IETF-124 Hackathon

Interface to In-Network Computing Functions (I2ICF) Project

November 2, 2025, Montreal

Champion: Jaehoon (Paul) Jeong

Members: Yoseop Ahn, Mose Gu, Jiwon Suh, **Jisuk Chae**

Department of Computer Science and Engineering at SKKU

Korea Electronics Technology Institute

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

IETF-124 Interface to In-Network Computing Functions (I2ICF)

Champion: Jaehoon Paul Jeong (SKKU)

IETF-124 Interface to In-Network Computing Functions (I2ICF) Hackathon

Professors:

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

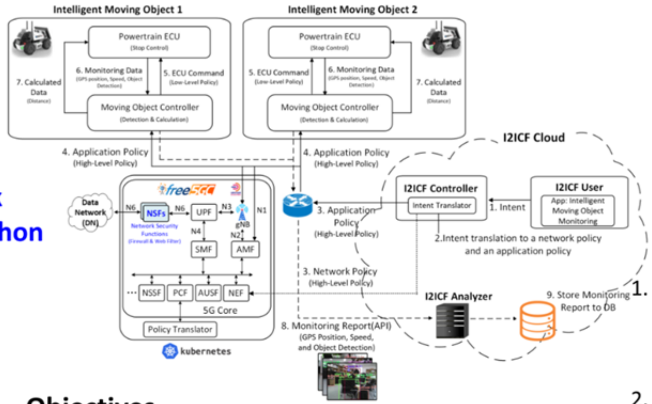
Researchers:

- Jung-Soo Park (ETRI)

Students:

