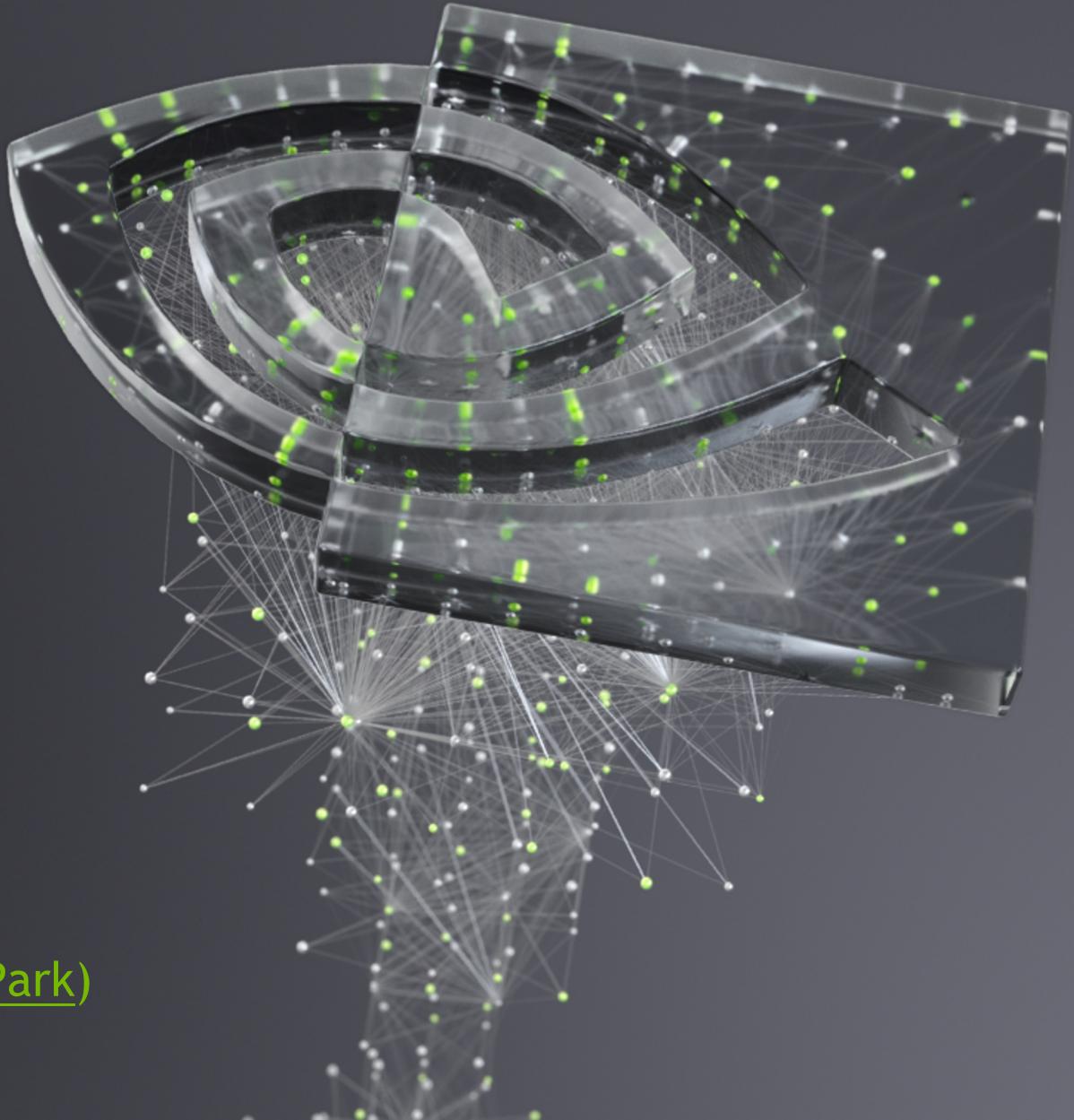




Introduction to the DeepStream SDK

Building a DeepStream Application
(Material and presentation: [Jeiyoon Park](#))



오늘 목표

- I. DeepStream SDK에 대한 설명
2. DeepStream을 활용한 Video Object Detection 간단한 예제

우와...



목차 – Introduction

- 딥러닝이 어려운 이유
- DeepStream

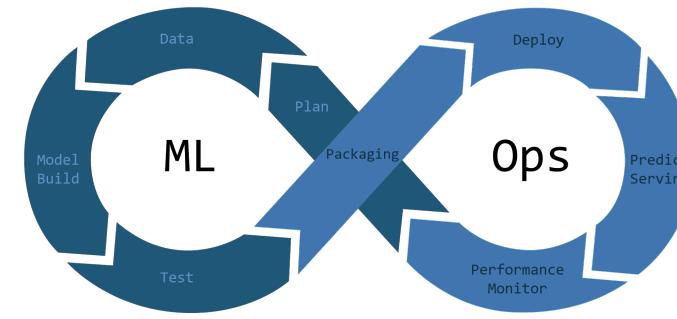


딥러닝이 어려운 이유

딥러닝이 어려운 이유

1. 방대한 전문지식을 가진 전문가들이 많이 필요함

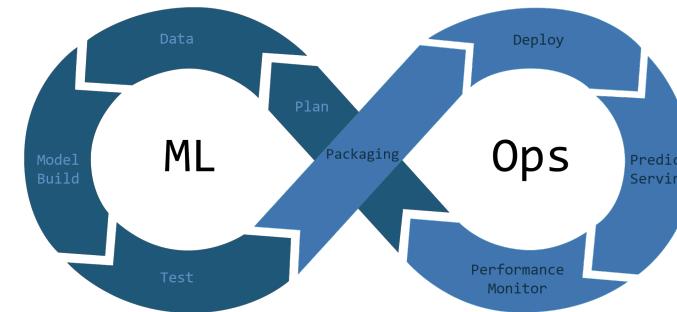
- 딥러닝 기본 이론
- Tensorflow, Pytorch framework를 능숙하게 다룰 줄 알아야함
- 딥러닝 모델을 최적화 하고 배포하기 위한 엔지니어링 지식
- 새로운 모델을 연구하고 개발하기 위한 리서치 능력



