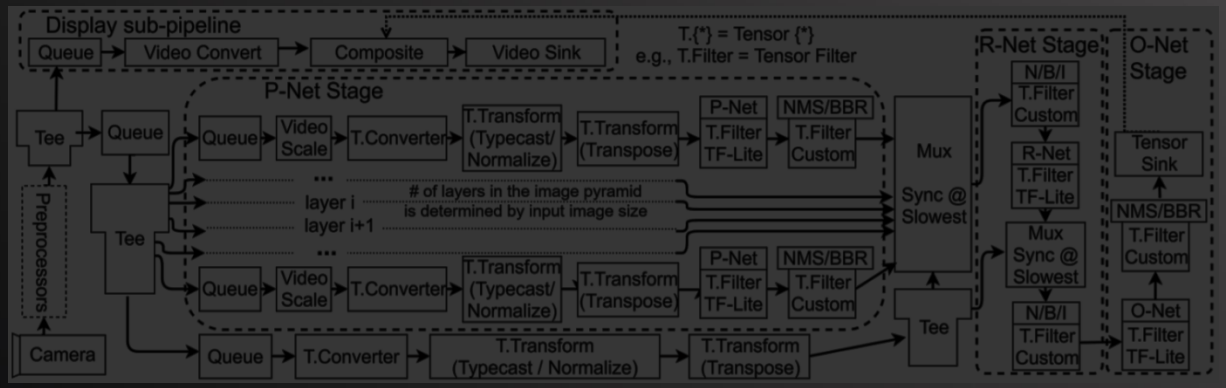
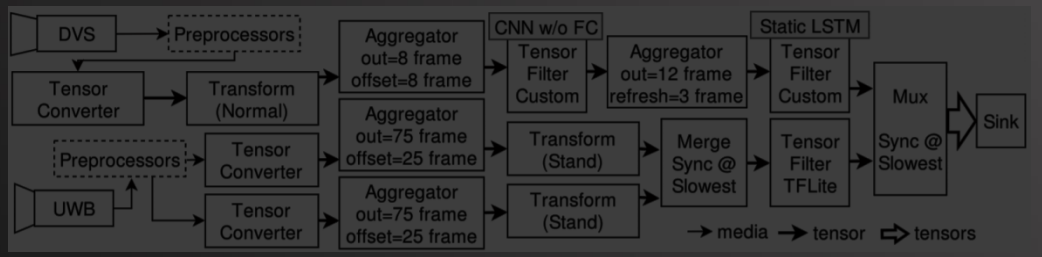
SAMSUNG Research



Neural Network Streamer

NNStreamer

October, 2019



SAMSUNG Research

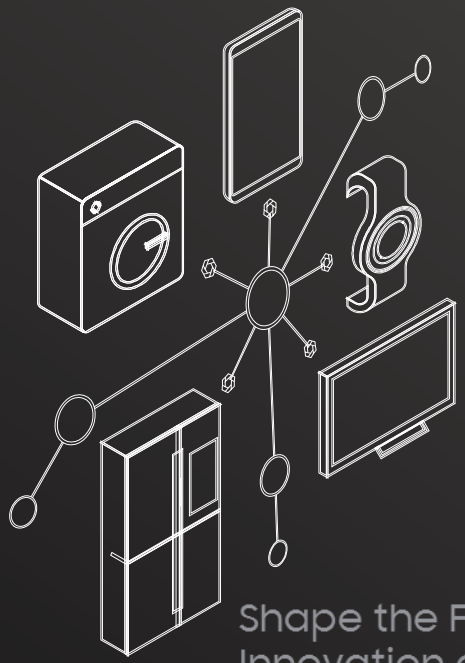
MyungJoo Ham, Ph.D.

About Speaker, an Open Source Contributor

- **NNStreamer: founder (2018)**
- **Linux Kernel**
  - Maintainer / founder (2012), "devfreq": DVFS for non-CPU devices
  - Maintainer / founder (2011), "extcon": External connectors (not active)
  - Committer: ARM, regulator, clock, PM, … (not active)
- **.NET Core**
  - Main committer: Linux-ARM C# Runtime (not active)
- **Tizen**
  - Architect: "System Framework", "Base Libraries" (2012~2014)
  - Architect: "Machine Learning"
- **Others (commits)**
  - Tensorflow, RPM, glibc, libunwind, obs-build, obs-build-service, libsolv, perl-BSSolv, pipeviz, …

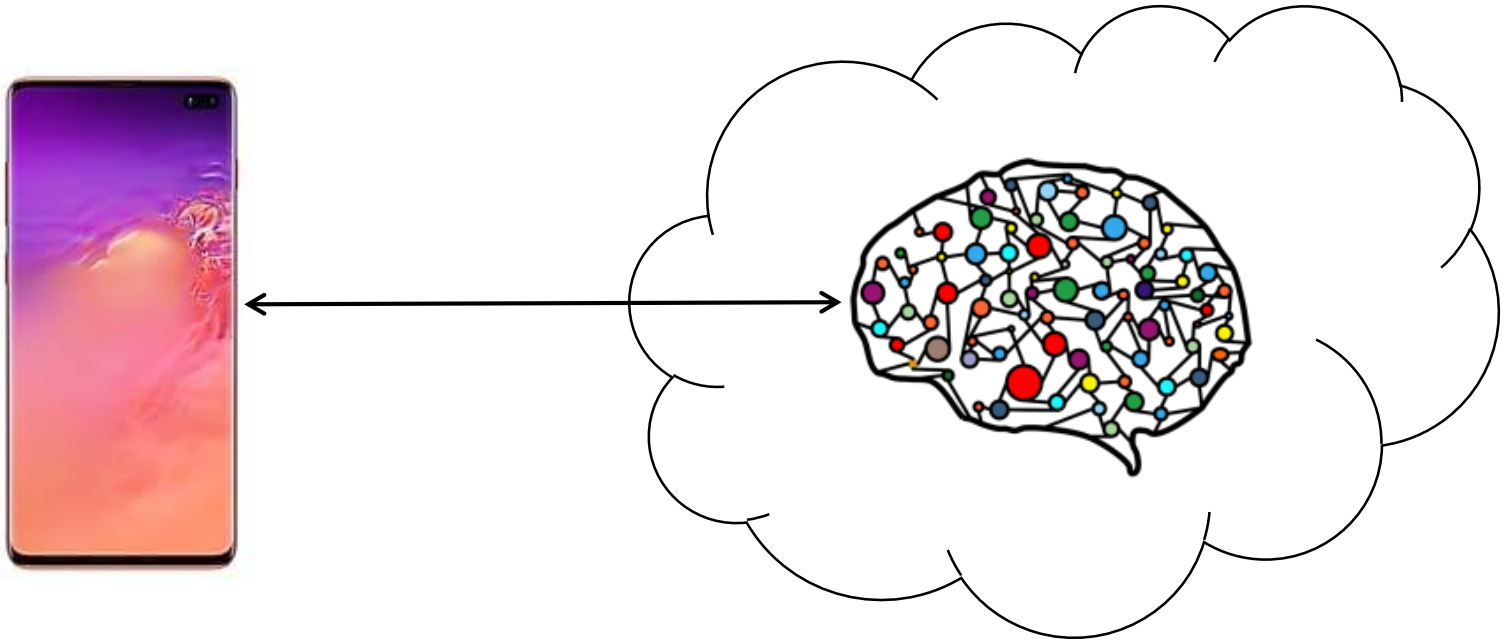
Talk is cheap. Show me the code.
- L. Torvalds

Contents



Shape the Future with
Innovation and Intelligence

Conventional. (Cloud AI)

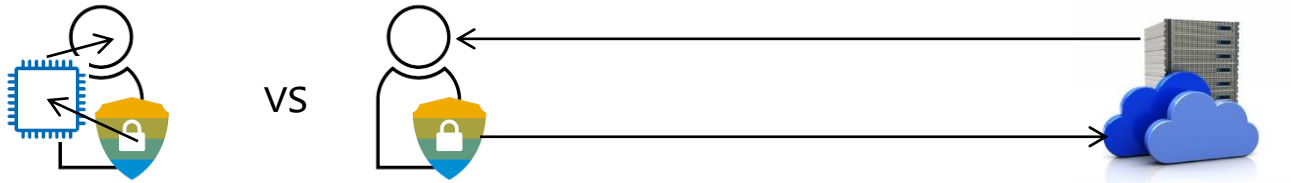


On-Device AI

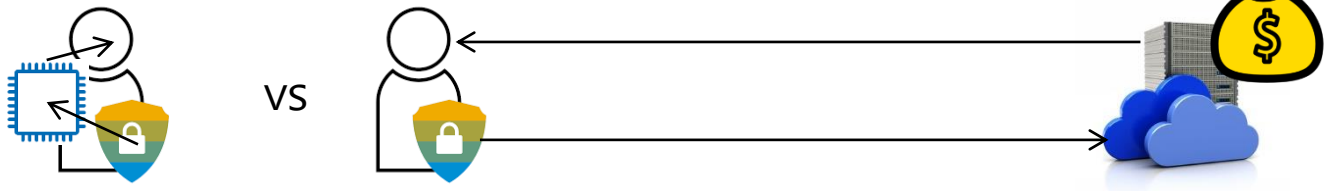
On-Device AI. Why?

Privacy & Data Protection

Low Latency



Operating Cost



On-Device AI. Performance.

Special-Purpose Hardware

- CNN H/W Accelerator
- ASR H/W Accelerator
- …
- …

Model Compression

- Quantization
- Pruning
- …
- …

On-Device AI. Issues

- **Higher Performance….**

- **Complexity**
  - Data paths
  - Multiple sensors
  - Multiple neural network models
  - Multiple applications

On-Device AI. Examples

Autonomous Vehicle

Multi-Modal Interaction

Case 1: Autonomous Vehicle

LIDAR

Camera

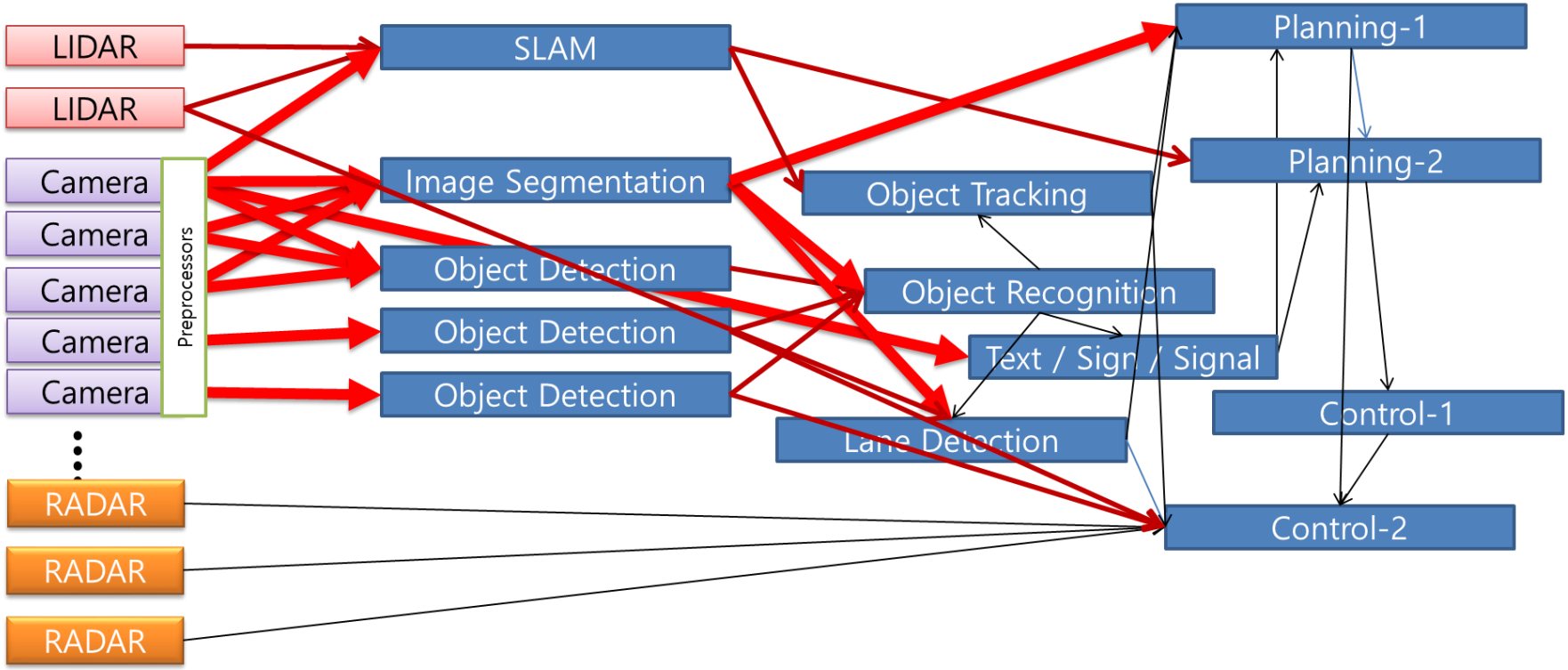
RADAR

- Object Detection
- Object Recognition
- Object Tracking
- Image Segmentation
- Lane Detection
- SLAM
- Planning
- Control

* 실제 시스템보다 단순화 되었음.