- Mose Gu (SKKU)
- Jiwon Suh (SKKU)
- Jisuk Chae (SKKU)
- Yeonjoo Lee (SKKU)
- Kihyun Shim (SKKU)
- Isaac Choi (SKKU)
- Wonjae Lee (SKKU)
- Deokhyeon Ryu (DJU)
- Eunjin Hwang (DJU)
- Sumin Park (DJU)
- Eunsan Seong (DEU)
- Taehyun Kwan (DEU)

I2ICF Framework



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/avoidance for obstacles and pedestrians.

Future Work

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

I2ICF Developing Environment

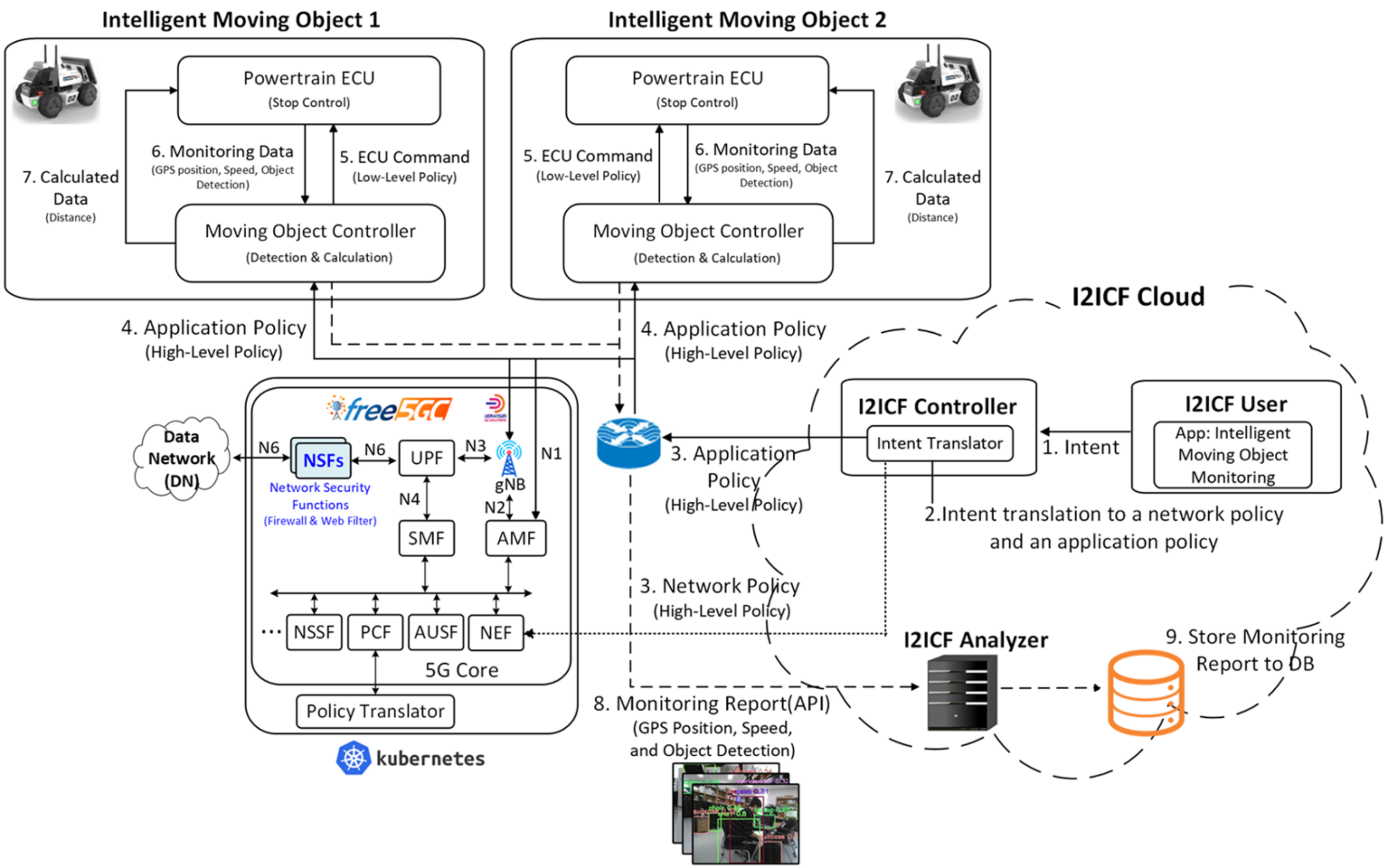
- OS: Ubuntu 20.04
- Kubernetes: Microk8s v1.32.2
- Object Detection: YOLO v8
- ROS2 Version: Humble
- GitHub Repository: https://github.com/jaehoonpauljeong/I2ICF/tree/main/IETF-124/SDV_Robocar