딥러닝이 어려운 이유

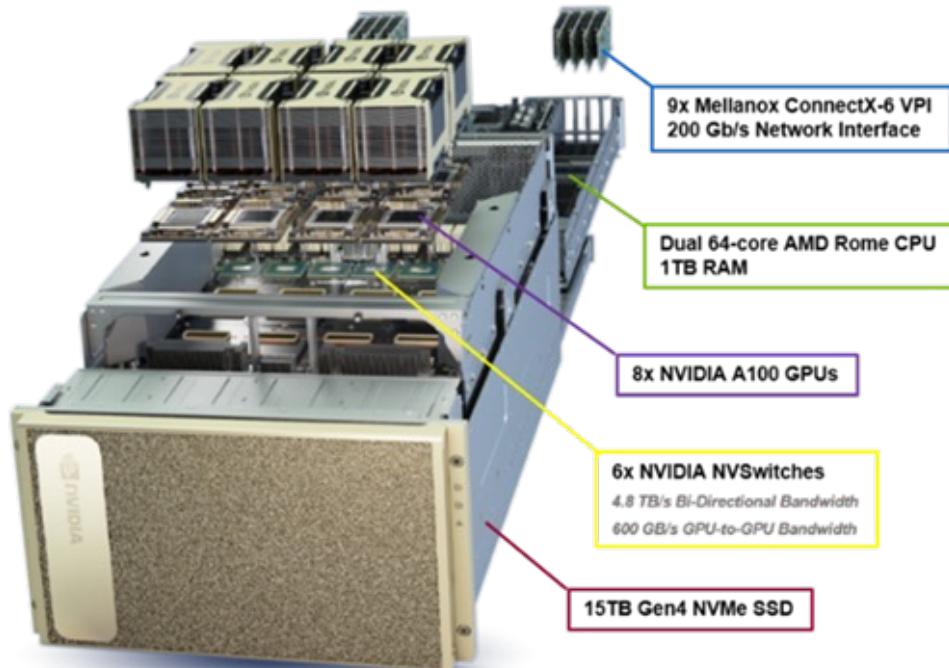
1. 방대한 전문지식을 가진 전문가들이 많이 필요함

- 딥러닝 기본 이론
- Tensorflow, Pytorch framework를 능숙하게 다룰 줄 알아야함
- 딥러닝 모델을 최적화 하고 배포하기 위한 엔지니어링 지식
- 새로운 모델을 연구하고 개발하기 위한 리서치 능력