Case 1: Autonomous Vehicle

실제보다 단순화 된 가상 모델

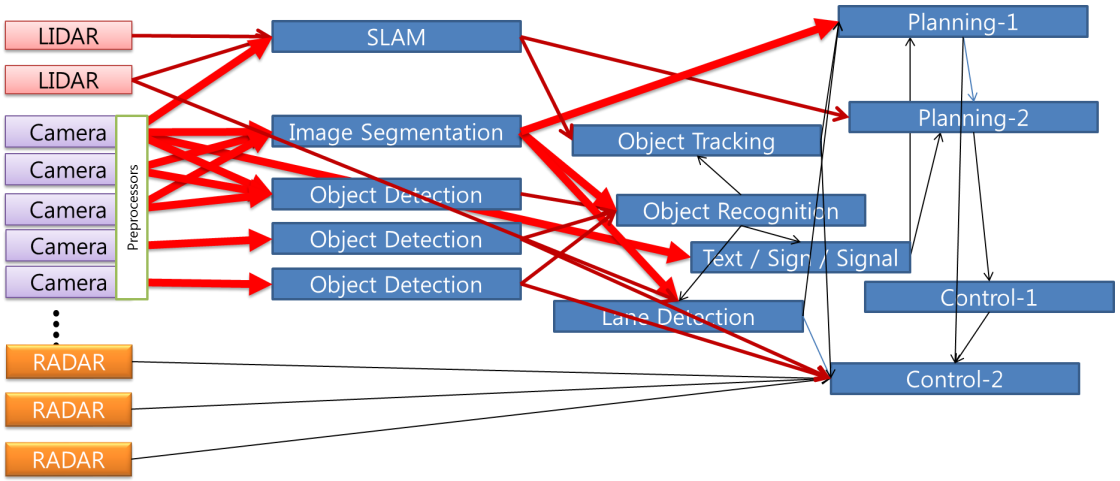


* 실제 시스템보다 단순화 되었음.

Case 1: Autonomous Vehicle

- High Bandwidth
  - 고해상도 이미지 실시간 전송
  - 모델간 데이터 전송
  - 중복 데이터 전송

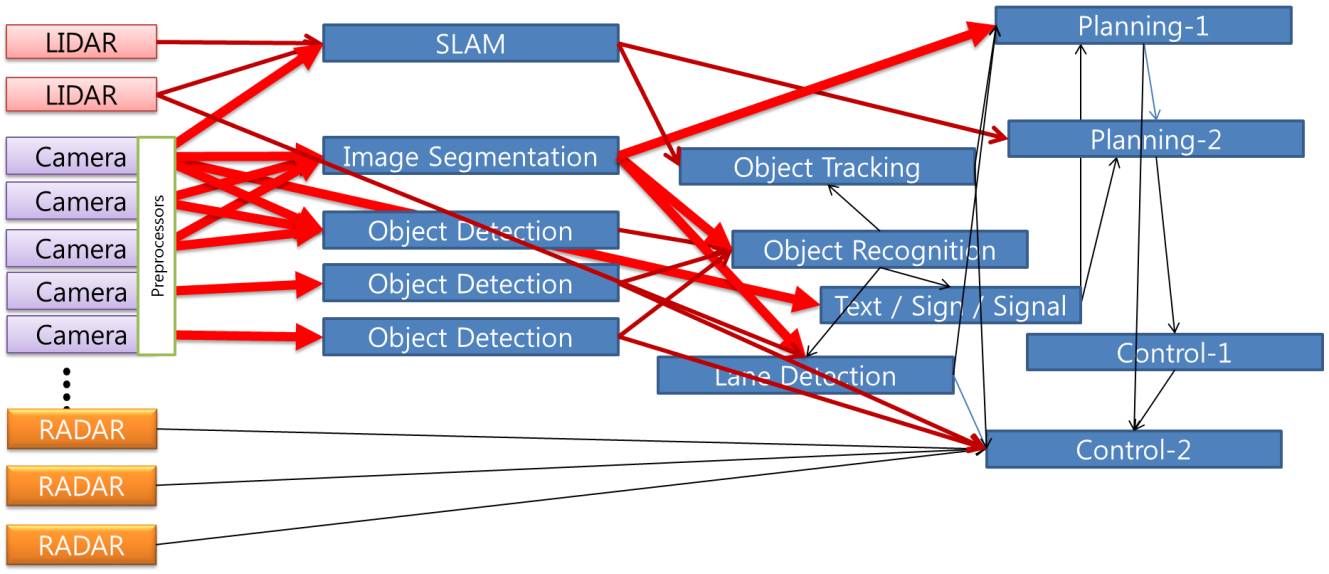
- 고성능 Workstation에서도
  감당하기 힘들 수 있음!!
  (Memory Transaction!)



Case 1: Autonomous Vehicle

Complex Pipeline Topology

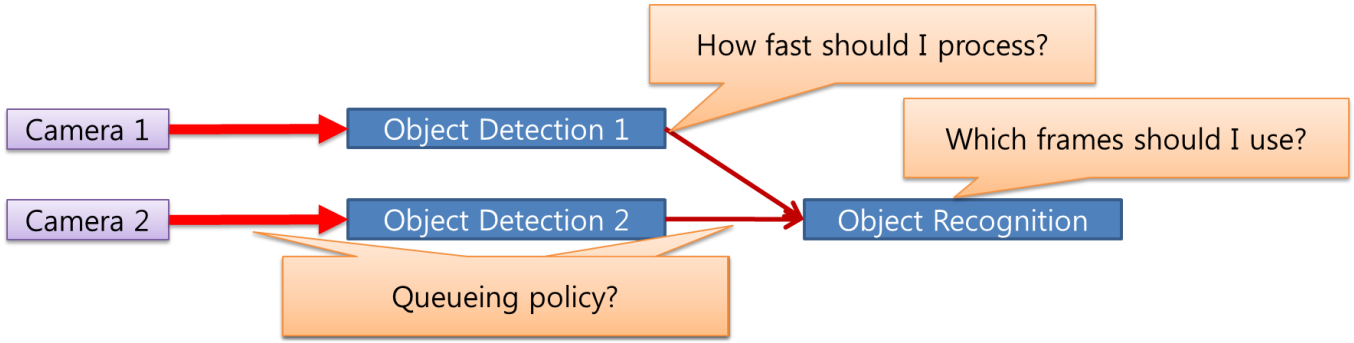
- 특히 여러 모델이 공용으로 쓰고 싶은 “기본 모델“?



Case 1: Autonomous Vehicle

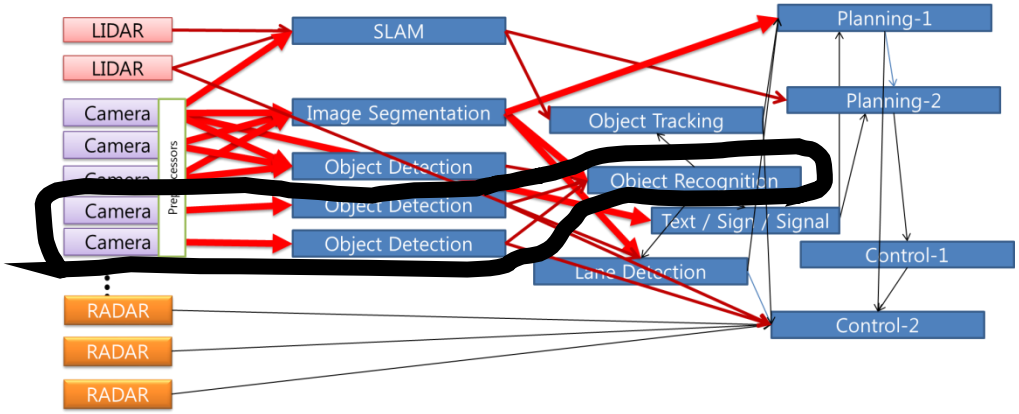
Synchronization

- 독립 Process로 동작하여 **골치 아플 수 있음!**



Parallelism

- 각 Node를 독립되게 동작.



Case 1: Autonomous Vehicle

- **Latency**
  - 과도한 Memcpy로 인해 Communication Latency 증가

- **Jitter Control**
  - 각 Node들에 대한 Resource/Rate 제어 어려움
  - Node간 생성물에 대하 동기화 어려움

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Multiple Sensors

카메라, 마이크, IR, 관성, 그 외 다수

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Multiple AI-based Services

감정인식, 화자인식, STT, 앱 연결, 그 외 상상하시는 아무거나...

- Intelligence Service 1
- Intelligence Service 2
- Intelligence Service 3
- Intelligence Service 4
- .....

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

마이크 입력을 사용할 기본 요소?

- 배경/환경인식
- 음성인식
- 화자인식
- 감정인식

- Intelligence Service 1
- Intelligence Service 2
- Intelligence Service 3
- Intelligence Service 4
- ……

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

비디오 입력을 사용할 기본 요소?

물체감지 | 물체인식 | 물체추적

영역구분 | 안면인식 | 감정인식 V

안면추적

상황인식

배경/환경인식

음성인식

화자인식

감정인식

Intelligence Service 1

Intelligence Service 2

Intelligence Service 3

Intelligence Service 4

⋮

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

오디오 비디오 입력을 함께 사용할 기본 요소?

