICF

[URL] https://github.com/jaehoonpauljeong/I2ICF/tree/main/IETF-125



Demonstration Video Clip for I2ICF

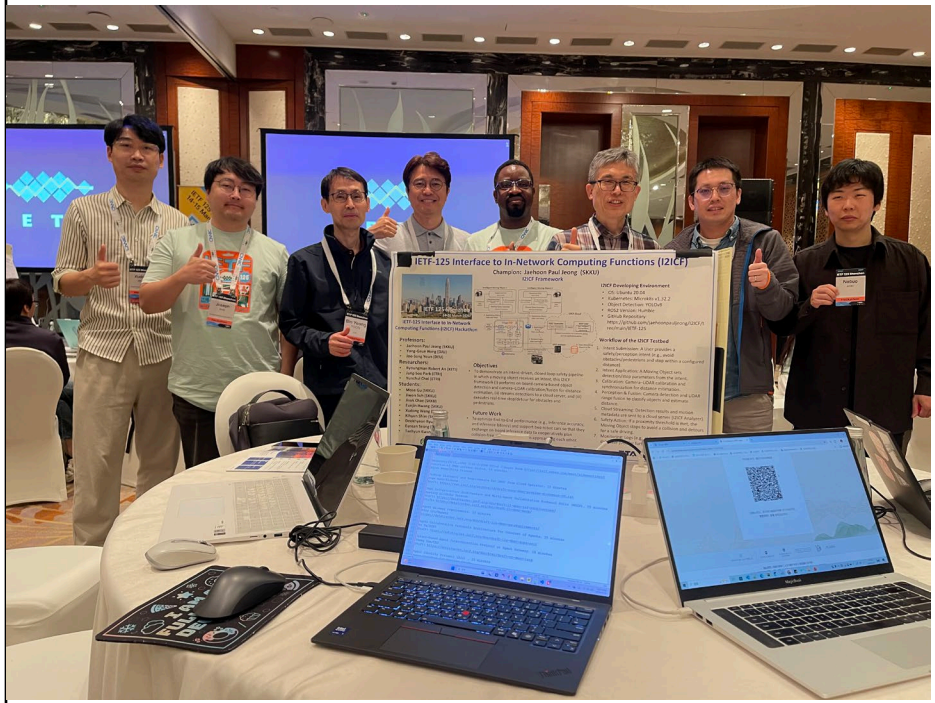
[URL] https://youtu.be/589GNGIX3fk?si=oZB06Nk9Y8ozSOLz



I2ICF Hackathon Team

- **Professors**:
  - Jaehoon Paul Jeong (SKKU)
  - Yong-Geun Hong (DJU)
  - Joo-Sang Youn (DEU)
  - Pyung Soo Kim (TU Korea)
- **Researcher**:
  - Kehan Yao (China Mobile)
  - Byoungman Robert An (KETI)
  - Jung-Soo Park (ETRI)
  - Yunchul Choi (ETRI)
  - Bin Yeong Yoon (ETRI)
- **Students**:
  - Mose Gu (SKKU) , Jiwon Suh (SKKU), Jisuk Chae (SKKU)
  - Eunjin Hwang (SKKU), Xudong Wang (SKKU)
  - Bien Aime Mugabarigira (SKKU), Kihyun Shim (SKKU)
  - Deokhyeon Ryu (DJU), Eunsan Seong (DEU)
  - Taehyun Kwan (DEU), Nobuo Aoki (Independent)