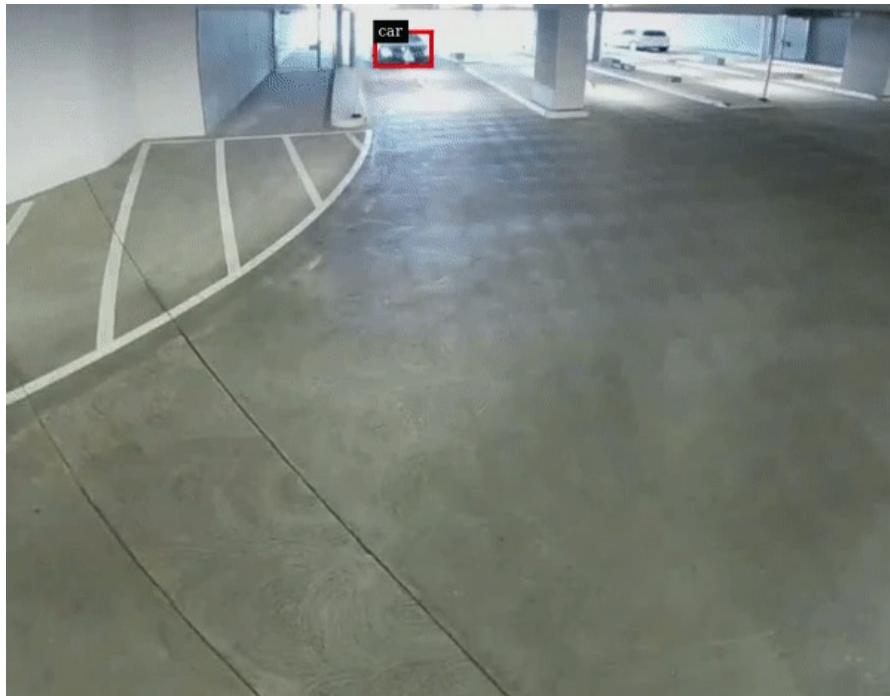
딥러닝이 어려운 이유

2. 하드웨어 (e.g., GPUs)와 하드웨어를 움직이는 소프트웨어를 능숙하게 다룰 줄 알아야 함



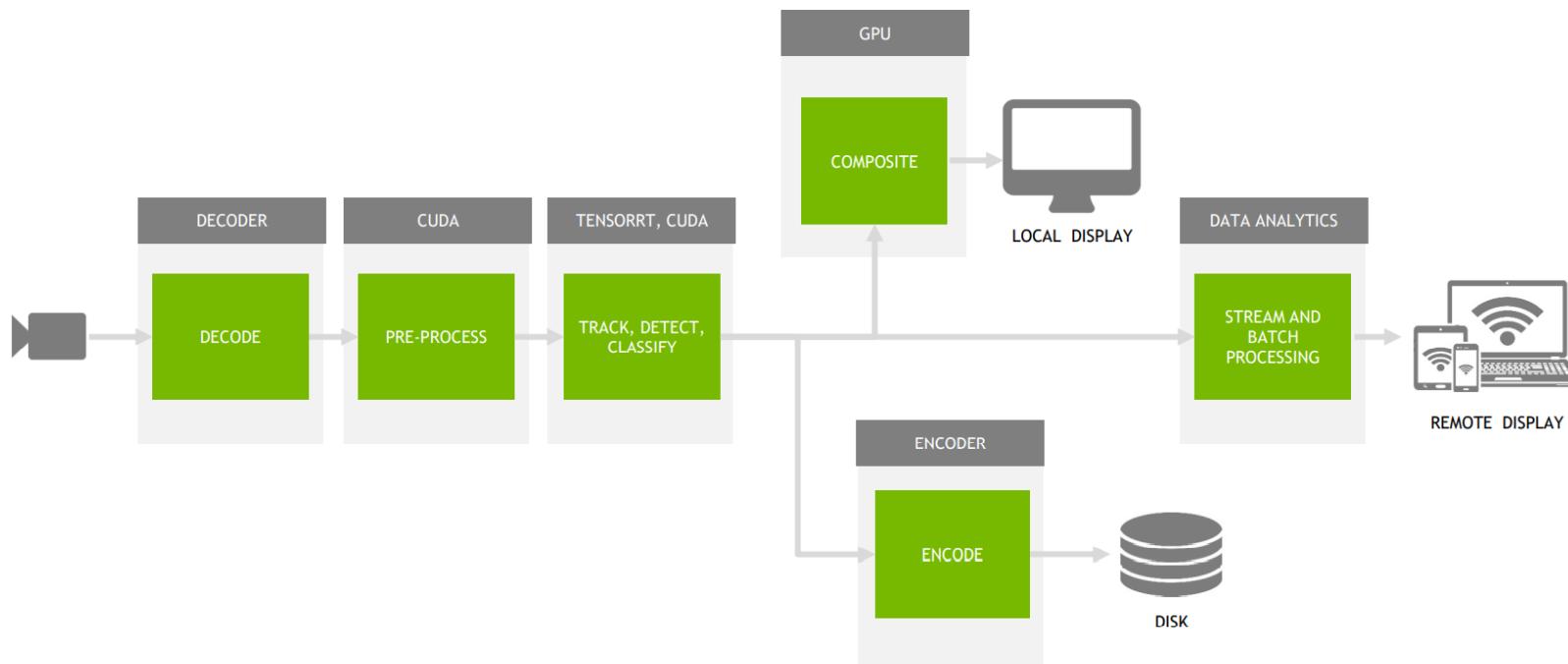
딥러닝이 어려운 이유

3. Real-time application (e.g., 무인차)들은 실시간으로 데이터를 처리하는데 어려움이 있음



딥러닝이 어려운 이유

4. 보통 딥러닝 모델 개발시 Small edge device (e.g., 휴대폰)부터 Public cloud까지 다양한 플랫폼에서 사용 가능하도록 고려해야 한다



딥러닝이 어려운 이유

- 보통 딥러닝 모델 개발시 Small edge device (e.g., 휴대폰)부터 Public cloud까지 다양한 플랫폼에서 사용 가능하도록 고려해야한다

아이디어와 데이터만 있으면 딥러닝 모델을
쉽게 만들고 배포할 수 있는 방법이 없을까?

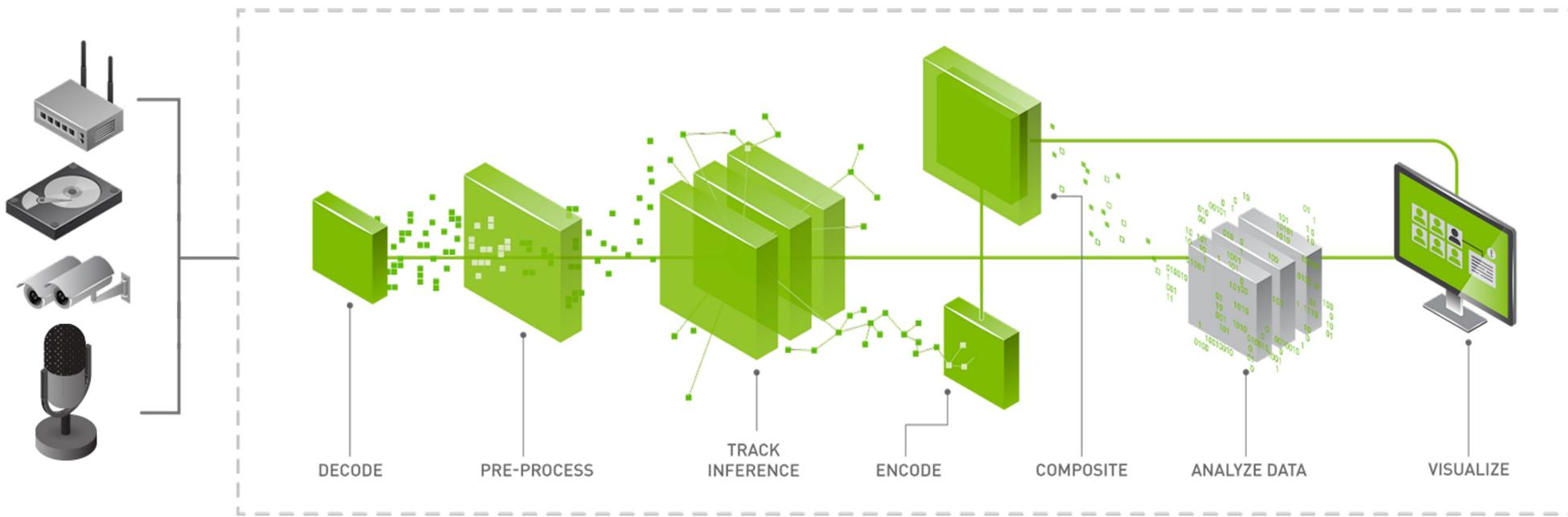


목차 – Introduction

- 딥러닝이 어려운 이유
- **DeepStream**

DeepStream

- DeepStream이란? 실시간 비디오 처리를 위한 모델 파이프라인을 쉽게 만들 수 있는 framework.



