Real Time Video Analytics on Jetson TX2 using DeepStream SDK

Author: Vivek Maran

Professor: Kaikai Liu

Revision: 1

Contents

[1. INTRODUCTION 1](#_Toc517687997)

[2. REAL TIME VIDEO ANALYTICS 1](#_Toc517687998)

[2.1 Overview of video content analytics 1](#_Toc517687999)

[2.2 Video content analytics workflow 1](#_Toc517688000)

[3 JETSON-TX2 1](#_Toc517688001)

[3.1 Jetson TX2 platform 1](#_Toc517688002)

[3.2 Jetson setup 1](#_Toc517688003)

[4 DEEPSTREAM SDK 2](#_Toc517688004)

[4.1 Overview of DeepStream SDK 2](#_Toc517688005)

[4.2 Video analytics workflow using DeepStream SDK 2](#_Toc517688006)

[4.3 DeepStream API (Source: Link) 2](#_Toc517688007)

[4.4 DeepStream installlation 2](#_Toc517688008)

[4.5 Running DeepStream samples 2](#_Toc517688009)

[4.6 Documents on DeepStream SDK 2](#_Toc517688010)

[5 REFERENCES 2](#_Toc517688011)

## 1. INTRODUCTION

This objective of this document is to provide an high level overview of real time video analytics, and its implementation on embedded devices using Jetson TX2 and DeepStream SDK.

## 2. REAL TIME VIDEO ANALYTICS

### 2.1 Overview of video content analytics

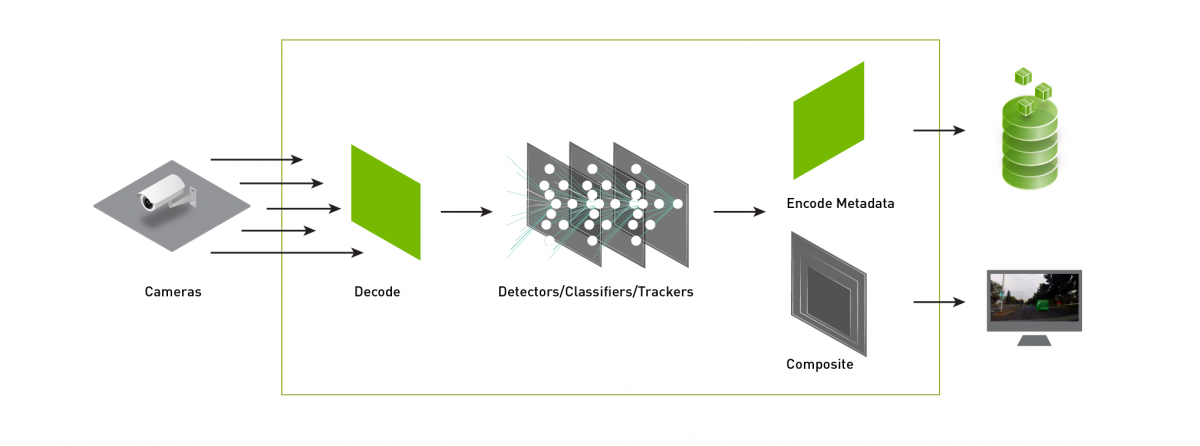
Video content analysis (also video content analytics, VCA) is a type of computer vision that analyzes the behavior of the objects in the scene being captured. Some forms of analysis include:

1. **Object detection:** Detecting instances of semantic objects of a certain class (such as humans, buildings, or cars).
2. **Segmentation:** Process of partitioning a digital image into multiple segments that share similar attributes (like color,) to simplify the representation and making it more useful for the analysis and interpretations.
3. **Object tracking:** Video tracking is the process of locating a [moving](https://en.wikipedia.org/wiki/Motion_(physics)) object (or multiple objects) over time using a camera.

This technical capability is beginning to gain traction in a wide range of domains such as entertainment, health-care, retail, security etc..