Workflow of the I2ICF Testbed

1. Intent Submission: A User provides a safety/perception intent (e.g., avoid obstacles/pedestrians; stop within a configured distance).
2. Intent Application: A Moving Object sets detection/stop parameters from the intent.
3. Calibration: Camera–LiDAR calibration; 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.

성균관대학교 SUNG KYUN KWAN UNIVERSITY · 숭실대학교 Soongsil University · 대전대학교 DAEJEON UNIVERSITY · 동의대학교 DONG-EUI UNIVERSITY · 아주대학교 AJOU UNIVERSITY · ETRI 한국전자통신연구원 · SAMSUNG · TTA 한국정보통신기술협회

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

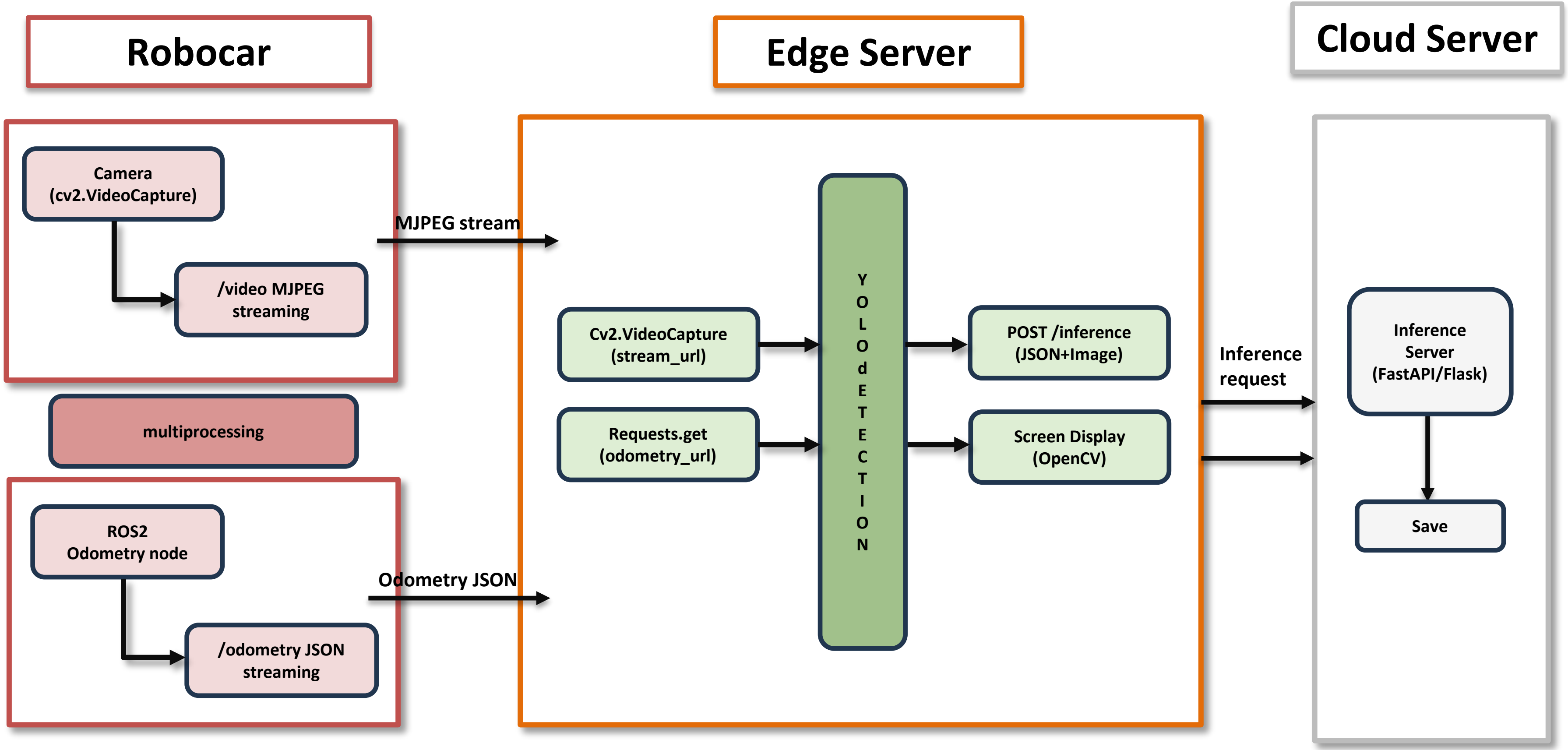


Goal of Hackathon Project

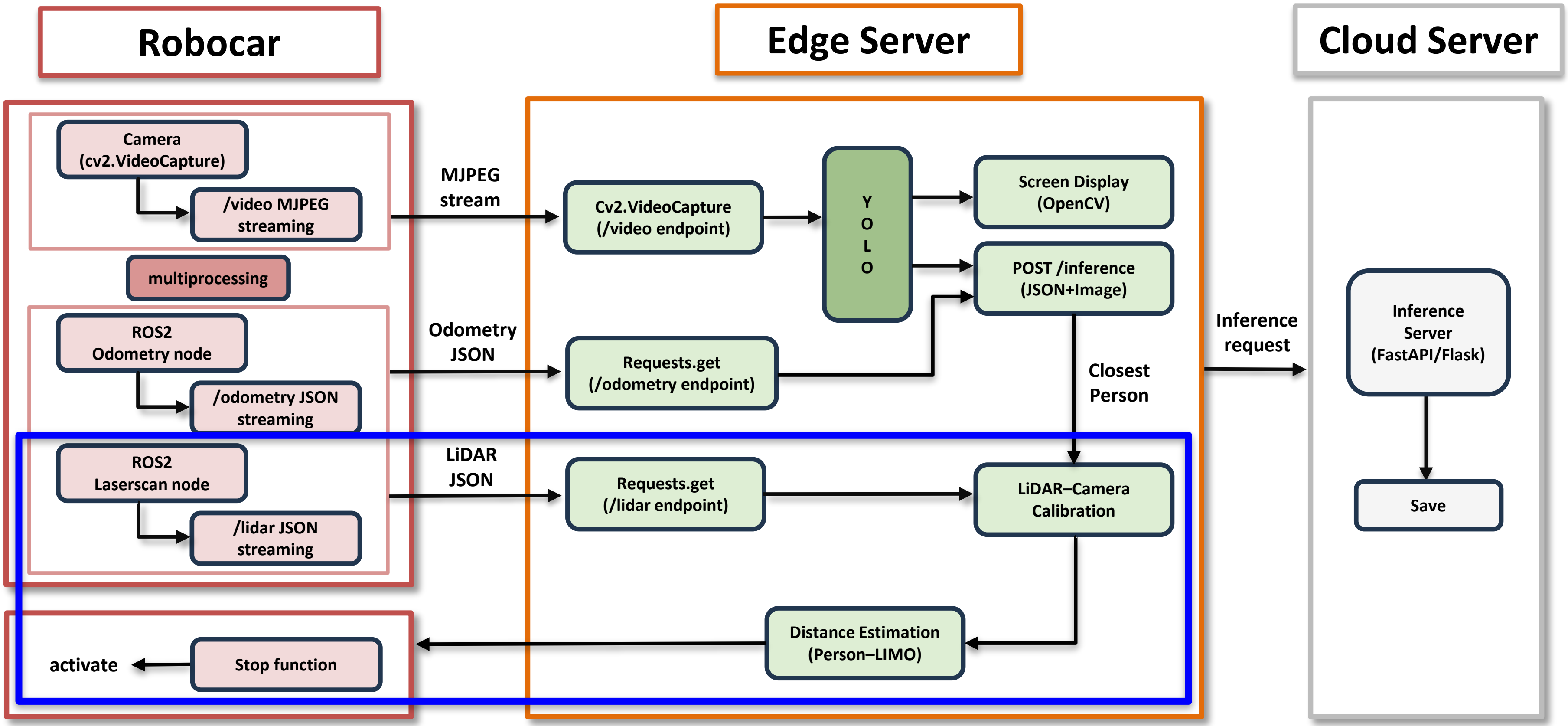
- The goal is to show the feasibility of a Configured moving objects to Collision avoidance by self-determining obstacles and pedestrians
  - **I2ICF (Interface to In Computing Functions) Moving Object Controller**
    - The robot car fuses the sensing data of both camera and LiDAR in real time to measure distance and stops automatically when it is too close.