DeepStream

- 별도의 학습과정 없이 Cloud에서 모델을 불러서 쓰기만 하면 된다.

The screenshot shows the NVIDIA NGC Catalog interface. On the left is a sidebar with a navigation menu:

- CATALOG
- Explore Catalog
- Collections
- Containers
- Helm Charts
- Models** (highlighted)
- Resources

The main area is titled "Models". It features a search bar and a sorting dropdown set to "Sort: Last Modified". Below is a grid of model cards:

Model Type	Model Name	Description	Action Buttons
NEMO	STT Ca Conformer-Transducer Large	Conformer-Transducer-Large model for Catalan automatic speech recognition, Trained on MCV-9.0 dataset.	View Labels Download
NEMO	STT Ca Conformer-CTC Large	Conformer-CTC-Large model for Catalan automatic speech recognition, trained on MCV 9.0 dataset.	View Labels Download
NEMO	STT Rw Conformer-CTC Large	Conformer-CTC-Large model for Kinyarwanda Automatic Speech Recognition, trained on Mozilla Common Voice 9.0 Kinyarwanda dataset.	View Labels Download
NEMO	STT Rw Conformer-Transducer Large	Conformer-Transducer-Large model for Kinyarwanda Automatic Speech Recognition, trained on Mozilla Common Voice 9.0 Kinyarwanda dataset.	View Labels Download
TAO / RIVA	RIVA Punctuation and Capitalization for F...	For each word in the input text, the model: 1) predicts a punctuation mark that should	View Labels Download
TAO / RIVA	RIVA Punctuation and Capitalization for H...	For each word in the input text, the model predicts a punctuation mark that should	View Labels Download
NEMO	Russian Tagger-based Inverse Text Norm...	Russian single-pass tagger-based model for inverse text normalization based on BERT	View Labels Download
TAO / RIVA	Riva ASR Hindi LM	Base Hindi 3-gram LM	View Labels Download

목차 – DeepStream

- API Key Setup
- DeepStream이 좋은 이유
- DeepStream Pipeline 구축 (Part 1)
- DeepStream Pipeline 구축 (Part 2)
- DeepStream 활용을 위한 참고 사이트

목차 – DeepStream

- **API Key Setup**
- DeepStream이 좋은 이유
- DeepStream Pipeline 구축 (Part 1)
- DeepStream Pipeline 구축 (Part 2)
- DeepStream 활용을 위한 참고 사이트

API Key Setup

- DeepStream을 사용하기 위해 Nvidia GPU Cloud (NGC)에서 API Key를 받아옴
- <https://catalog.ngc.nvidia.com/>

The screenshot shows the NVIDIA NGC setup interface. On the left, there's a sidebar with a catalog icon and a dropdown menu. The main content area has a header "Setup" and a sub-header "Setup". It features two sections: "Generate API Key" (with a key icon) and "CLI" (with a terminal icon). The "Generate API Key" section contains the text: "Generate your own API key in order to use the NGC service through the Docker client or through NGC CLI." Below this is a green "Get API Key" button. A red box highlights this entire section. To the right of the "Generate API Key" section is another green "Documentation" button. The "CLI" section includes the text: "The NGC command line interface (NGC CLI) can run deep learning jobs on NVIDIA Docker containers." Below it are two green buttons: "Documentation" and "Downloads". On the far right, there's a "My Account Settings" menu with options: "Setup" (highlighted with a red box), "Terms of Use", "Privacy Policy", and "Sign Out". The top right corner shows a user profile with the name "jeyoon" and a dropdown menu.

API Key Setup

- DeepStream을 사용하기 위해 Nvidia GPU Cloud (NGC)에서 API Key를 받아옴

The screenshot shows the 'API Key' setup page in the NVIDIA NGC interface. At the top, there's a navigation bar with 'Catalog' and a user dropdown. The main content area has a title 'API Key' and a 'Generate API Key' button. Below this, there are sections for 'API Information' and 'Usage'. The 'API Information' section explains that it generates an API key for Docker client access. The 'Usage' section provides a command-line example for the NGC CLI:

```
$ nvc config set
```

For the Docker™ CLI, it shows a command to log in:

```
$ docker login nvcr.io
```

Username: \$oauthToken
Password: dIVqOW50amVndW5ndmZqcTAyNHNOMWtMzE6YWMMyMmYzNDatNjAwYi0ONzUxIWIxNjYtYmPm2jE2MjJiOWIY

A note says: 'API Key generated successfully. This is the only time your API Key will be displayed. Keep your API Key secret.' Below this, a large green box contains the generated API key.

At the bottom left, there are 'Collapse' and 'NGC Version: v2.96.1' buttons.

API Key Setup

- DeepStream을 사용하기 위해 Nvidia GPU Cloud (NGC)에서 API Key를 받아옴

```
%%writefile config
;WARNING - This is a machine generated file. Do not edit manually.
;WARNING - To update local config settings, see "ngc config set -h"

[CURRENT]
apikey=<<<MY_NGC_API_KEY>>>
format_type=json
org=nvidia
```

```
# DO NOT CHANGE THIS CELL
!mkdir -p ~/.ngc & mv config ~/.ngc/
```

목차 – 2부: DeepStream

- API Key Setup
- **DeepStream0| 좋은 이유**
- DeepStream Pipeline 구축 (Part 1)
- DeepStream Pipeline 구축 (Part 2)
- DeepStream 활용을 위한 참고 사이트

DeepStream이 좋은 이유

- 세상에는 정말 다양한 곳에서 Video AI가 필요하다.



