AN INTRODUCTION TO DEEPSTREAM SDK

Kaustubh Purandare
March 2018
AGENDA

• Introduction to DeepStream SDK
• DeepStream SDK Basic Building Blocks
• Setup & Installation
• Application Examples
• Performance Analysis
• 3rd Party Integration
• Q&A
WHAT IS DEEPSTREAM SDK

1) NVIDIA DeepStream simplifies the development of scalable intelligent video analytics (IVA) applications

2) Developers can now use this to quickly build new applications to transform video into valuable insight.

3) Applications for the DeepStream SDK include image classification, scene understanding, video categorization, content filtering etc..
DEEPSTREAM SDK FOR INTELLIGENT VIDEO ANALYTICS

Access control
Public Transit
Parking Management
Traffic Engineering
Retail Analytics
Securing Critical Infrastructure
In-Vehicle Analytics
Law Enforcement
DEEPSTREAM SDK

User Applications

Modular, Scalable, Architecture
3rd party application plugins
Sample DNNs & plug-ins
Solution adaptation guide
Development Tools

DeepStream SDK

Deep Learning
TensorRT
cuDNN

Multimedia
Multi-stream Encode & Decode

Imaging
Image capture & processing

Linux, CUDA

NVIDIA Platforms

* Early release and limited availability.
DEEPSTREAM SDK BUILDING BLOCKS
SETUP & INSTALLATION

A] Jetson

3) Ubuntu 16.04 64-bit operating system (host)
4) Jetson TX1 / TX2 Development Platform

B] Running the Sample Application

```shell
nvgstiva-app -c <HOME_dir>/configs/<Config.txt> \
-i /home/nvidia/<path_to_input_stream>
```
SAMPLE APPLICATION

Situational awareness -
• Identify cars, pedestrians, and two-wheelers
• Classify Make (Mercedes, BMW, Audi, …)
• Classify Type (SUV, Sedan, Truck, …)
• Identify Color (Black, Blue, …)
• Read license plate
APPLICATION FLOW

Parse application configuration file

Create graphs
(add/don’t add elements, set properties based on configuration)

Set application graphs to playing

Metadata generated Callback
(Called per frame with metadata generated by the graphs – object coordinates, secondary labels, unique tracking id, etc)

Overlay Graphics Callback
(Overlay custom text / graphics per frame)

Loop

Destroy graphs
(Free up resource)
[application]
enable-perf-measurement=1
tracker-id=1
roi-marking=0
debg-mode=0
:
[source0]
enable=1
#Type - 1=CameraCSI 2=CameraV4L2 3=URI
uri=file:///home/ubuntu/00023.MTS
:
[sink0]
enable=1
#Type - 1=FakeSink 2=OverlaySink 3=EglSink 4=XvImageSink 5=File
:
[osd]
enable=1
osd-mode=2
border-width=2
text-size=15
:
[primary-gie]
enable=1
model-file=file:///home/ubuntu/Model/resnet/ResNet_*.txt
APPLICATION FLOW

1. Parse application configuration file
2. Create graphs (add/don’t add elements, set properties based on configuration)
3. Set application graphs to playing
4. Metadata generated Callback (Called per frame with metadata generated by the graphs – object coordinates, secondary labels, unique tracking id, etc)
5. Overlay Graphics Callback (Overlay custom text / graphics per frame)
6. Destroy graphs (Free up resource)

Loop
App adds elements if required based on the configuration
Individual elements/stages are configured
Links the elements
Gstreamer framework performs caps negotiation, buffer allocations/deallocations transparent to the application
Gstreamer pipeline and elements take care of zero buffer copies, buffer management
APPLICATION FLOW

Parse application configuration file

Create graphs
(add/don’t add elements, set properties based on configuration)

Set application graphs to playing

Metadata generated Callback
(Called per frame with metadata generated by the graphs – object coordinates, secondary labels, unique tracking id, etc)

Graphics Overlay
(Overlay custom text / graphics per frame)

Destroy graphs
(Free up resource)

Loop
CREATING SIMPLE GRAPHSS

1) Gstreamer provides powerful tool like `gst-launch` to create trial / experimental graphs as per use cases.

2) File stream with Primary object detection and OnScreen Display
   - `gst-launch-1.0 uridecodebin uri=file:///home/nvidia/video.mp4 ! nvinfer <primary-infer-properties> ! queue ! nvosd <osd-properties> ! nveglglessink`

3) RTSP stream with primary object detection + tracking + secondary classification labels + OnScreen Display
PERFORMANCE ANALYSIS
APPLICATION SOFTWARE STACK

Applications

Sample DL Networks  ISV/ 3rd party solutions  Custom DL Networks

Plugin based compute Graph • Gstreamer Framework

DeepStream

Deep Learning  Multimedia  Imaging

TensorRT  cuDNN  Multistream Encode & Decode  Image capture & processing

Linux, CUDA  NVIDIA Platforms
START DEVELOPING WITH DEEPSTREAM

DeepStream Early Access program . Explore Metropolis . Intelligent Video Analytics Forums
THANK YOU