물체감지 | 물체인식 | 물체추적

영역구분 | 안면인식 | 감정인식V

안면추적

상황인식

배경/환경인식

음성인식 | 감정인식AV

화자인식 | 화자추적AV

감정인식 | 상황인식AV

Intelligence Service 1

Intelligence Service 2

Intelligence Service 3

Intelligence Service 4

⋮

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

오디오 비디오 미디어 전처리

물체감지 | 물체인식 | 물체추적

영역구분 | 안면인식 | 감정인식V

안면추적

상황인식

배경/환경인식

음성인식 | 감정인식AV

화자인식 | 화자추적AV

감정인식 | 상황인식AV

Intelligence Service 1

Intelligence Service 2

Intelligence Service 3

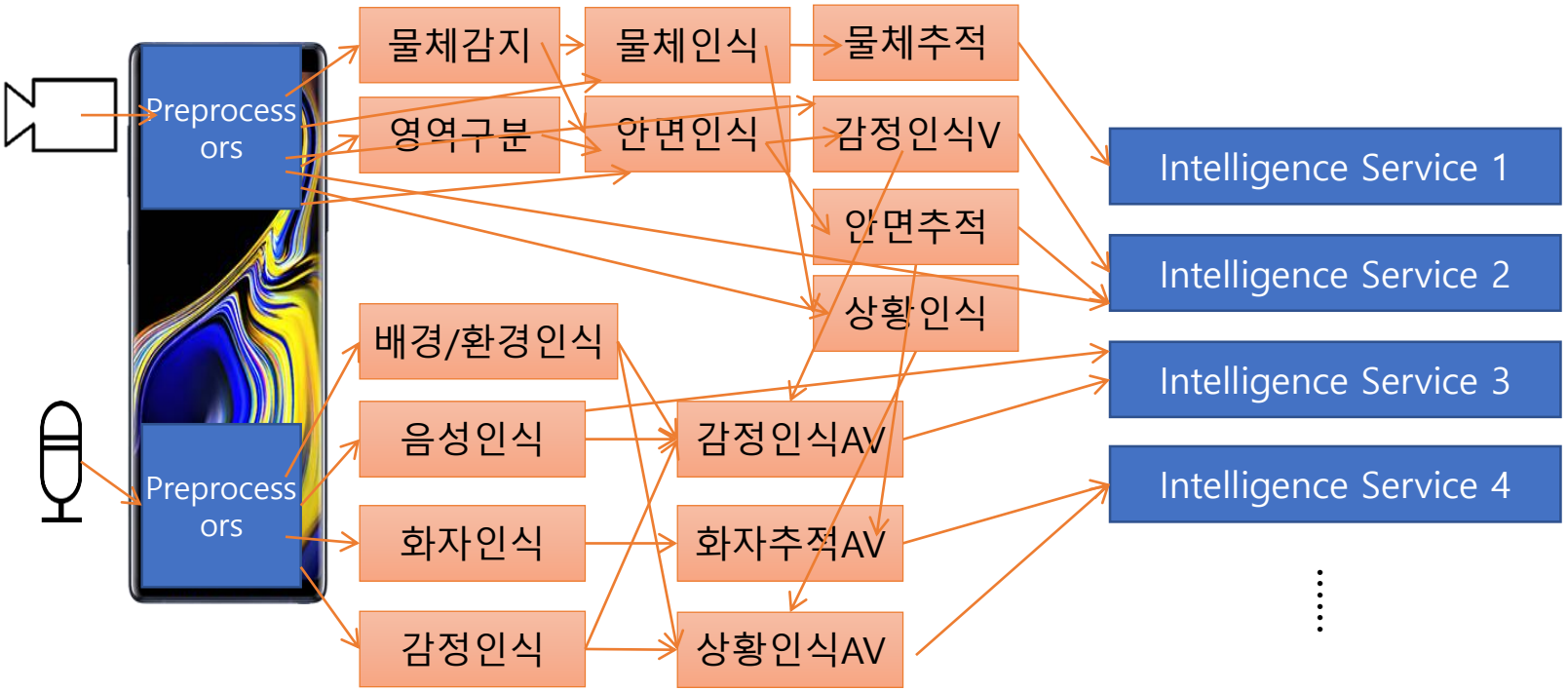
Intelligence Service 4

⋮

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

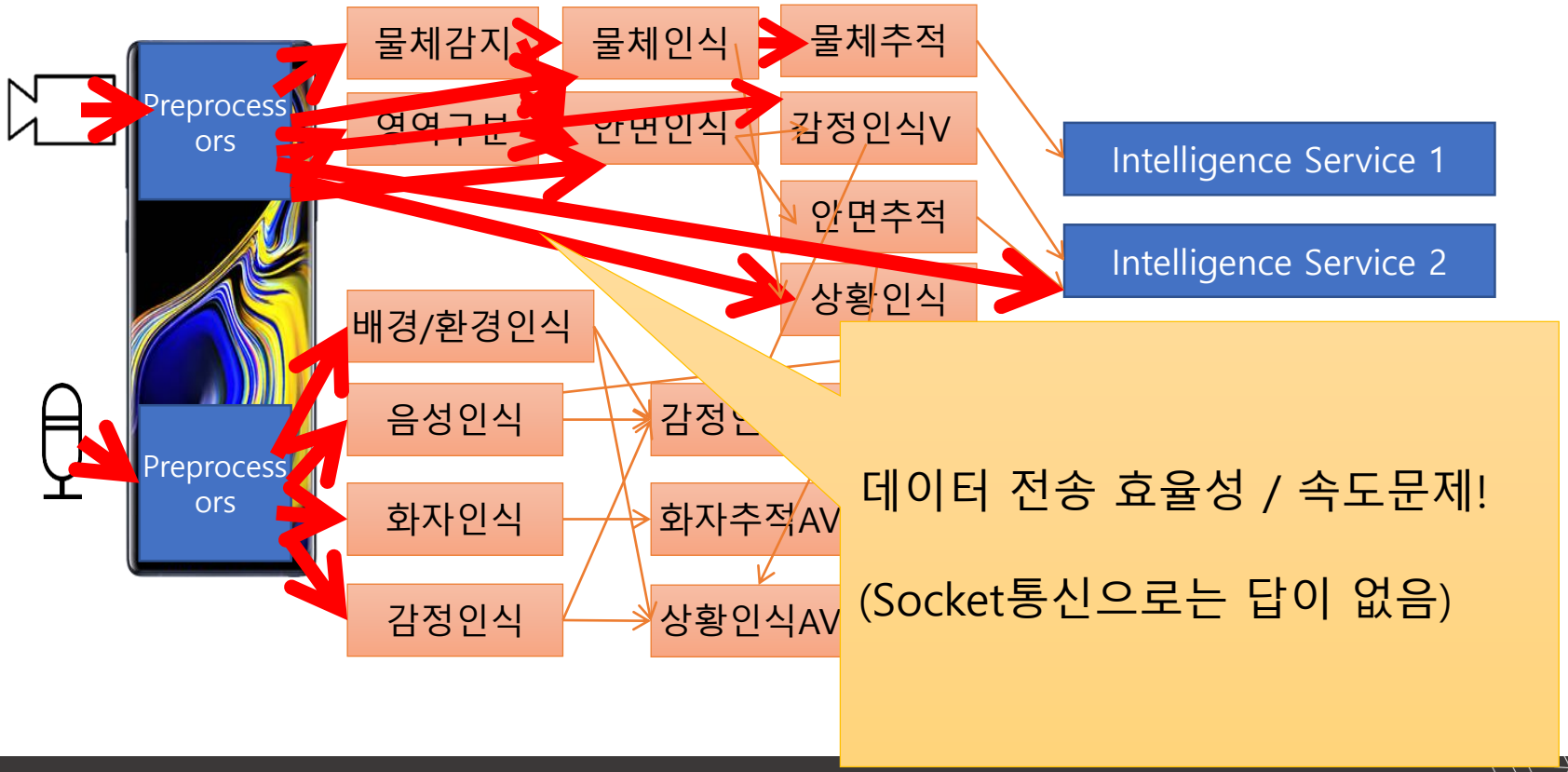
Data Stream Pipelines



Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 1. Data Stream Performance.

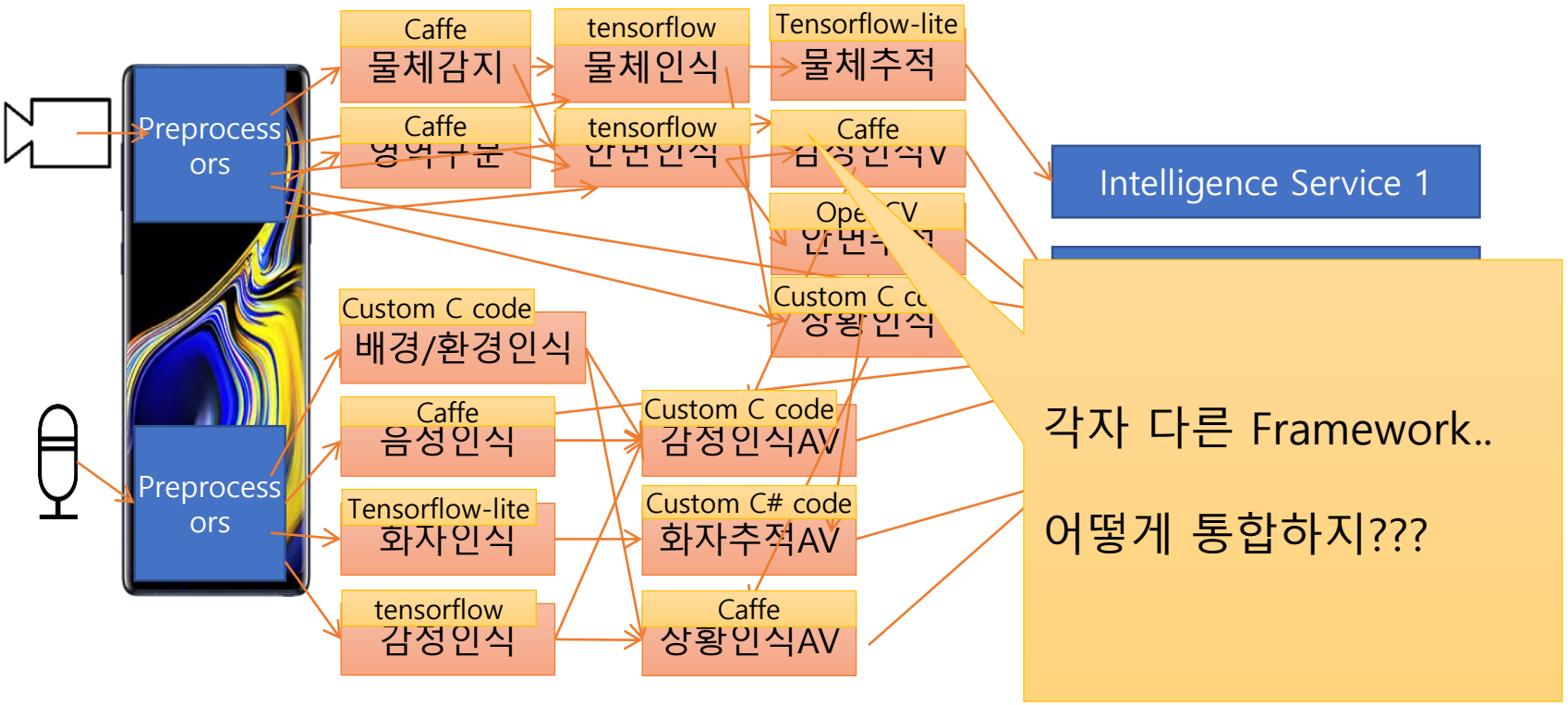


Case 2: Multi-Modal Interactions

SAMSUNG Research

Robots, Consumer Electronics, Mobile Devices

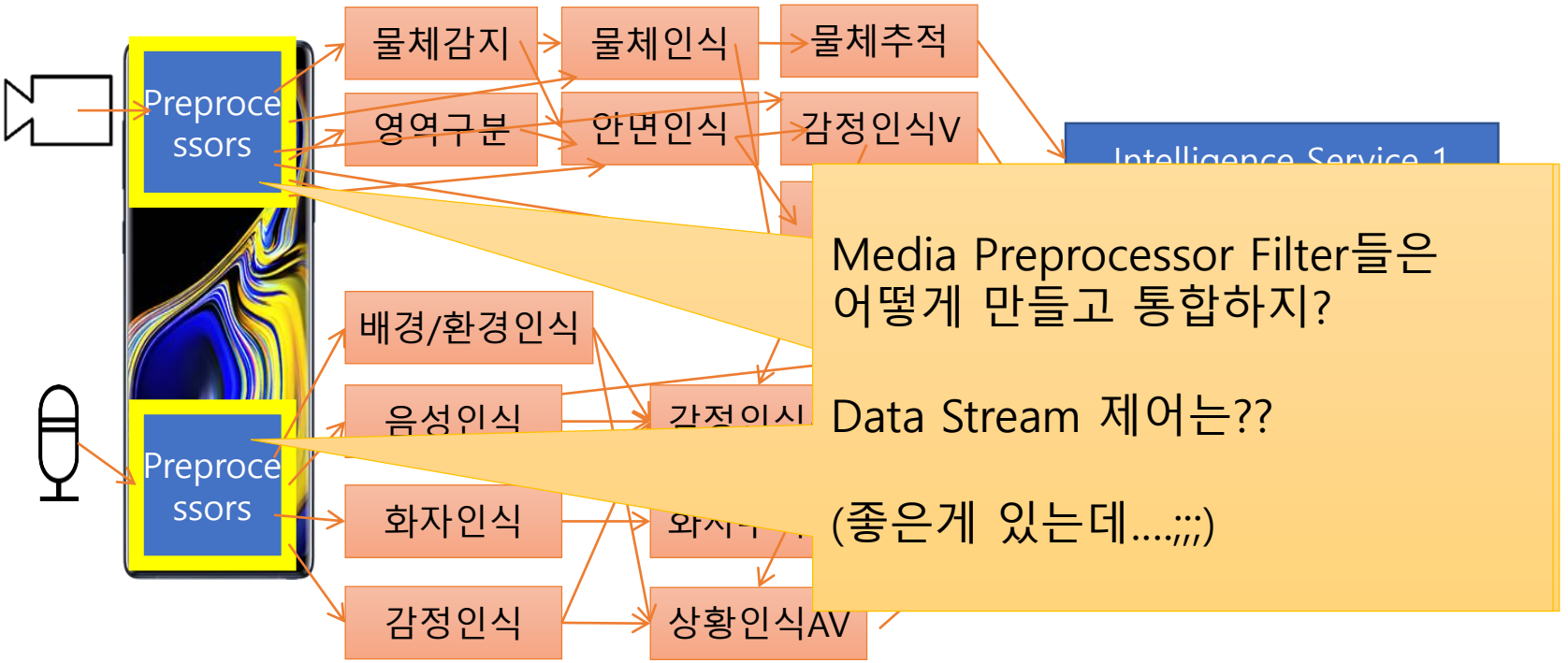
Issue 2. Neural Network Frameworks



Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 3. Multimedia Data Processing



설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 4. Data Stream Synchronization

Model A

Model B

Model C

Intelligence Service 4

A, B가 다른 속도로 동작하면
C에서는 어떻게 받아들일 것인가?