- 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-123 I2ICF Framework (Before)

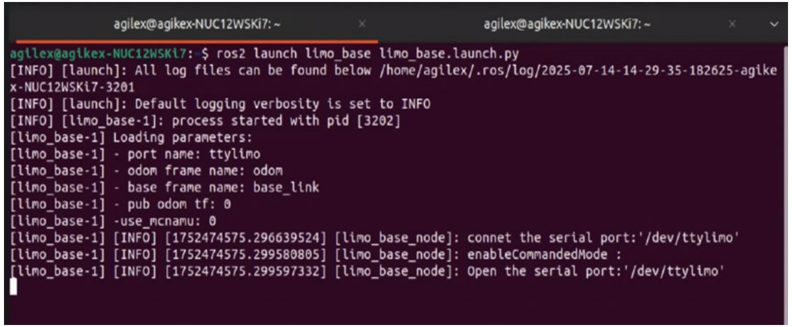


IETF-124 I2ICF Framework (Now)

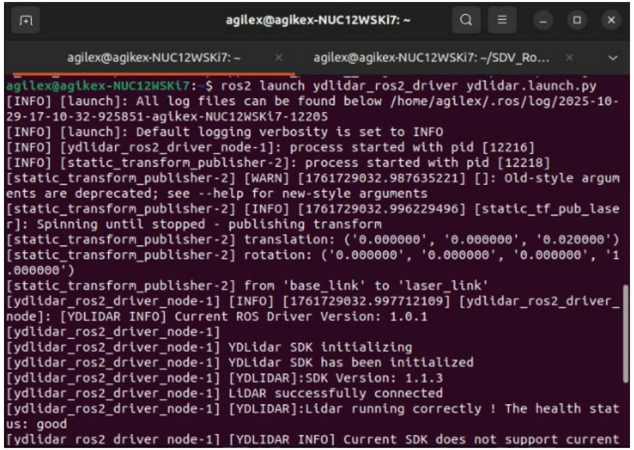


New Feature: Calibrating LiDAR and Camera for Distance Measurement

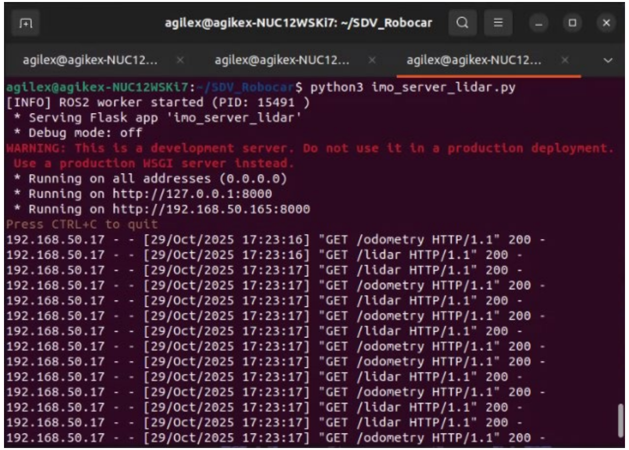
Demonstration (1/3)



1. [LIMO] Run ‘limo_base.launch.py’

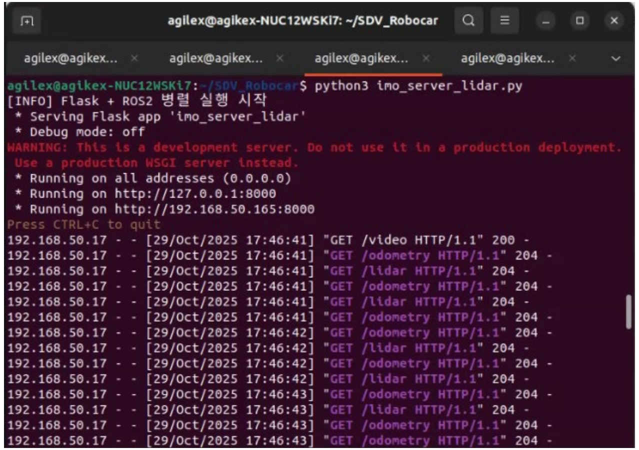


2. [LIMO] Run ‘ydlidar.launch.py’

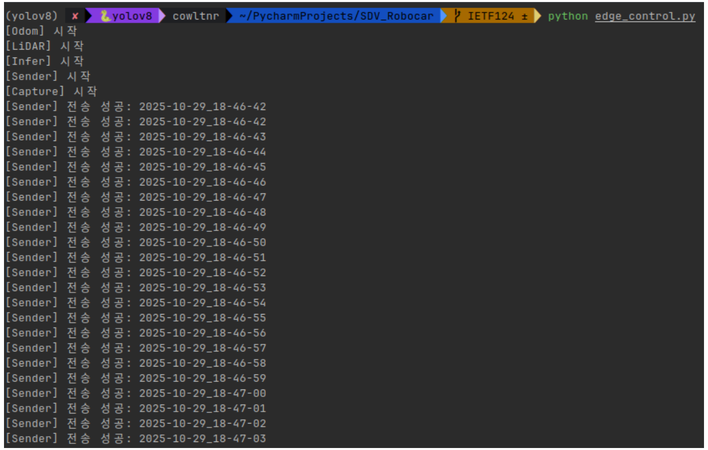


3. [LIMO] Run ‘imo_server_lidar.py’

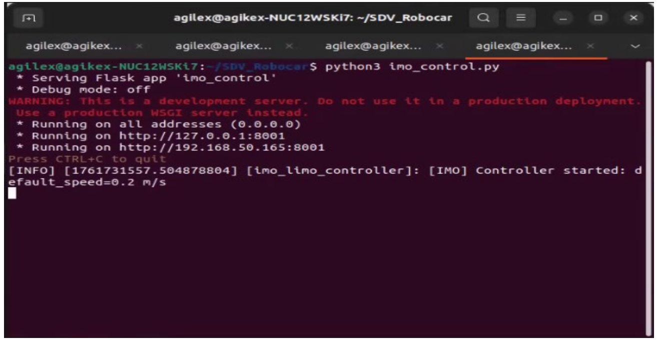
Demonstration (2/3)