Hackathon Team Photo

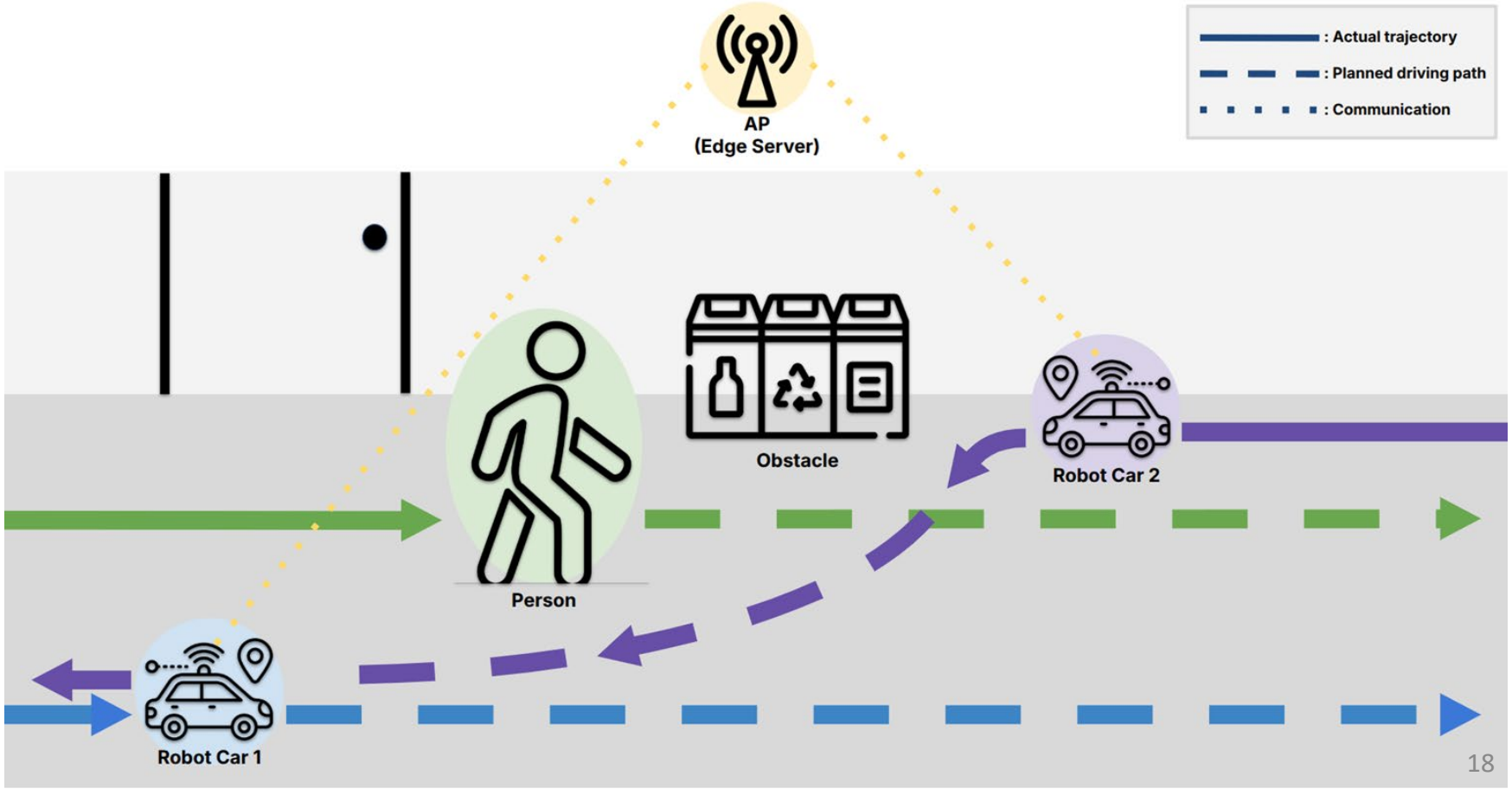


Appendix

Differences from the previous work

- The Previous Goal
  - To perform onboard camera-based object detection and camera-LiDAR calibration/fusion for distance estimation.
  - To let the moving object deliver the detection results to a cloud server to perform real-time stopping to avoid the collision with obstacles.
  - Camera-LiDAR data is calibrated and fused; collision control is performed via the powertrain ECU; detection results are transferred to a cloud server.
- The Present Goal
  - To have a moving object (i.e., Robot Car) receive an intent and autonomously drive.
  - The robot automatically plans a detour when an object gets too close.

Final Goal Scenario Illustration



Thank you!