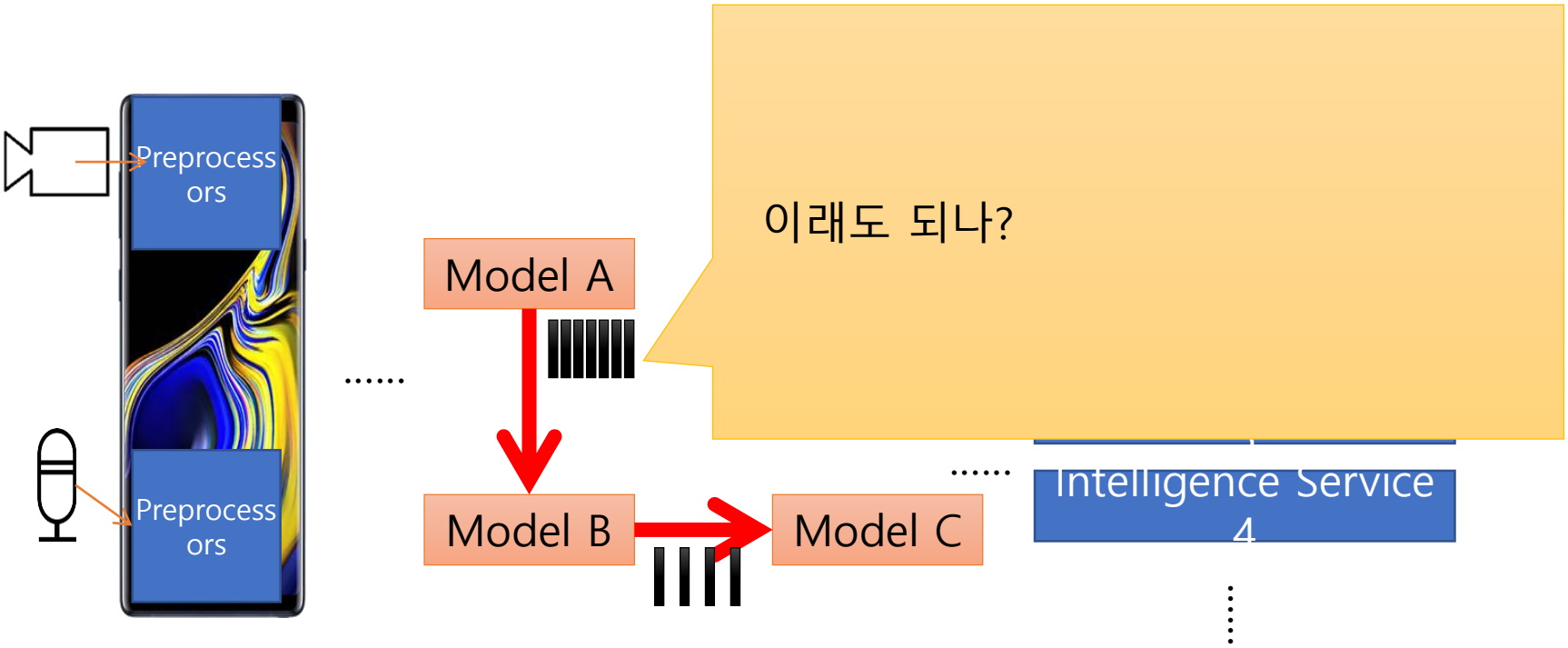
A, B의 출력 시기가 일정하지 않으면
C에서 어떻게 받아들일 것인가?