4. Run ‘k8s_server.py’

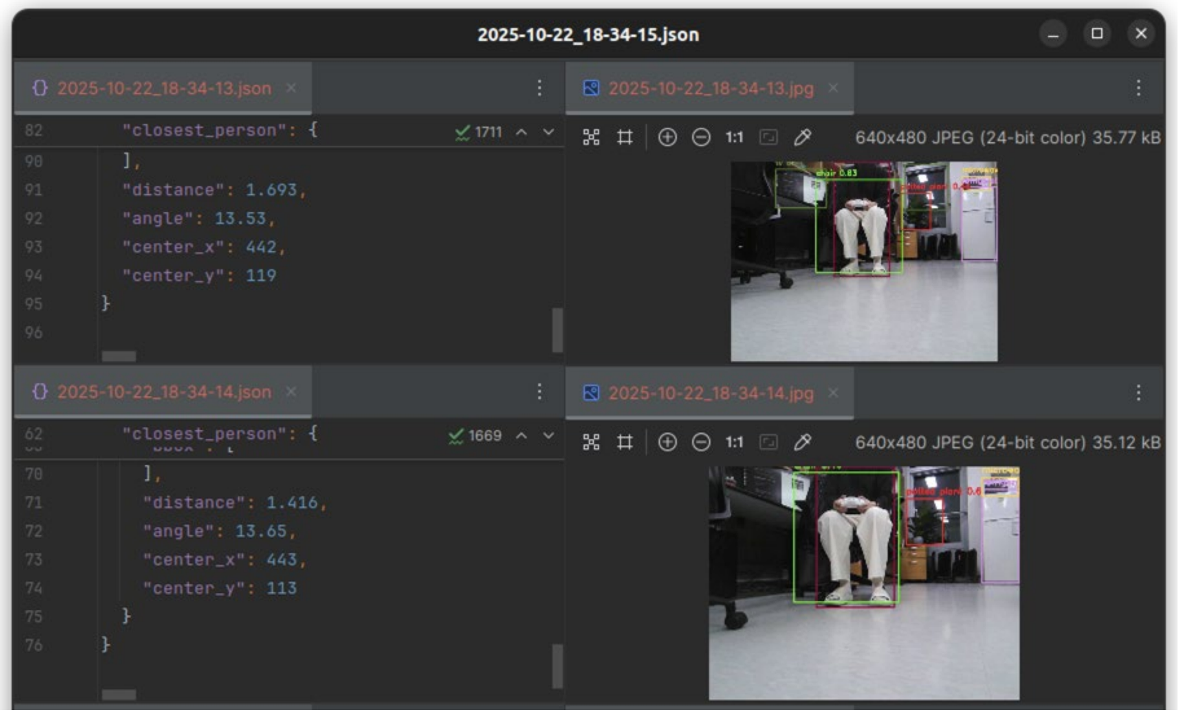


5. Run ‘edge_control.py’

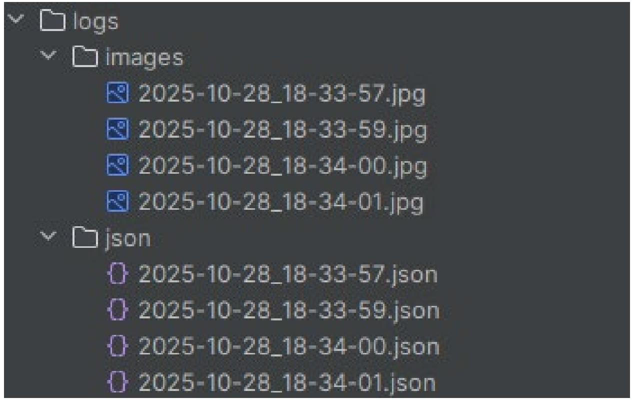


6. Run ‘imo_control.py’

Demonstration (3/3)



Result 1: YOLO Detection & Distance



Result 2: Logs

What we learned

- We implemented a real-time perception-to-stop pipeline.
  - Distance Measurement with Camera and LiDAR and Auto-Stopping for an Obstacle.
- Intent-Based Driving
  - Robot stops when an obstacle is near.