### 2.2 Video content analytics workflow

Contemporary video content analytics are built using deep learning technologies. The captured video is ingested to trained deep neural networks for content analysis.



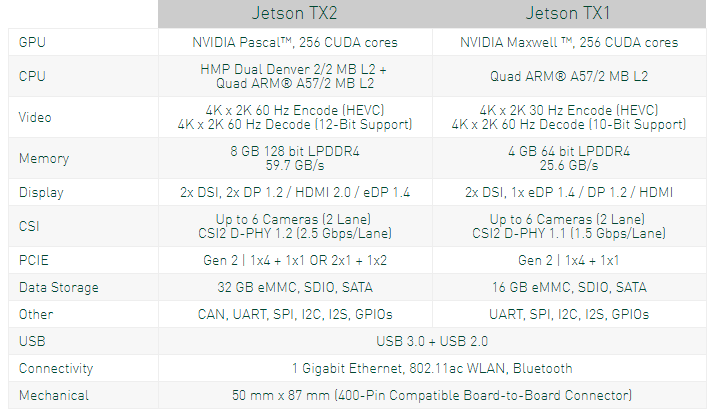
**Figure 1:** *Video content analytics DNN (Img src:* [Link](https://developer.nvidia.com/sites/default/files/akamai/deeplearning/DeepStream/embedded-web-deepstream-image-option04_FIXED.png)*)*

The above figure shows an example workflow of video analytics using deep neural networks. In such a workflow, it is critical to have computationally powerful, low latency, less power consuming hardware to perform multimedia and DNN computations. Jetson TX2 is one such platform which is described in the next section.

## 3 JETSON-TX2

### 3.1 Jetson TX2 platform

NVIDIA's [Jetson TX2](https://developer.nvidia.com/embedded-computing) is an embedded system-on-module (SoM) designed for running deep-learning and computer-vision based applications. Following table shows the specification of Jetson platforms TX-1 and TX-2

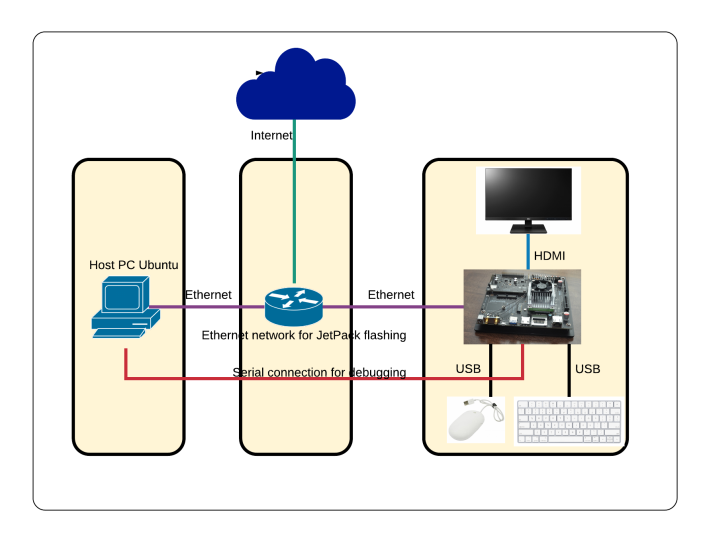


**Figure 2:**  *JetSon specification (Img src:* [*Link*](https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/)*)*

More information can be found on the platform’s wiki page at https://elinux.org/Jetson\_TX2

### 3.2 Jetson setup

***The Jetson TX2 comes pre-flashed with a Linux environment.. On top of which, JetPack can be installed.*** JetPack includes host (Ubuntu Desktop) and target (Jetson) development tools, APIs, and packages (OS images, tools, middleware, samples, and documentation) for developing on the NVIDIA Jetson Embedded platform. ***The components and dependencies required for DeepStream SDK (discussed in the consecutive section) are also installed as a part of JetPack. Following is the development setup for JetPack installation.***