설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

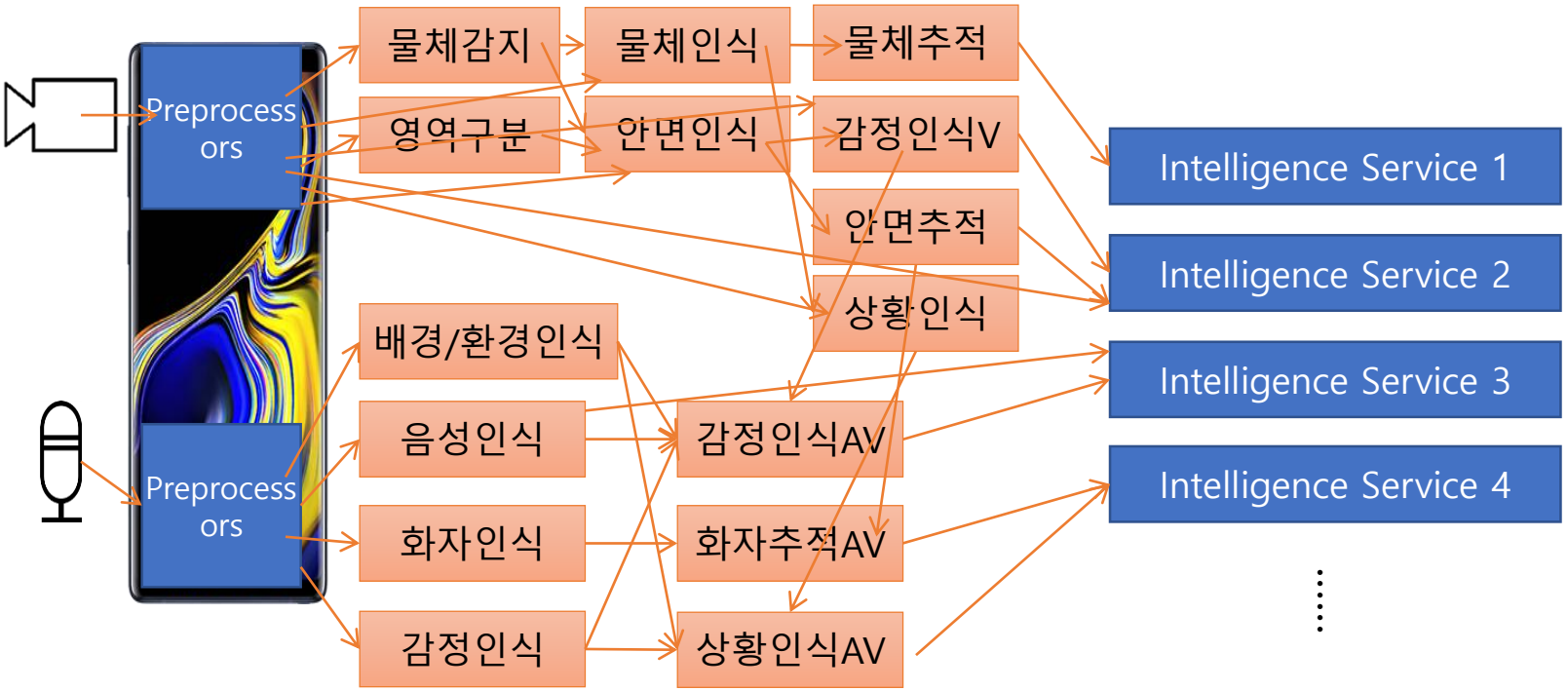
Issue 5. Data Stream Rate Control



Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

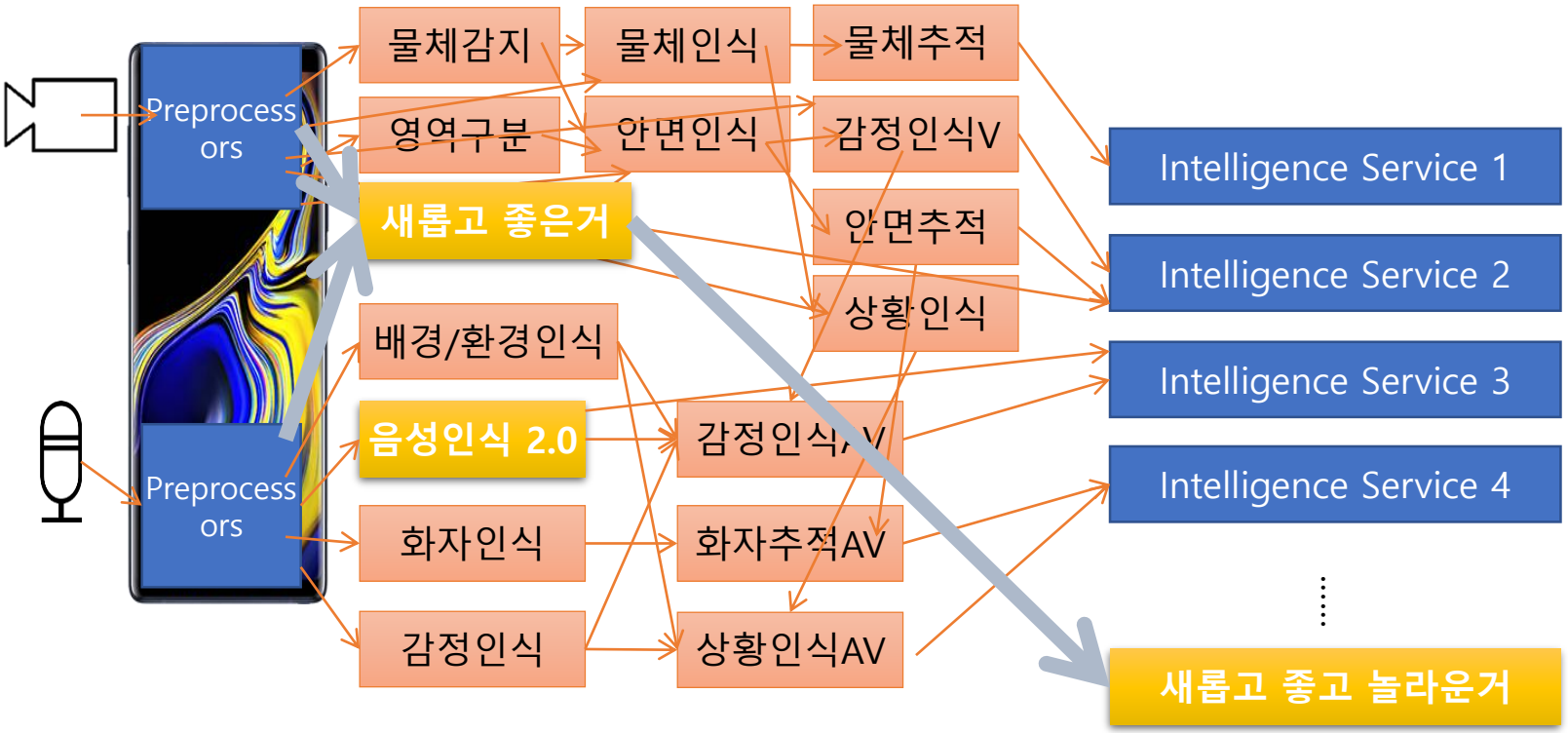
Issue 6. Dynamic Stream Pipelines



Case 2: Multi-Modal Interactions

Robots, Consumer Electronics, Mobile Devices

Issue 6. Dynamic Stream Pipelines

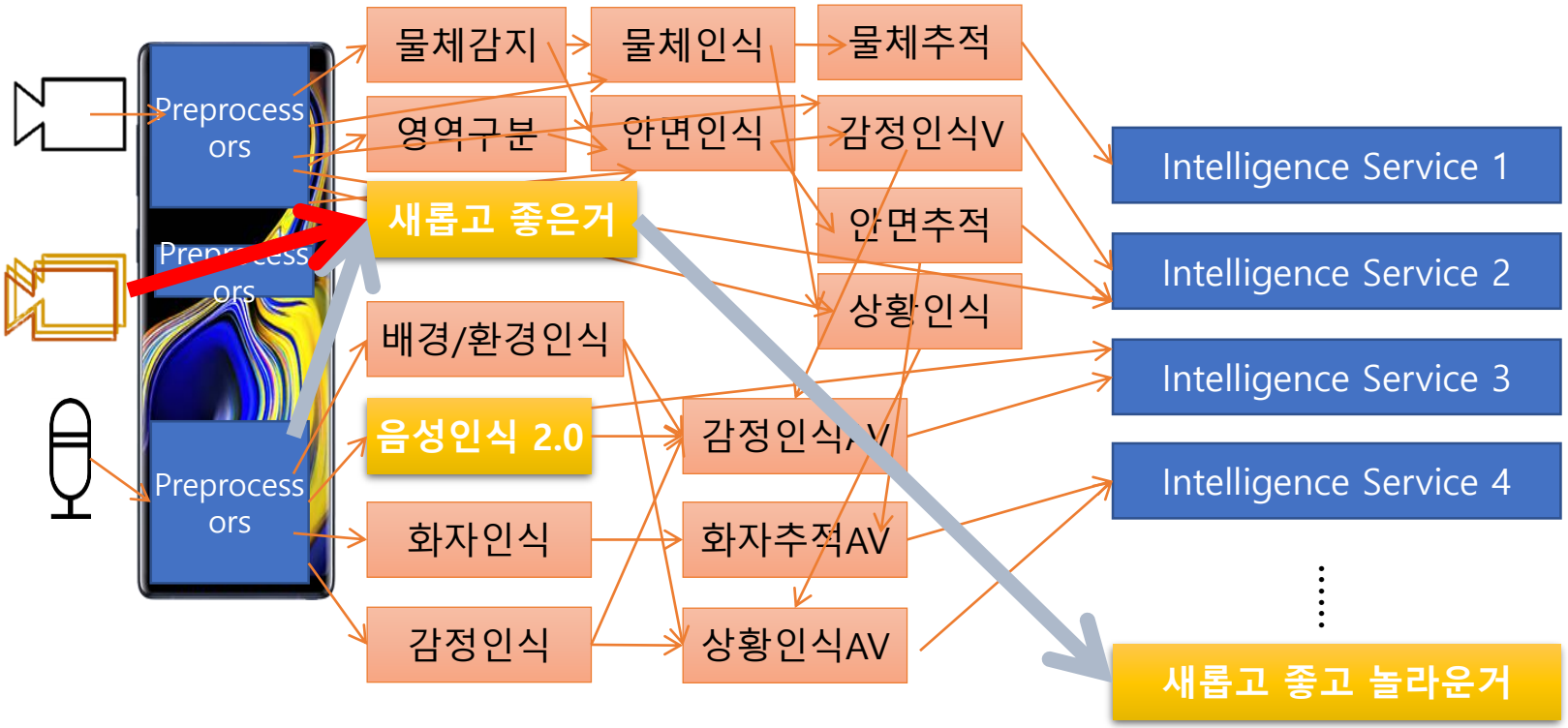


설명을 위한 가상 예제입니다.

Case 2: Multi-Modal Interactions

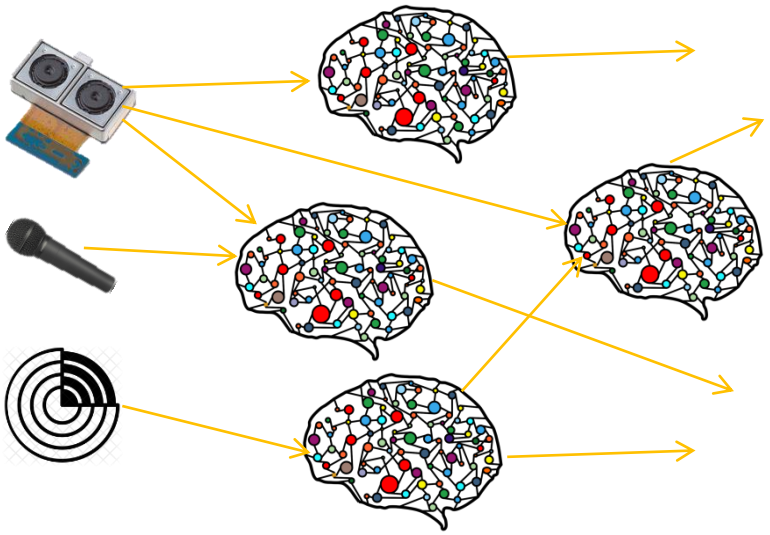
Robots, Consumer Electronics, Mobile Devices

Issue 6. Dynamic Stream Pipelines



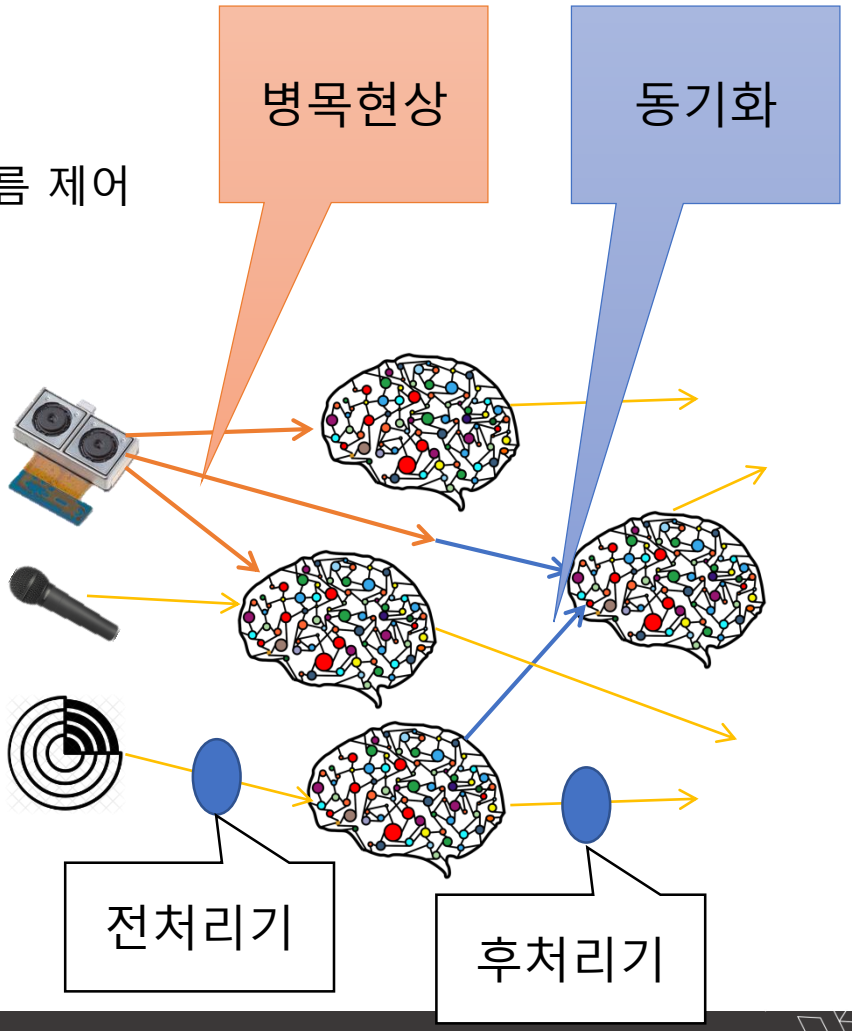
Recap: On-Device AI 환경

- 단말 내 Sensor 입력의 실시간 처리
  - Stream 형태의 Data 처리
  - 실시간 전처리/후처리 필요
- Sensor / Data Stream 다수 존재
- Neural Net 및 서비스 다수 존재
  - 여러 Neural Net의 협력
  - 여러 Neural Net 기반 서비스의 탑재

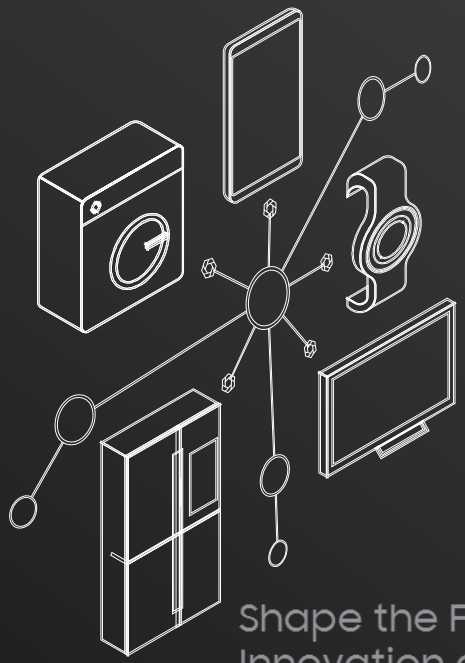


Recap: On-Device AI S/W 개발 이슈

- 개발 노력 과다 / 성능문제 발생
  - 데이터 전-후처리기 개발, 데이터 흐름 제어
  - 데이터 동기화, 병목현상 분석
  - 각 센서/신경망/전-후처리기 연결



Contents



Shape the Future with
Innovation and Intelligence

https://pixabay.com/photo-2490210/

DO NOT REINVENT THE WHEEL

https://pixabay.com/photo-2490210/

Do NOT Reinvent the Wheel

Do NOT Reinvent the Wheel

gstreamer for intelligence!

What is gstreamer?

- Pipeline 기반 Open Source Multimedia Framework (2001.1.~)
- Linux 계열 표준
  - Linux, BSD, macOS, iOS, Windows, ….
  - TV: 삼성, LG, Sony, …
- Audio/Video 재생, 기록, 스트리밍/방송, 변환, 편집 등
  - Stream Filter로 사용 가능한 수백가지 Plugin 제공.
  - Plugin을 제작하기 위한 환경 포함
  - C/C++/C#/Java/Python/Perl/CLI/… 에의 표준 API 제공
  - Pipeline 구성을 표현하는 것 만으로 동작하는 Application이 완성됨!
    - GUI Tool로 Pipeline 그려서 만들기도…

Do NOT Reinvent the Wheel

gstreamer for intelligence!

Why gstreamer?

- Data Stream Pipeline 을 표현, 관리, 실행하는
  안정된 표준 Framework
- 언급된 Issue는 이미 수십년 된 Multimedia의 주요 Issue!
  - 이미 해결한 문제 다시 풀지 말자!
- 신경망 == Data Stream Filter!

Do NOT Reinvent the Wheel

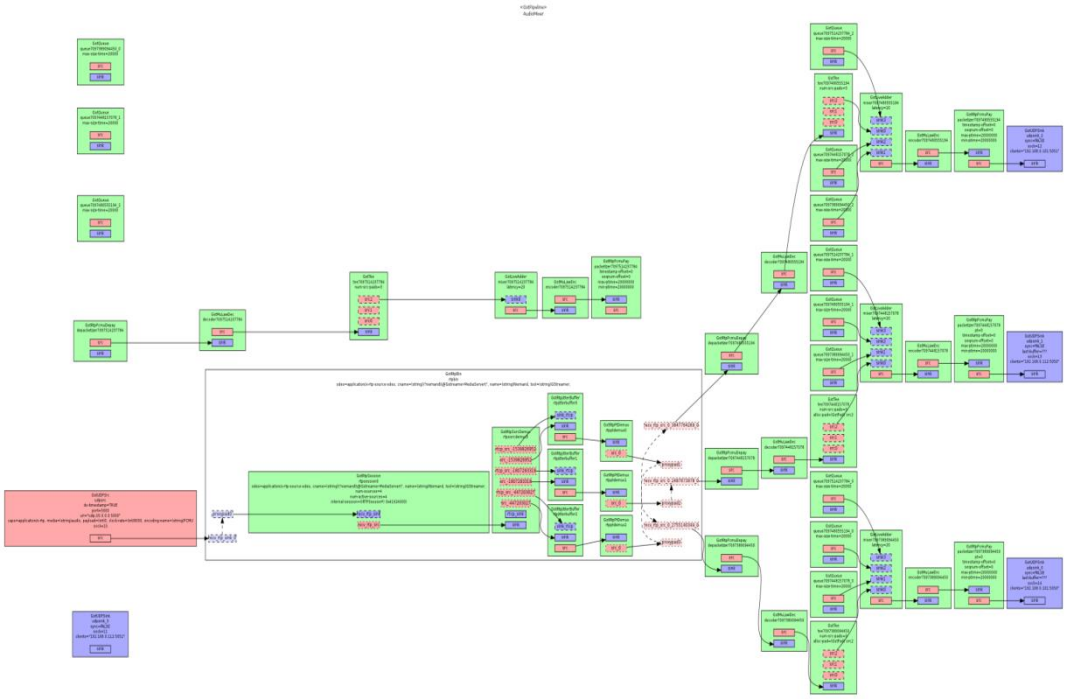
gstreamer

SAMSUNG Research

gstreamer for intelligence!

