(a) Compilation / Run-time Interrupt

Compilation

- CNN model → (Instr. Gen) → Normal Intructions
- Normal Intructions → Interruptible?
- Interruptible? → Yes → Interruptible Positions
- Interruptible? → No → High-priority Task
- Interruptible Positions → (Enable Interrupt) → Virtual Instr. ISA sequence
- High-priority Task → (Extend ISA) → Virtual Instr. ISA sequence

Run-time Interrupt

- IAU: Hardware for Interrupt
- Software Driver in ROS nodes
- Normal Intruction sequence

(b) Robotic Software in ROS

Robotic Software in ROS

- Feature-Point Extraction (FE)
- Place Recognistion (PR)
- VO with Feature-Point
- Exploration/ Navigation/ ...

Compilation for CNN backbone

- CNN 0 → not-Interruptible Instructions
- CNN1 → Interruptible Instructions

Post-Processing

- Norm
- Rank
- SoftMax
- ...

CPU Tasks

- C++
- Python

Programmable Logic (PL, FPGA)

- Interruptible CNN Accelerator
- Special Logic

ScratchPad memory

Processing System (PS, CPU)

- Software Optimizing
- ROS Nodes

(a) The offline compilation with virtual-instruction ISA and the online interrupt between tasks. The normal instructions are generated from the CNN model, and then translated to virtual instruction ISA sequence. IAU translate the virtual instruction ISA sequence to normal sequence executed on the accelerator.

(b) The INCAME framework. The tasks (FE/PR/VO) are in different ROS nodes. The CNN backbone are translated to instrucitons, which enable runtime interrupt on the accelerator. The post-processing is accelerated by the special logic. Different modules on FPGA and CPU side share data through scrachpad memory.