Access Control



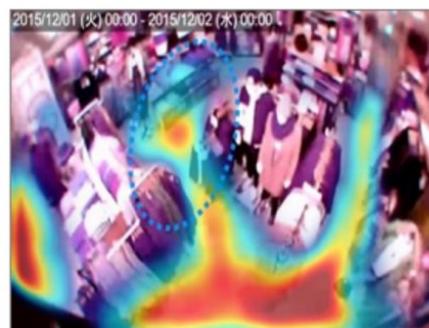
Managing operations



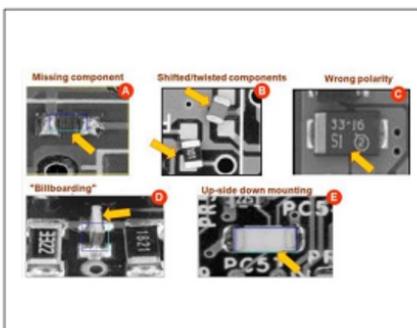
Parking Management



Traffic Engineering



Retail Analytics



Optical Inspection



Managing Logistics



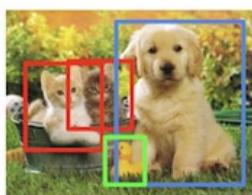
Content Filtering

DeepStream이 좋은 이유

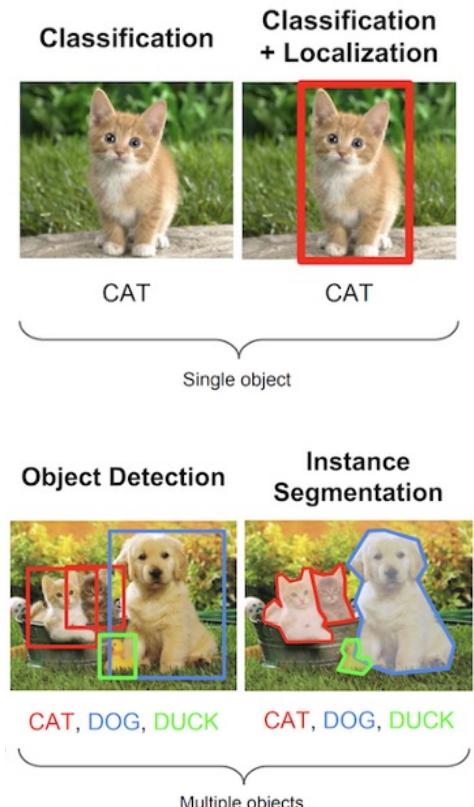
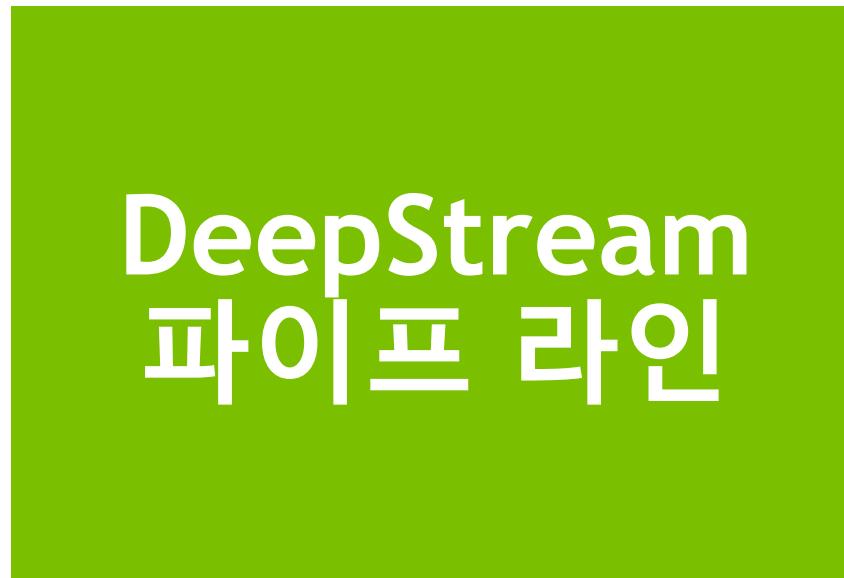
- 다양한 Real-Time Video AI application들을 쉽게 만들 수 있다.



CAT



CAT, DOG, DUCK

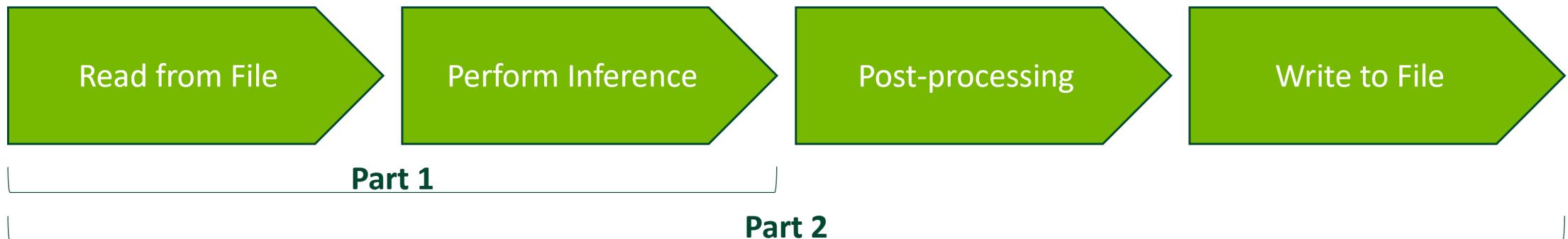


목차 – 2부: DeepStream

- API Key Setup
- DeepStream이 좋은 이유
- **DeepStream Pipeline 구축 (Part 1)**
- DeepStream Pipeline 구축 (Part 2)
- DeepStream 활용을 위한 참고 사이트