Next Steps

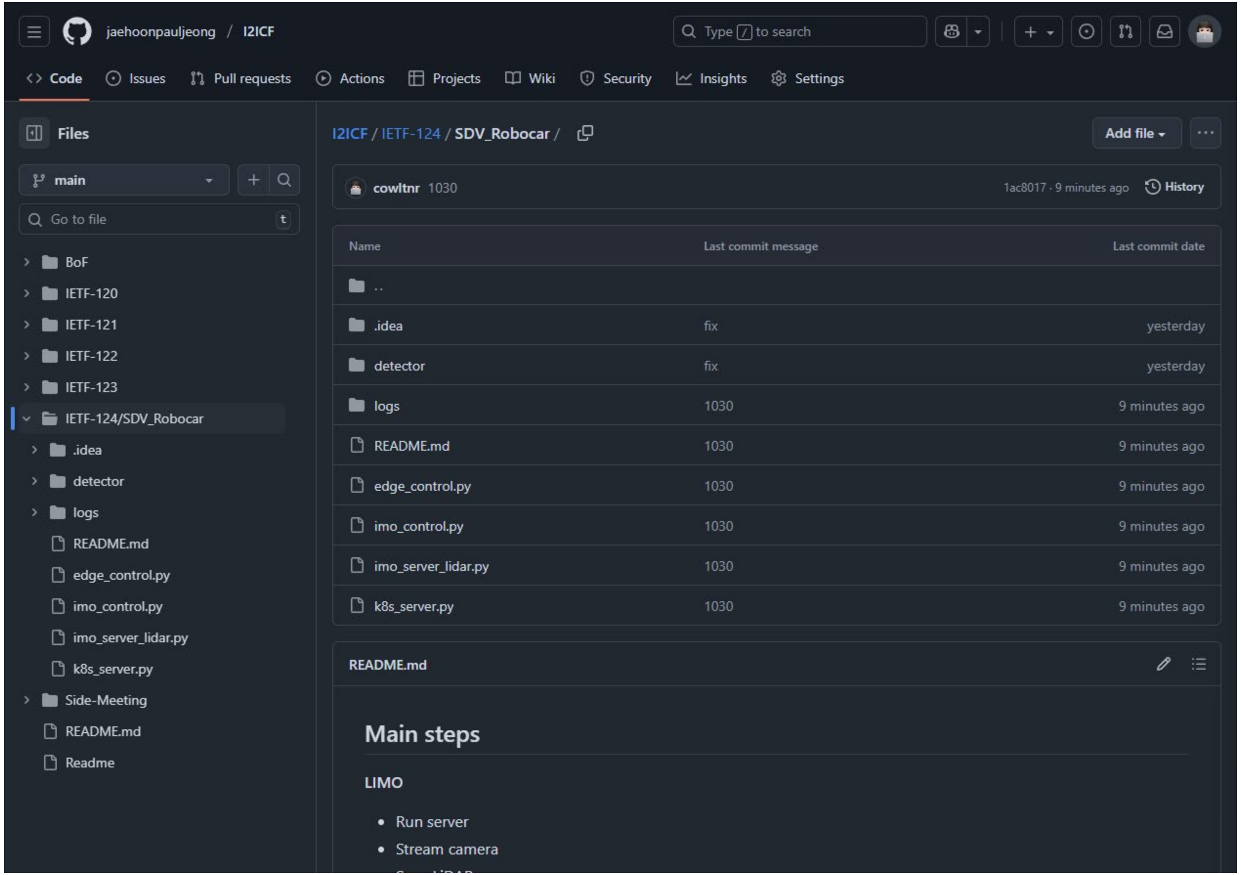
- **Design & Implementation of Intent Translator and Policy Translator**

- **Optimization of End-to-End Pipeline**
  - Reduction of frame drops and inference latency
  - Stabilization of distance estimation

- **IETF-125 Hackathon**
  - Extension of I2ICF Testbed
    - Exchange inference data and cooperative maneuver control.
  - YANG Data Models for I2ICF Moving Object Management
    - https://datatracker.ietf.org/doc/draft-jeong-opsawg-i2icf-framework/