Why gstreamer?

- Configure & run complex stream pipelines, dynamically in run-time!



Do NOT Reinvent the Wheel

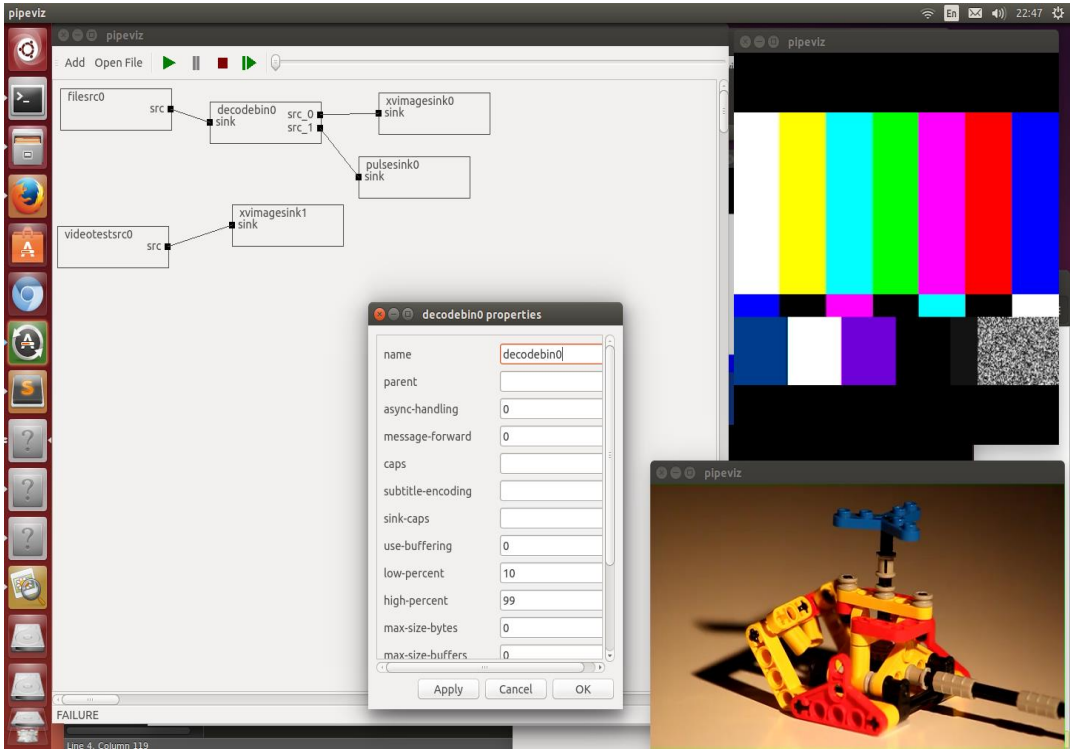
gstreamer

SAMSUNG Research

gstreamer for intelligence!

Why gstreamer?

- Configure & run complex stream pipelines… even with GUI tools



Don’t Reinvent the Wheel, PERFECT THE WHEEL!

https://pixabay.com/photo-2

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되도록…

- 다양한 신경망 Framework / Model 들에게 공통으로 주어질 Data Stream 표준
- 다양한 신경망 Framework / Model 들을 Invoke할 방법
- Audio/Video/Text 외 다른 Data Stream을 GStreamer에서 다루도록 할 방법

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되도록…

- 다양한 신경망 Framework / Model 들에게 공통으로 주어질 Data Stream 표준
- 다양한 신경망 Framework / Model 들을 Invoke할 방법
- Audio/Video/Text 외 다른 Data Stream을 GStreamer에서 다루도록 할 방법

➔ nnstreamer

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

➔ nnstreamer

License: LGPL 2.1

기존 GStreamer LGPL 코드 재사용

Public Open: 2018.7. (v0.0.1) …… 2019.9. (v0.3.0)

https://github.com/nnsuite/nnstreamer

Please add “STAR” ☺

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되면…

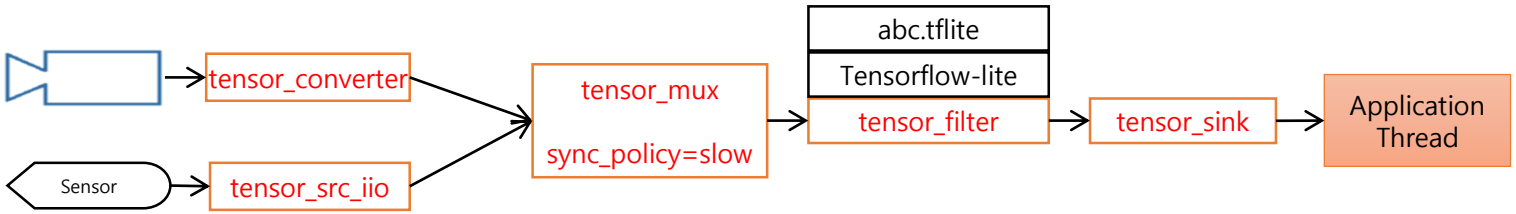
- 신경망 알고리즘 개발자
- 신경망 사용 디바이스/애플리케이션 개발자
- 멀티미디어 개발자

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되면…

- 신경망 알고리즘 개발자
  - 쉽고 즉각적인 통합 ➔ 빠른 테스트 / 프로토타이핑
  - 예:




```
$ gst-launch-1.0
        v4l2src name=cam_src ! tensor_convert ! mux.sink_0 \
        tensor_source_iio device=name device-number=100 trigger=abc ! mux.sink_1 \
        tensor_mux sync_option=slow ! \
            tensor_filter framework=tensorflow-lite model=abc.tflite ! \
            tensor_sink name=sink
```

Don’t Reinvent the Wheel, Perfect it!

gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되면…

- 신경망 사용 디바이스/애플리케이션 개발자
  - 복잡한 Data Stream Pipeline 의
    - 표현이 쉽다
    - 변경, 관리도 쉽다
    - 실행이 효율적이고 빠르다
    - 구성하는 알고리즘/모델의 교체도 쉽다.

    } 저비용! 고효율!
  - 다양한 언어/OS에 표준 API 지원
    - C, C++, C#(.NET), Java, Python, Perl, Rust, D, Qt, Haskell, Ruby, CLI (shell)
    - Linux, BSD, Solaris, Android, macOS, iOS, Windows, OS/400
  - 다양한 Input 과 Media Filter의 재사용
  - 기존 SW Platform의 재사용 / 통합.

Don’t Reinvent the Wheel, Perfect it!

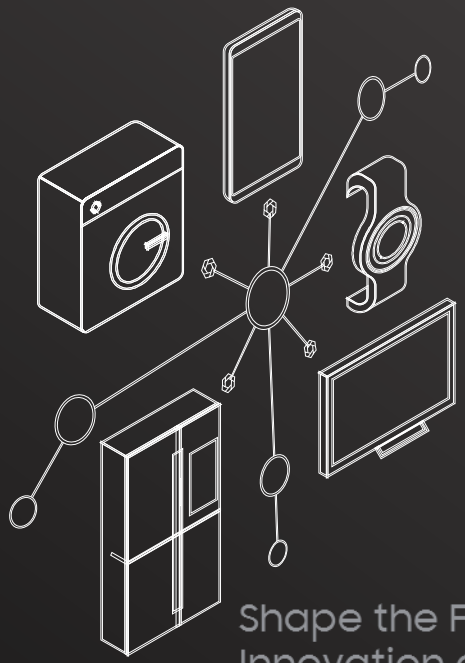
gstreamer for intelligence ➔ **nnstreamer**

Gstreamer가 신경망을 이해하여 AI Device의 기반이 되면…

- 멀티미디어 개발자
  - 신경망 알고리즘을 Media Filter로 사용하자!

    화질 개선, Context 인식, Style 변형, ……

Contents



Shape the Future with
Innovation and Intelligence

How GStreamer handle neural networks?

신경망도 또 하나의 Media Filter 일뿐.

그럼 신경망 사이의 데이터는 Video Stream? Audio Stream? Text Stream? …

- ➔ “Tensor Stream”!

- Tensor : Multi-Dimensional Array

How GStreamer handle neural networks?

신경망도 또 하나의 Media Filter. Tensor도 또 하나의 Media Type.

Data Stream으로서의 “Tensor”에 대한 정의

```
other/tensor
          framerate: (fraction) [ 0/1, 2147483647/1 ]
          dimension: [1, 65535]:[1, 65535]:[1, 65535]:[1, 65535]
          type: (string) { uint8, int8, uint16, int16, uint32, int32, uint64, int64,
float32, float64 }
```

- Refer: https://github.com/nnsuite/nnstreamer/wiki/AD%3A-Data-Type%2C-Caps%2C-and-Flow-Design “**GStreamer data types (pad capabilities)**”

Tensors: Tensor의 묶음

신경망도 또 하나의 Media Filter. Tensor도 Tensors도 또 하나의 Media Type.

Data Stream으로서의 “Tensors”에 대한 정의

```
other/tensors
          num_tensors = (int) [1, 16] # GST_MAX_MEMCHUNK_PER_BUFFER
          framerate: (fraction) [ 0/1, 2147483647/1 ]
          types = (string) Typestrings
          dimensions = (string) Dimensions

Typestrings = Typestring | Typestring, Typestrings
Dimensions = Dimension | Dimension, Dimensions

Typestring = (string) {float32, float64, . . . , int16, uint16, int8, uint8}
Dimension = (string) [1-65535]:[1-65535]:[1-65535]:[1-65535]
```

- Refer: https://github.com/nnsuite/nnstreamer/wiki/AD%3A-Data-Type%2C-Caps%2C-and-Flow-Design “**GStreamer data types (pad capabilities)**”

Neural Network를 Stream Pipeline에 붙이기

- **Neural Network 도 하나의 Stream Filter 일 뿐..**


```
Code: src ! sink
```

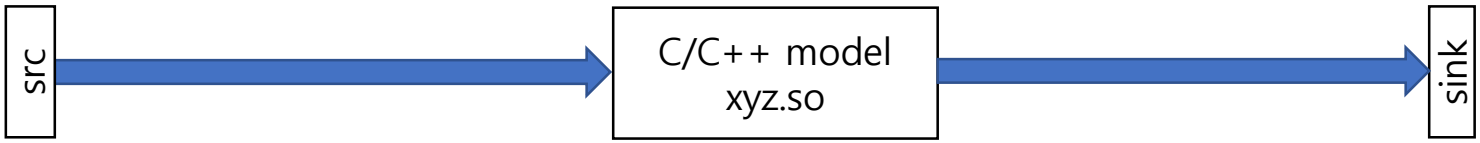

```
Code: src ! tensor_filter mode=tensorflow-lite model=abc.tflite ! sink
```


```
Code: src ! tensor_filter mode=tensorflow model=def.pb ! sink
```

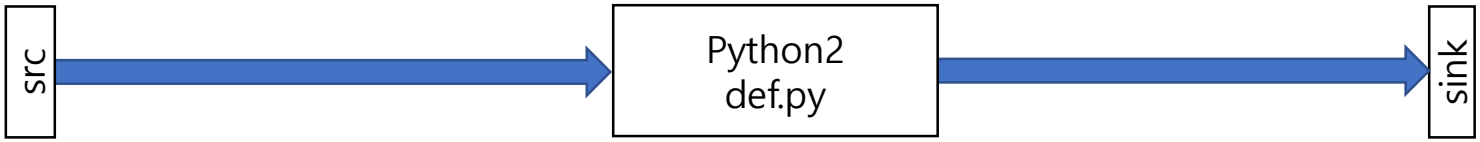

```
Code: src ! tensor_filter mode=caffe2 model=ghi.pb ! sink
```

Neural Network를 Stream Pipeline에 붙이기

- NN F/W 를 쓰지 않는 NN Model 들도 있어요...




```
Code: src ! tensor_filter mode=custom model=xyz.so ! sink
```