DeepStream Pipeline 구축

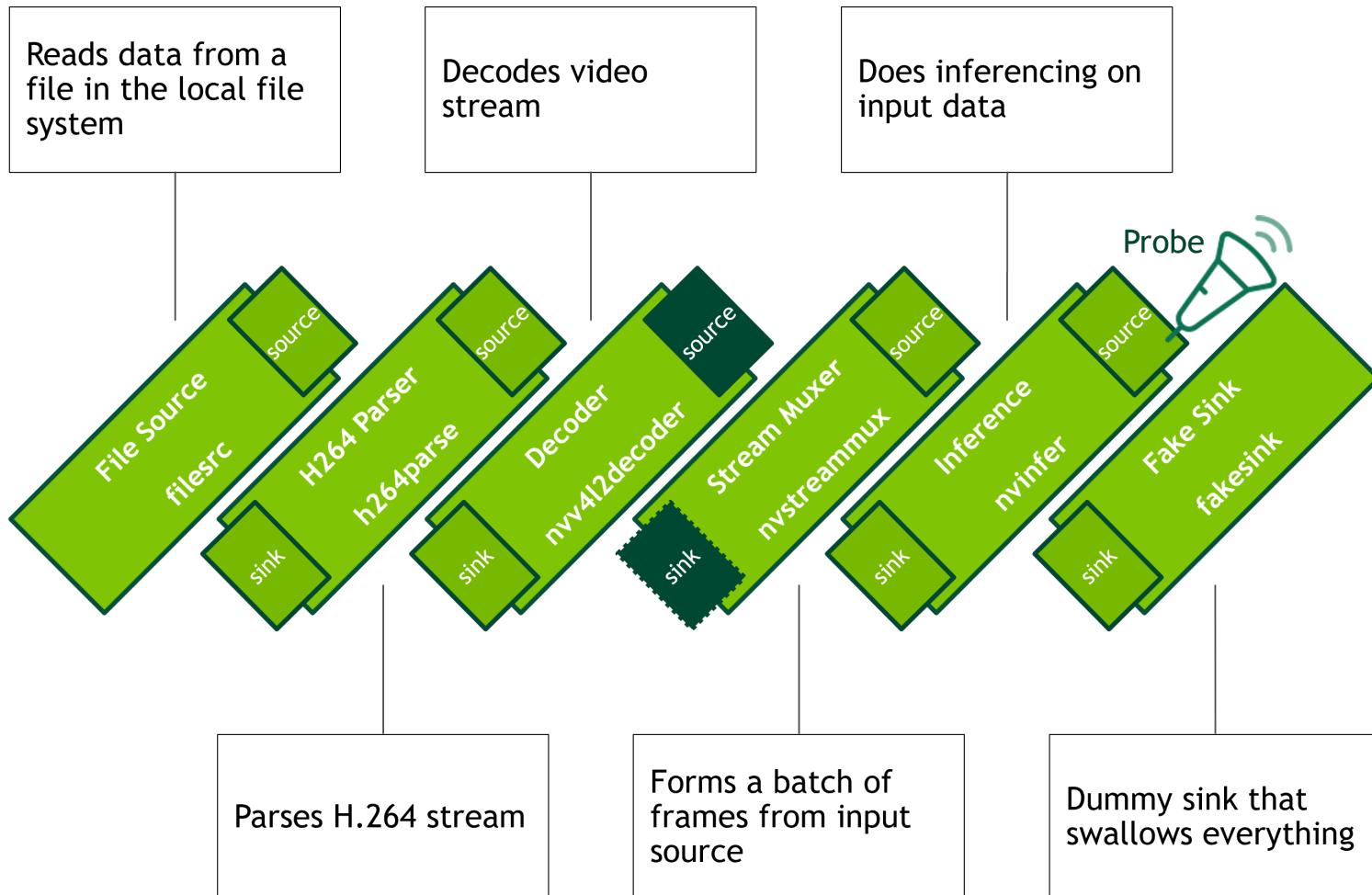
- Part 1: 데이터를 읽고 파이프라인에 넣고 출력 결과 확인
- Part 2: 파이프라인 출력결과에 Bounding box, text label등을 추가, 저장





