# Lab 2: Learn to Use DeepStream Application and YOLO

## Objectives

* Configure and Run DeepStream Application for object detection via video file input

## Deliverables

**Compulsory (7%)**

* DeepStream application configuration files from Part 1: Step 5, saved to your home folder ~/deepstream-detectnet/configs
* Screenshot from Part 1: Step 4 before and after running DeepStream Application

**Extra (3%)**

* Output RTSP stream from DeepStream Application.

## Prerequisites

* Use `tmux` terminal multiplexer. Create a window for each part.
* Install `jetson-stats` to monitor jet son hardware status

sudo -H pip install -U jetson-stats

1. Install a media player in your OS that can stream RTSP stream, e.g. VLC Player.

**Part 1: Configure and Run DeepStream Application**

**Objective: Understanding GStreamer pipeline**

**Step 1: Prepping DeepStream project folder**

1. **Create shared volume from host to container. This will be the project root folder, any content modified here from localhost or container will be synchronized.**

mkdir -p ~/deepstream-detectnet/data

1. **Copy video file**

cp /opt/videos/traffic.mp4 ~/deepstream-detectnet/data/

**Step 2: Run DeepStream SDK container**

1. Run DeepStream SDK container using nvidia-runtime that will automatically pass GPU driver inside. The container will contains all the DeepStream SDK inside without having to install files in the host OS.

docker run --gpus all -it \

-v $HOME/deepstream-detectnet:/opt/nvidia/deepstream/deepstream/sources/project \

-w /opt/nvidia/deepstream/deepstream/sources/project \

-p 855**4**:8554 \

--name deepstream-detectnet**-s1155160788** \

nvcr.io/nvidia/deepstream-l4t:6.0-samples

Note:

-v: Bind local host folder to container folder

-w: Set working directory

-p: Expose port from localhost to container port

--name: Name of the container

1. **Get the currently running container ID and name that runs DeepStream SDK.**

docker ps

1. **Login to the container**

docker exec -it <CONTAINER\_NAME>

**Step 3: Create DeepStream app configuration file**

1. **Create a `configs` folder under the project folder to store DeepStream application configuration file `/opt/nvidia/deepstream/deepstream/sources/project`**

mkdir configs

1. **Create a DeepStream app configuration file called `deepstream\_basic.txt` that has the following GStreamer pipeline:**

**[source0] -> [streammux] -> [primary-gie] -> [tiled-display] -> [osd] -> [sink0]**

[application]

enable-perf-measurement=1

perf-measurement-interval-sec=1

[source0]

enable=1

#Type - 1=CameraV4L2 2=URI 3=MultiURI 4=RTSP

type=3

uri=file://../data/traffic.mp4

num-sources=1

gpu-id=0

cudadec-memtype=0

[streammux]

## Boolean property to inform muxer that sources are live

## Set to 1 to process @ source framerate, 0 for asap

live-source=1

batch-size=1

##time out in usec, to wait after the first buffer is available

##to push the batch even if the complete batch is not formed

batched-push-timeout=40000

## Set muxer output width and height

width=640

height=480

# config-file property is mandatory for any gie section.

# Other properties are optional and if set will override the properties set in

# the infer config file.

[primary-gie]

enable=1

model-engine-file=../engines/resnet10.caffemodel\_b30\_gpu0.engine

#Required to display the PGIE labels, should be added even when using config-file

#property

batch-size=1

interval=0

#Required by the app for SGIE, when used along with config-file property

gie-unique-id=1

config-file=model\_basic.txt

[tiled-display]

enable=1

rows=1

columns=1

width=1280

height=720

[osd]

enable=1

border-width=2

text-size=15

text-color=1;1;1;1;

text-bg-color=0.3;0.3;0.3;1

font=Serif

show-clock=1

process-mode=2

[sink0]

enable=1

type=3

#1=mp4 2=mkv

container=1

#1=h264 2=h265 3=mpeg4

codec=1

#encoder type 0=Hardware 1=Software

enc-type=0

sync=1

bitrate=4000000

#H264 Profile - 0=Baseline 2=Main 4=High

#H265 Profile - 0=Main 1=Main10

profile=0

output-file=out.mp4

source-id=0

rtsp-port=8554

udp-port=5400

[tests]

file-loop=0

1. **Create the config file for primary GIE (GPU Inference Engine) detector. Save this file to `configs/model\_basic.txt`**

[property]

gpu-id=0

net-scale-factor=0.0039215697906911373

model-file=/opt/nvidia/deepstream/deepstream/samples/models/Primary\_Detector/resnet10.caffemodel

proto-file=/opt/nvidia/deepstream/deepstream/samples/models/Primary\_Detector/resnet10.prototxt

model-engine-file=../engines/resnet10.caffemodel\_b30\_gpu0.engine

labelfile-path=/opt/nvidia/deepstream/deepstream/samples/models/Primary\_Detector/labels.txt

int8-calib-file=/opt/nvidia/deepstream/deepstream/samples/models/Primary\_Detector/cal\_trt.bin

force-implicit-batch-dim=1

batch-size=1

network-mode=1

num-detected-classes=4

interval=0

gie-unique-id=1

output-blob-names=conv2d\_bbox;conv2d\_cov/Sigmoid

#scaling-filter=0

#scaling-compute-hw=0

[class-attrs-all]

pre-cluster-threshold=0.2

eps=0.2

group-threshold=1

**Step 4: Run the DeepStream App Pipeline**

1. **Run the pipeline**

cd /opt/nvidia/deepstream/deepstream/sources/project

deepstream-app -c configs/deepstream\_basic.txt

**Step 5: Monitor the Jetson Hardware engine status**

jtop

**Step 5: Customize the DeepStream App to achieve the following settings. Save each config into individual config name. Copy from the `deepstream\_basic.txt` file.**

1. **Run on CPU [deepstream\_cpu.txt]**
2. **Output to local file [deepstream\_out\_file.txt]**
3. **Loop the input video continuously [deepstream\_loop.txt]**
4. **Change the sink video bitrate to 8Mbps [deepstream\_bitrate\_8000000.txt]**

**[Extra Marks]**

**Output to local RTSP server [deepstream\_out\_rtsp].**

**Take a screenshot with left/right splitted view. Left: Console running deepstream app, Right: RTSP video stream from the video player.**

## References

1. Deepstream Application Configuration Group - [https://docs.nvidia.com/metropolis/deepstream/5.0DP/dev-guide/index.html#page/DeepStream\_Development\_Guide/deepstream\_app\_config.3.2.html#](https://docs.nvidia.com/metropolis/deepstream/5.0DP/dev-guide/index.html#page/DeepStream_Development_Guide/deepstream_app_config.3.2.html)