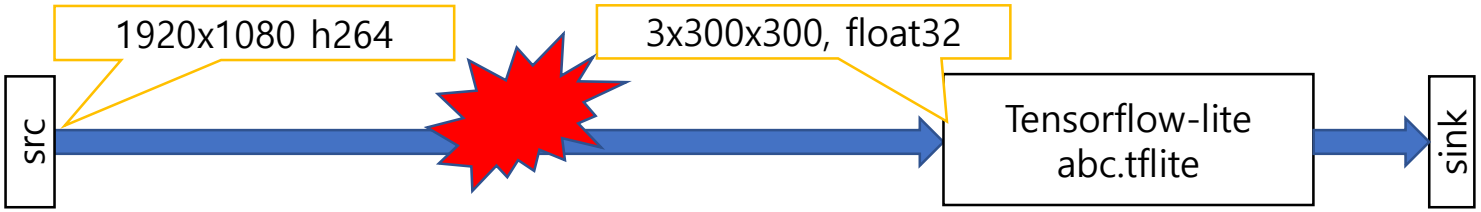



```
Code: src ! tensor_filter mode=python2 model=def.py ! sink
```

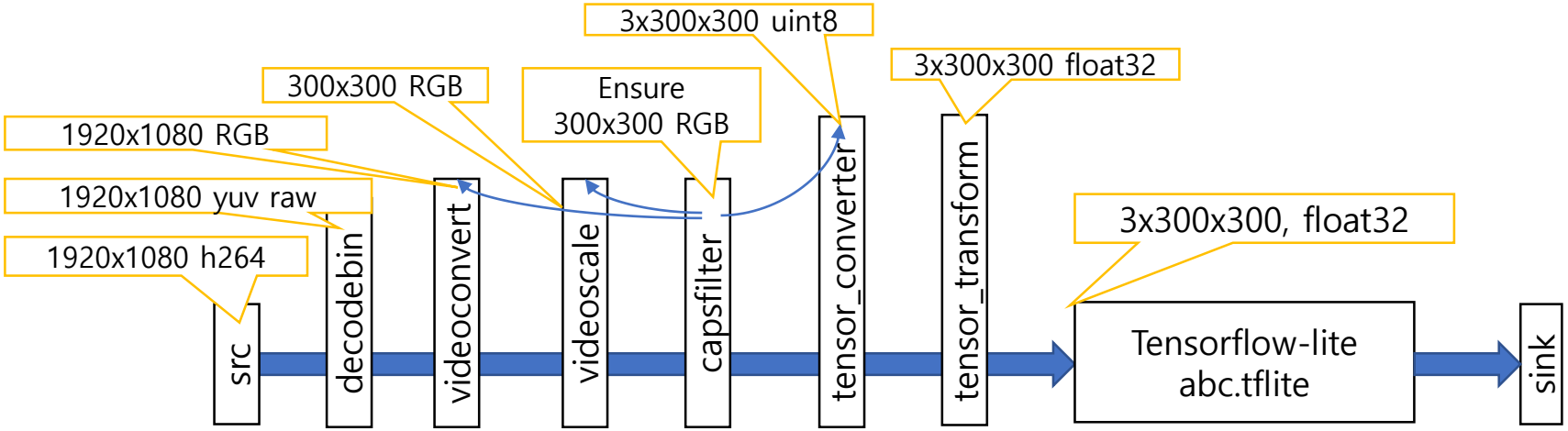

Need to wrap custom models with nnstreamer APIs

Neural Network를 Stream Pipeline에 붙이기

- NN 에게 데이터를 주기 위해 미리 준비 할 일들 처리




```
Code: src ! tensor_filter mode=tensorflow-lite model=abc.tflite ! sink
```

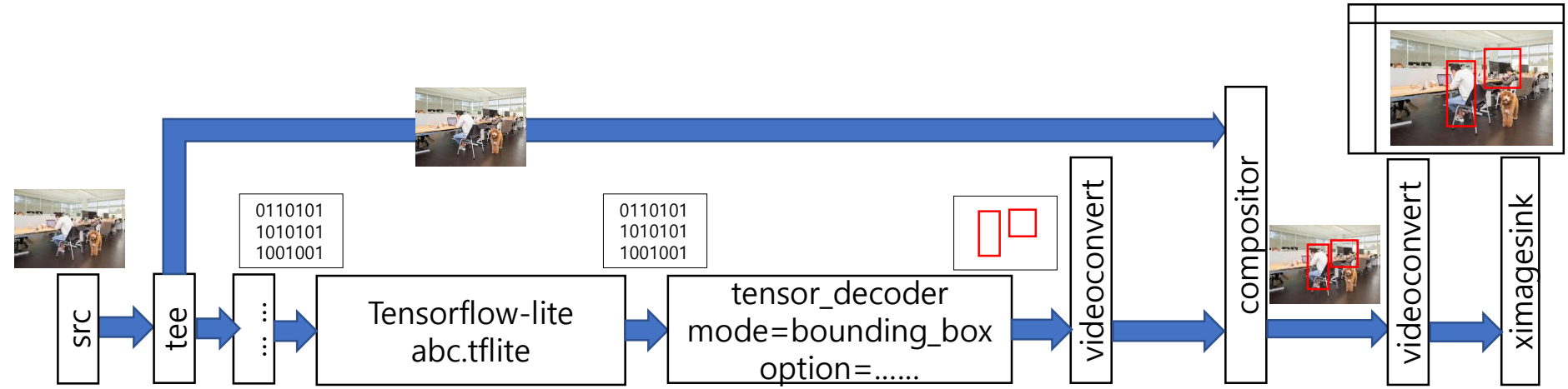



```
Code: src ! decodebin ! videoconvert ! videoscale
          ! video/x-raw,format=RGB,width=300,height=300
          ! tensor_convert
          ! tensor_transform mode=typechg option=float32
          ! tensor_filter mode=tensorflow-lite model=abc.tflite ! sink
```

Neural Network를 Stream Pipeline에 붙이기

- GUI로 데모도 해보고 싶은데요... 다른 App Code 만들지 말고...

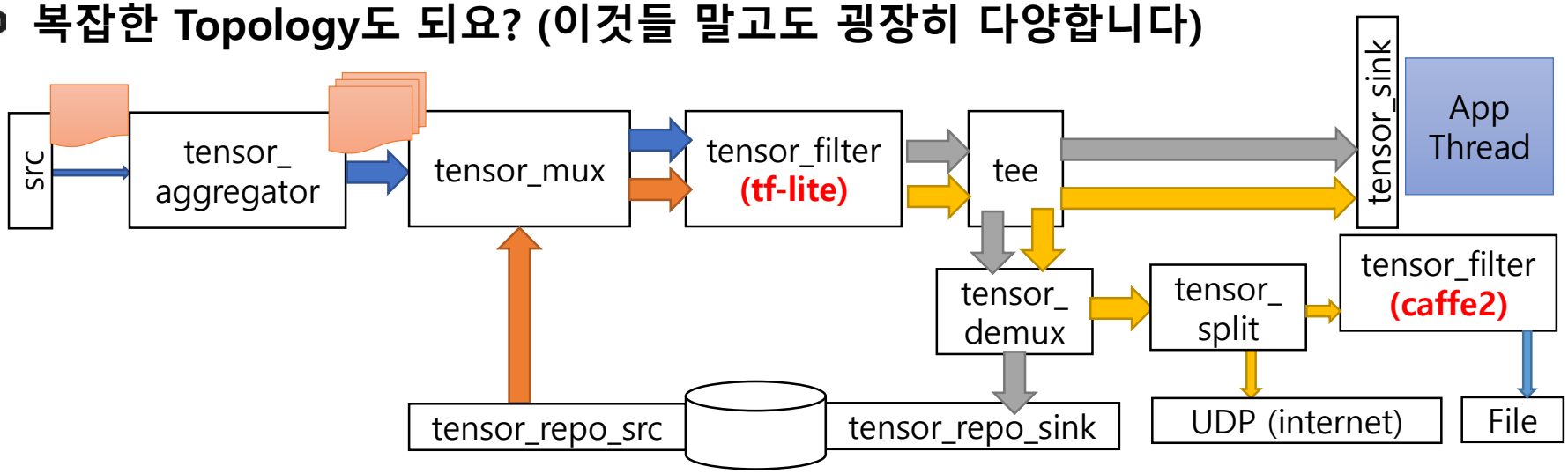

```
Code: src ! ... ! tensor_filter mode=tensorflow-lite model=abc.tflite ! sink
```




```
Code: src ! tee name=t
        t. ! queue ! ... ! tensor_filter mode=tensorflow-lite model=abc.tflite
           ! tensor_decoder mode=bounding_box option=... ! videoconvert
           ! compositor name=c sink_0::zorder=2 sink_1::zorder=1
           ! videoconvert ! ximagesink
        t. ! queue ! c.
```

Neural Network를 Stream Pipeline에 붙이기

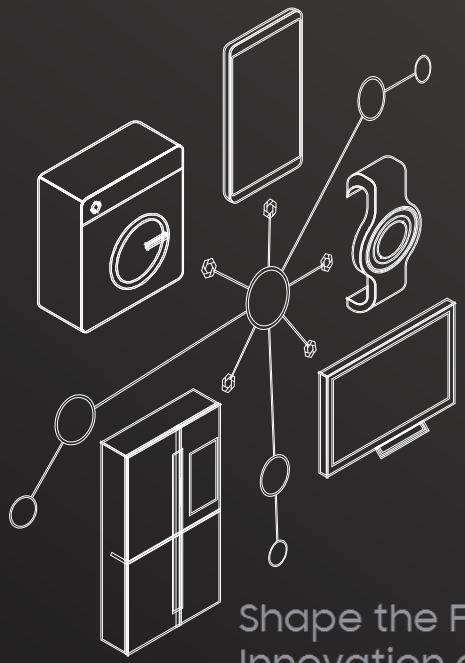
- 복잡한 Topology도 되요? (이것들 말고도 굉장히 다양합니다)




```
Code: src ! tensor_aggregator frames-out=3 ! tensor_mux name=m
          ! tensor_filter mode=tensorflow-lite model=a.tflite ! tee name=t ! queue
          ! tensor_demux name=d
      d.src_0 ! tensor_repo_sink name=a
      tensor_repo_src name=a ! m.sink_1
      d.src_1 ! tensor_split name=s option=... ! udpsink host=Samsung port=8000
          s.src_1 ! tensor_filter mode=caffe2 model=b.pb ! filesink location=/tmp/tmp
      t. ! queue ! tensor_sink name=appsink


Keep adding filters, stream paths!
```

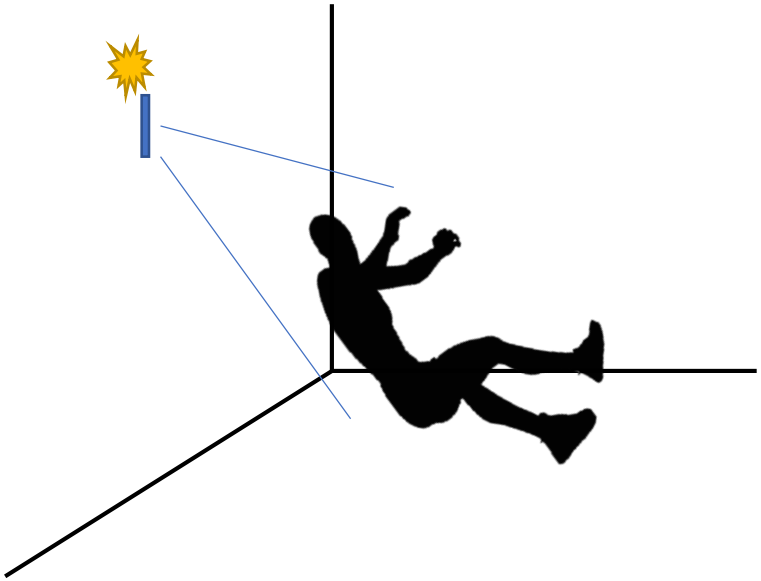
Contents



Shape the Future with
Innovation and Intelligence

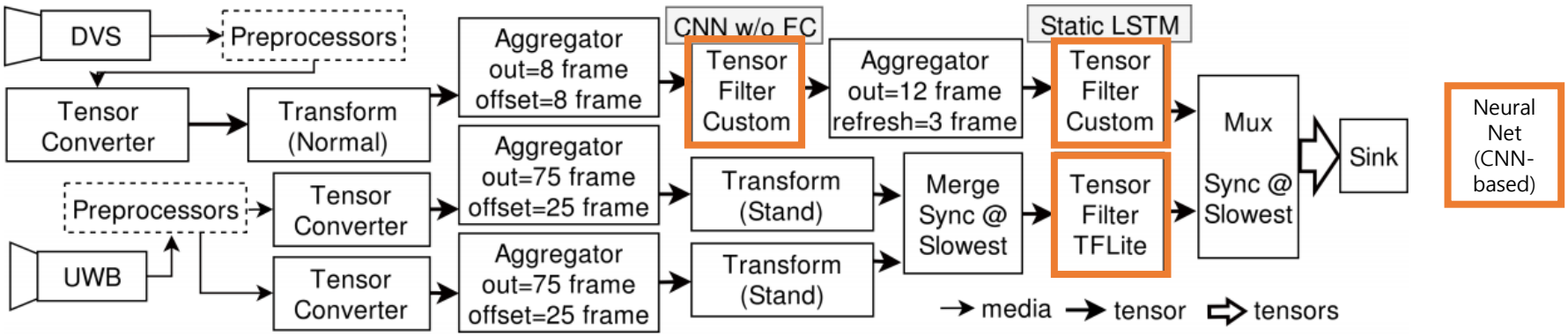
Activity Recognition Sensors

실 적용 사례



Images from: https://pixabay.com/ko/%EC%8B%A4%EB%A3%A8%EC%97%A3-%EA%B0%80-%EB%96%A8%EC%96%B4%EC%A7%80%EB%8A%94-%EC%B6%95%EC%86%8C-%EC%84%9C%EC%82%AC%EC%8B%9C-%EB%94%94%EC%9E%90%EC%9D%B8-%EC%95%84%EC%9D%B4%EC%BD%98-%EA%B8%88%EC%9C%B5-3172280/