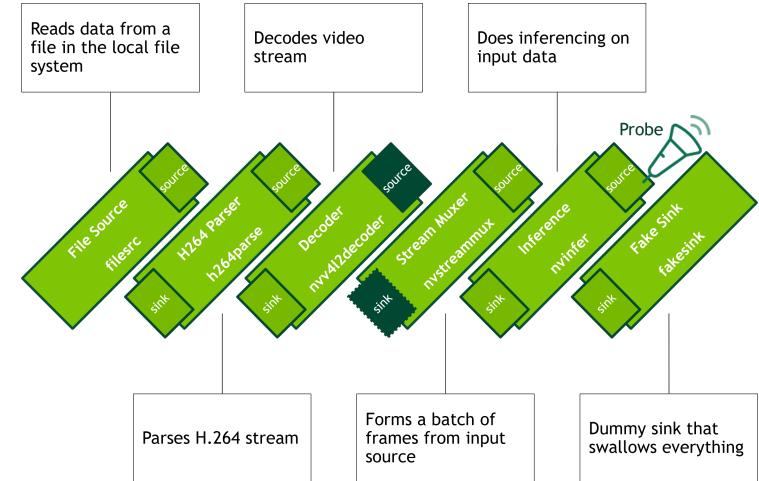
Part1 Code Review

Part 1 요약



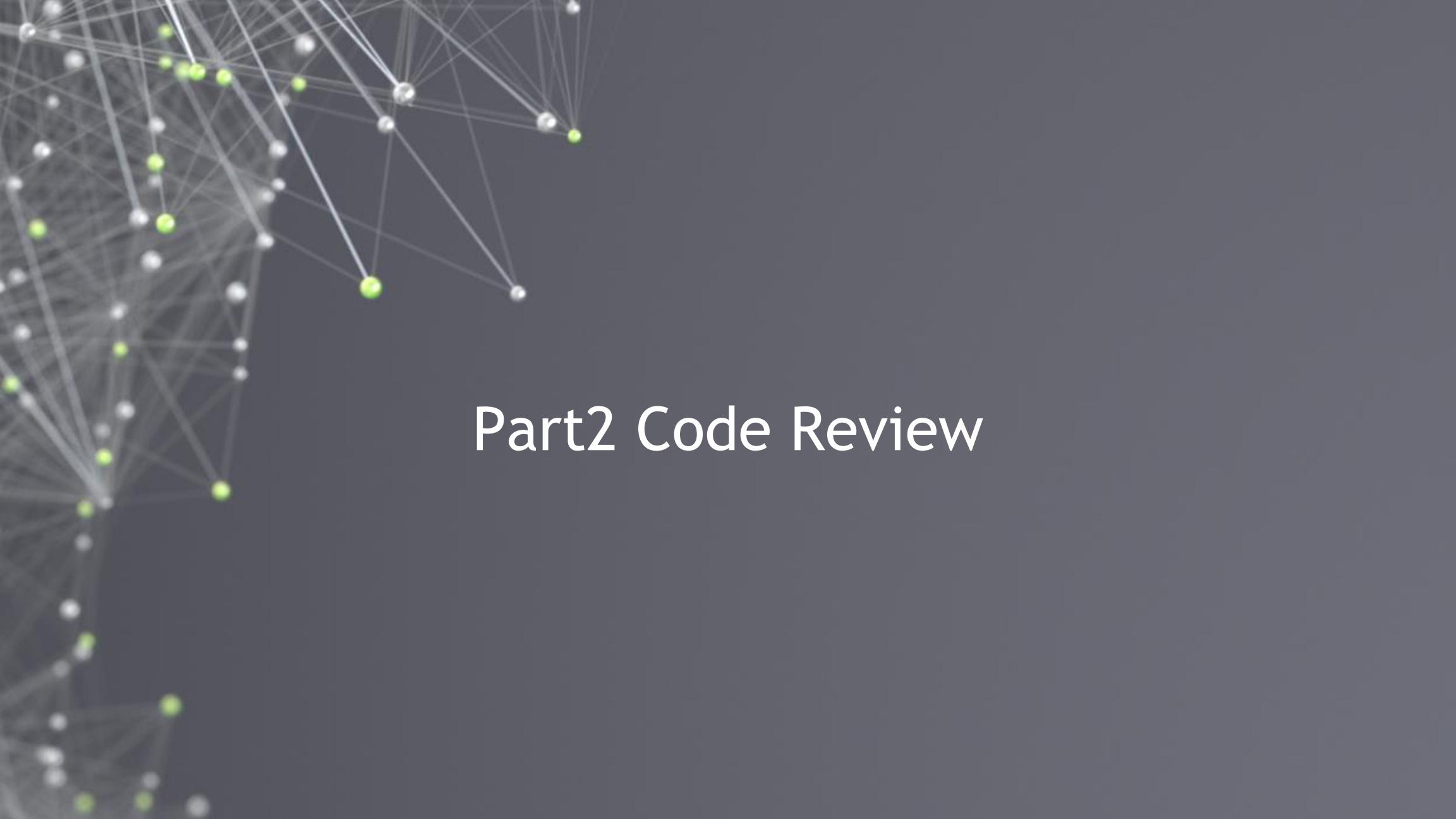
Part 1 요약

- NGC 설치
- TrafficCamNet Object Detection Model 불러오기
- GStreamer and Pipeline 초기화 (`Gst.init(None)`)
- Pipeline 요소 생성 (`pipeline.add()`)
- Configuration 설정 (`pgie.set_property()`)
- 파이프라인 내 요소들 링킹 (`link()`)
- Probe를 이용하여 metadata 확인 (e.g., 프레임 당 예측된 차량 수)
 - Data가 pad (i.e., source or sink pad)를 지나갈 때 probe로 확인
 - `gst_buffer_get_nvds_batch_meta()` / `pyds.NvDsFrameMeta.cast()`
- 파이프라인 시작
 - Bus: 메세지 모니터 하는 곳 / `get_bus()` / `add_signal_watch()`
 - loop = `Glib.MainLoop()`
- Inference
 - `loop.run()`