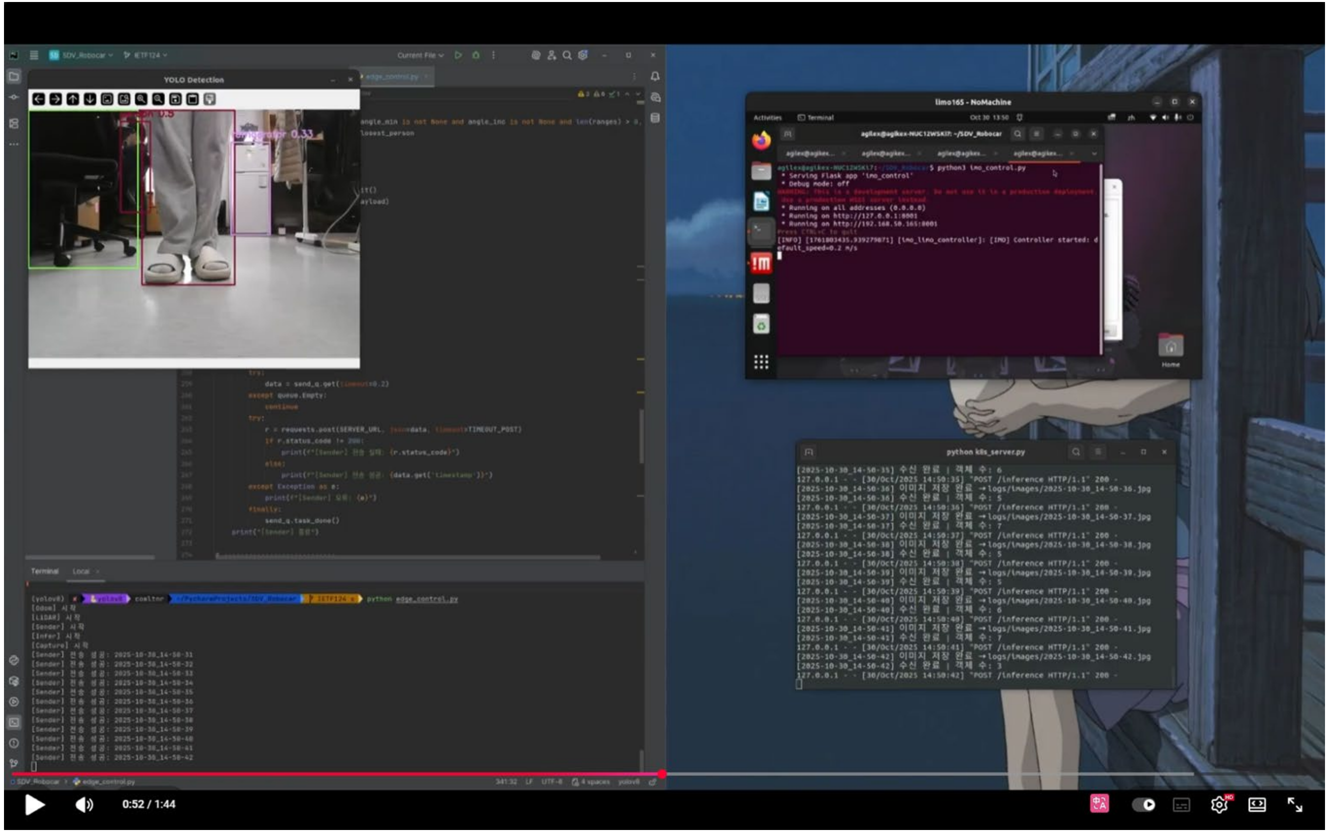
Open-Source Project for I2ICF

[URL] https://github.com/jaehoonpauljeong/I2ICF/tree/main/IETF-124/SDV_Robocar



Demonstration Video Clip for I2ICF

[URL] https://www.youtube.com/watch?v=lw0GV3yN5Nw



I2ICF Hackathon Team

- **Professors:**
  - **Jaehoon Paul Jeong (SKKU)**
  - **Yong-Geun Hong (DJU)**
  - **Joo-Sang Youn (DEU)**
- **Researcher:**
  - **Jung-Soo Park (ETRI)**
- **Students:**
  - **Mose Gu (SKKU)**
  - **Jiwon Suh (SKKU)**
  - **Jisuk Chae (SKKU)**
  - **Yeonjoo Lee (SKKU)**
  - **Kihyun Shim (SKKU)**
  - **Isaac Choi (SKKU)**
  - **Wonjae Lee (SKKU)**

Hackathon Team Photo

Appendix

Differences from the previous work

- The Previous Goal
  - ✓ To implement User Equipment (UE)'s data collecting ability and a monitoring interface to collect data from the UE which is directed by a user's intent.
  - ✓ We collect data from a UE (i.e., Robot Car) and deliver it to an Analyzer Component for the framework's closed loop control.
- The Present Goal
  - ✓ To have a moving object (i.e., Robot Car) receive an intent and autonomously drive.
  - ✓ 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.