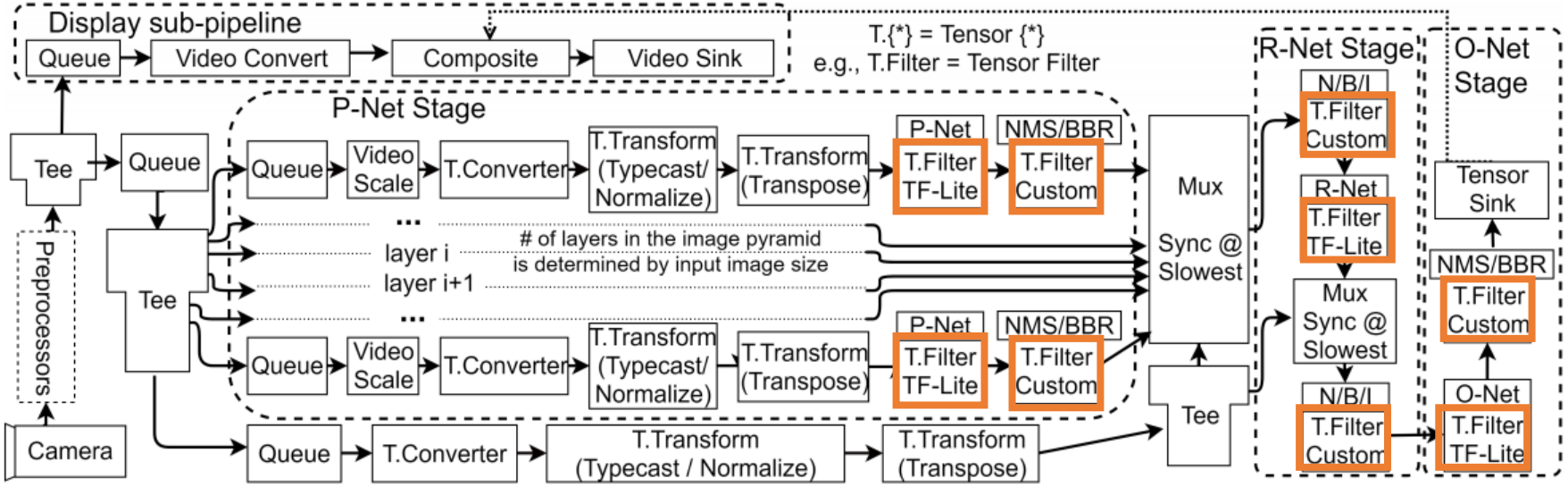
Activity Recognition Sensors



- ~1000 lines → 16 lines
- @ 30FPS input, 90.4% → 51.4% CPU
- ~40 MiB → ~17 MiB Memory (RSS)

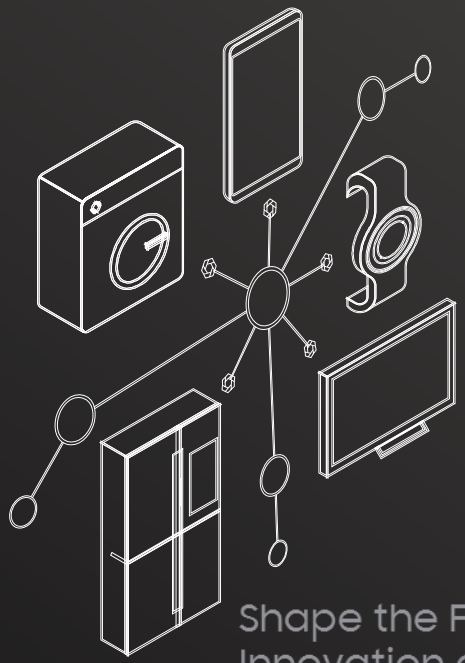
```
$ gst-launch-1.0 tensor_mux name=mux ! fakesink
! tensor_converter ! tensor_trans mode=arith
! tensor_aggregator in=1 out=8 flush=8
! tensor_filter frame=custom m=./cnn.so
! tensor_aggregator in=1 out=12 flush=3
! tensor_filter frame=custom m=./lstm.so
! mux.sink_0
tensor_merge name=merge sync-mode=slowest
! tensor_filter frame=tflite m=./uwb.tflite
! mux.sink_1
multifilesrc location="./input_uwb0_%04d.data"
! tensor_converter dim=1:1:32:1 type=float32
! tensor_aggregator in=1 out=75 flush=25
! tensor_trans mode=stand ! merge.sink_0
multifilesrc location="./uwb1_%04d.data"
! tensor_converter dim=1:1:32:1 type=float32
! tensor_aggregator in=1 out=75 flush=25
! tensor_transform mode=stand ! merge.sink_1
```

MTCNN for Face Detection & Alignment



- **Easier to implement**
  - Easier Parallelism & Synchronization
- **Higher performance**
  - Throughput +83.2%
  - Latency +15.4%

Contents



Shape the Future with
Innovation and Intelligence

Related Work

- **NNStreamer: 2018.7.~**
  - Production ready for Tizen & Android.
  - Integration w/ OS Standard APIs/Packages
- **DeepStream: 2018.7.~ (NVidia, Gstreamer-based)**
  - For Nvidia H/W.
- **GST-MSDK: 2018.9.~ (Intel, Gstreamer-based)**
  - For Intel H/W. Not actively maintained.
- **Core-ML 2: 2018.9.~ (Apple)**
  - Limited pipelining. Closed-source
- **Media Pipe: 2019.6.~ (Google)**
  - Not production-ready?

NNStreamer 0.3.0 (1.0-rc1)

- **Supported Platform**
  - Tizen 5.5+
    - Standard machine learning API
  - Ubuntu 16.04/18.04
    - PPA: `ppa:nnasteramer/ppa`
  - Android
    - Android Studio: JCenter, "nnstreamer"
    - Play Store: search "nnstreamer"
  - Yocto/OpenEmbedded
    - "neural-network" layer standard framework.
  - macOS
    - Brew: nnsuite/nnstreamer
  - Incoming: iOS, Windows

NNStreamer 1.0.0 (1.0-rc2)

Neural Network Frameworks

- 완료: TensorFlow (1.9~1.14), TensorFlow-lite (1.9~1.14)

  Custom C Model, Custom Python2/3 Model,

  Caffe2, PyTorch, Movidius-X (NCSDK),

- WIP: nnfw-runtime, openVINO
- 계획: Tensorflow (2.x), Keras

입출력 관련 Frameworks

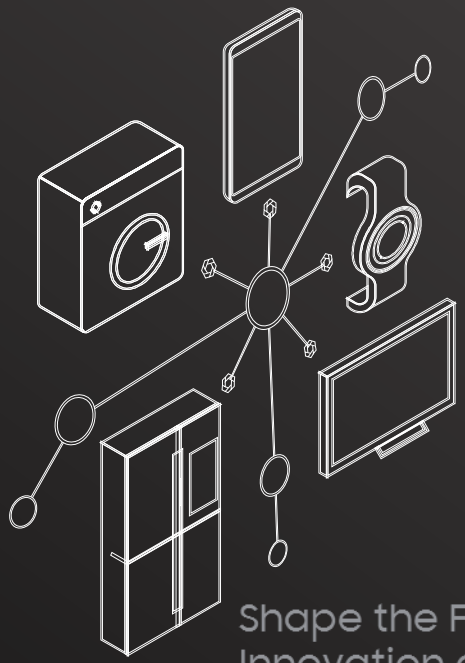
- 완료: ROS, Linux-IIO

- 계획: Protobuf, Flatbuffer, ROS2

ROS.org

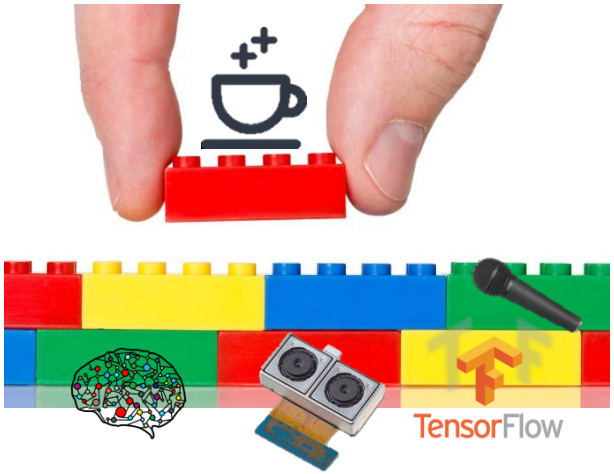
ROS2

Contents



Shape the Future with
Innovation and Intelligence

NNStreamer Benefit



- 쉬운 구현
  - 부품간의 연결관계의 표현으로 App/System 작성
  - 기존 고성능 미디어 처리기와 스트림 제어의 재사용
    - 대부분의 관련 문제는 이미 Multimedia 개발자들이 지난 십 수년간 고민하고 해결해온 문제임.
- 최적화된 구동
  - Data Pipeline Parallelism을 통한 병렬화
  - Memcpy-less Data 전송
  - Stream 간 동기화 및 속도 조절

보다 적은 비용으로

더 다양한 기능을

저가의 기기에 탑재

Open Source Project!

누구나 자유롭게 쓰실 수 있습니다.

- License: LGPL 2.1
  - NNStreamer를 Shared Library로 사용한 App은 별도 코드 공개 의무 없음
  - License 고지 의무
- 상업적 사용도 자유로움.

누구나 자유롭게 참여하실 수 있습니다.

- https://github.com/nnsuite/nnstreamer
  - Github Issue를 통한 토론, 문답, 정보 공유
- 코드 추가/수정 대환영!
  - Code of Conduct: https://github.com/nnsuite/nnstreamer/blob/master/CODE_OF_CONDUCT.md
  - How to Contribute: https://github.com/nnsuite/nnstreamer/blob/master/CONTRIBUTING.md
  - Pass all test cases (coding rule/style, unit tests, integration tests (Tizen, Ubuntu, Android, Yocto), static analysis, …) and code reviews
  - https://sethrobertson.github.io/GitBestPractices/
    Commit Often, Perfect Later, Publish Once!

Samsung Research: https://research.samsung.com/

NNStreamer DevTeam

일하는 곳

- https://github.com/nnsuite
- Samsung Research (서초구 우면동)

Software Release

- 매월 회식일.
- https://github.com/nnsuite/nnstreamer/wiki/NNStreamer-Conferences

분야

- System Software
  - Linux kernel, system libraries, frameworks, firmware
  - LinuxCon/ELCx, ATC ’15, ‘16, Eurosys ‘16, …
- Software Engineering
  - CI/CD systems, OS integration & release
  - ICCE ‘19, ICSE ‘19, …

SAMSUNG Research

Thank you