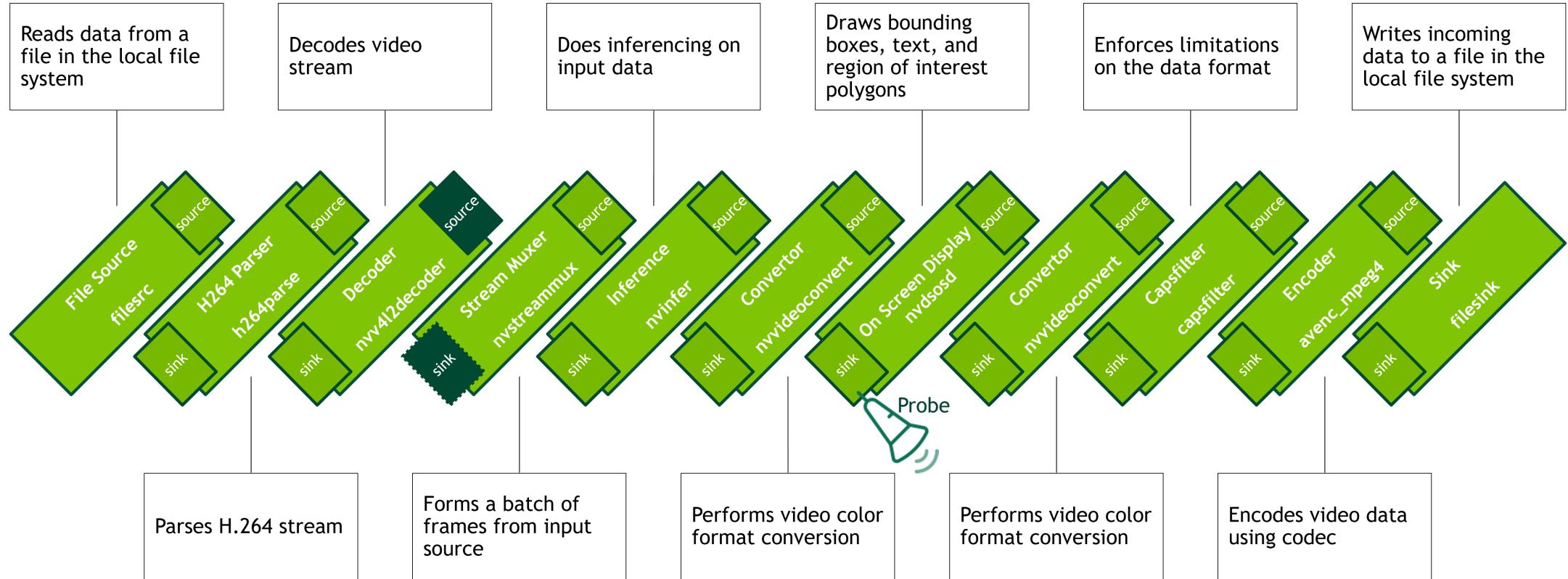
목차 – 2부: DeepStream

- API Key Setup
- DeepStream0| 좋은 이유
- DeepStream Pipeline 구축 (Part 1)
- **DeepStream Pipeline 구축 (Part 2)**
- DeepStream 활용을 위한 참고 사이트



Part2 Code Review

Part 2 요약



Part 2 요약

- Part1에서 했던거
- fakesink 제거 (`pipeline.get_by_name('fakesink')`)
- nvdsosd: bounding box랑 text label 그려주는 모듈
- nvvideoconvert: video frame format 설정 (e.g., RGBA to I420 (YUV))
- capsfilter: 출력 데이터 포맷 설정 (e.g., I420)
- avenc_mpeg4: MPEG4 codec을 이용하여 I420 formatted frame으로 encode
- filesink: 데이터 저장

목차 – 2부: DeepStream

- API Key Setup
- DeepStream이 좋은 이유
- DeepStream Pipeline 구축 (Part 1)
- DeepStream Pipeline 구축 (Part 2)
- **DeepStream 활용을 위한 참고 사이트**

DeepStream 활용을 위한 참고 사이트

- DeepStream github:

(https://github.com/NVIDIA-AI-IOT/deepstream_python_apps)

- DeepStream SDK:

(<https://developer.nvidia.com/deepstream-sdk>)

- GST-python github:

(<https://github.com/GStreamer/gst-python>)

- DeepStream Plugin Guide:

(<https://docs.nvidia.com/metropolis/deepstream/dev-guide/index.html#plugins-development-guide>)

- GStreamer Plugin Guide:

(https://gstreamer.freedesktop.org/documentation/plugins_doc.html?gi-language=c)

DeepStream 활용을 위한 참고 사이트

- e.g., DeepStream github:

The screenshot shows the GitHub repository page for 'NVIDIA-AI-IOT / deepstream_python_apps'. The repository has 57 stars and 27 commits. It contains several branches and tags. The 'Code' tab is selected, showing a list of files and their last commit details. Key files include 'nv-rpalwal Update to 1.1.3 release', '3rdparty', 'apps', 'bindings', 'docs', 'notebooks', 'tests', '.gitmodules', '.python-app-pipeline.png', '.test3-app.png', 'FAQ.md', 'HOWTO.md', 'LICENSE', 'README.md', and 'THIRD_PARTY_LICENSE'. The 'About' section describes the DeepStream SDK Python bindings and sample applications. It includes links to 'Readme', 'View license', '833 stars', '57 watching', and '319 forks'. The 'Releases' section lists one release: 'DeepStream_Python_Apps_Bind...' (Latest). The 'Packages' section indicates 'No packages published'. The 'Contributors' section lists three contributors: nv-zhliu, nv-rpalwal, and aparnachhajed.

Download the Deepstream Python bindings file from here:

https://developer.nvidia.com/deepstream_python_v0.5 Untar the file in the following folder: tar xf ds_pybind_0.5.tbz2 -C <DeepStream 4.0.1 ROOT>/sources

```
!tar xf ds_pybind_0.5.tbz2 -C /home/nvidia/Downloads/deepstream/sources
```

Package Contents

The DeepStream Python package includes:

1. Python bindings for DeepStream Metadata libraries These bindings are provided as a compiled module, available for x86_64 and Jetson platforms. Find them at:
bindings
 - | - pyds.so
 - | - jetson
 - | - pyds.so
2. DeepStream test apps in Python Four test apps are available: deepstream-test1/deepstream_test_1.py deepstream-test2/deepstream_test_2.py deepstream-test3/deepstream_test_3.py deepstream-test3/deepstream_test_4.py

Run code:

Note - Make sure the folder path of where pyds.so file is located is in PATH.

Importing necessary libraries

```
import sys
sys.path.append('..')
sys.path.append('/home/nvidia/Downloads/deepstream/sources/python/bindings/x86_64')
sys.path.append('/usr/lib/python3/dist-packages')
from common.isearch64 import isearch64
import gi
gi.require_version('Gst', '1.0')
from gi.repository import GObject, Gst
from common.isearch64 import isearch64
from common.bus_call import bus_call
import pyds
```

Declaring class label ids

감사합니다



<https://jeiyoong.github.io/>



DEEP
LEARNING
INSTITUTE