**Figure 3:**  *Jetpack installation setup.*

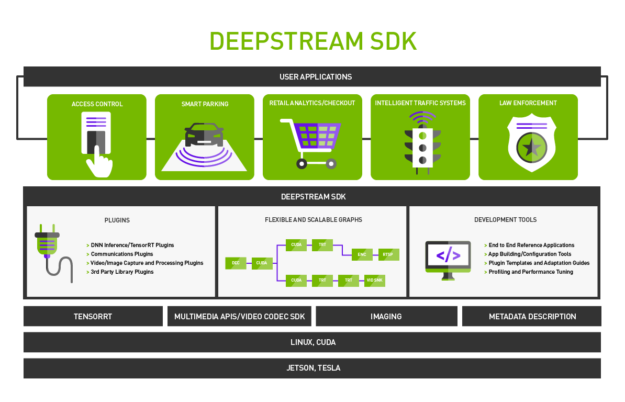
Some useful links for Jetson setup & JetPack installation:

1. [Video demo](https://www.youtube.com/playlist?list=PLXYLzZ3XzIbgtQYph1AYLvhFoBNbNTp8G)
2. [User guide](https://drive.google.com/file/d/1J61ZxGnf8msw0kJ2fUYGpKNaxAWPRIBo/view?usp=sharing)

## 4 DEEPSTREAM SDK

### 4.1 Overview of DeepStream SDK

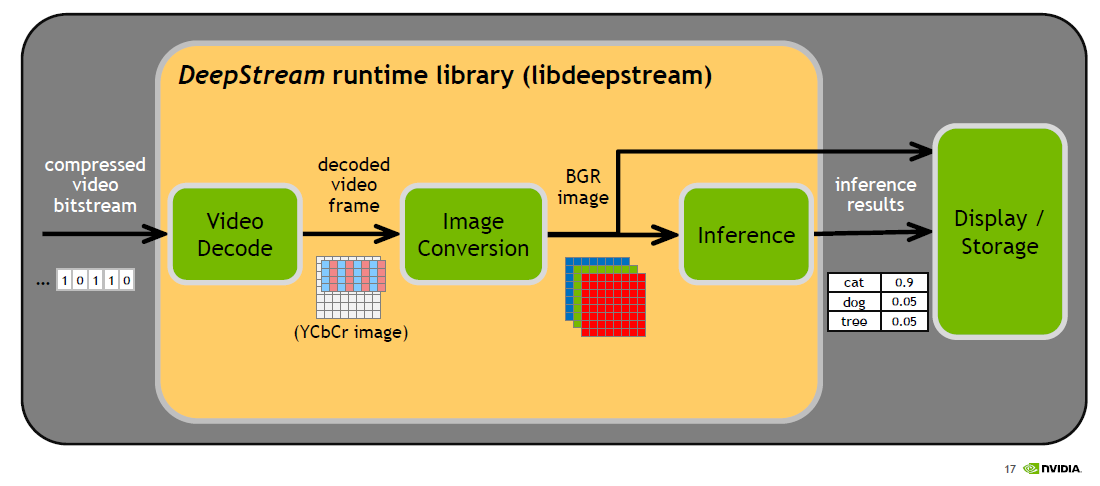
DeepStream SDK on Jetson uses Jetpack, which includes L4T, Multimedia APIs, CUDA, and TensorRT.  The SDK offers a rich collection of plug-ins and libraries, built using the Gstreamer framework to enable developers to build flexible applications for transforming video into valuable insights. Following is the software stack of the DeepStream SDK.

 **Figure 4 :** *DeepStream SDK (Img src:* [*Link*](https://devblogs.nvidia.com/wp-content/uploads/2018/06/deepstream-sdk-infographic-r6-no-copy-625x402.png)*)*

### 4.2 Video analytics workflow using DeepStream SDK

***The DeepStream SDK provides modules that encompass decode, pre-processing and inference*** of input video streams. The decode module accepts video encoded in H.264, H.265, and MPEG-4 among other formats and decodes them to render raw frames in NV12 color format. Video decoding uses the hardware-accelerated [NVIDIA Video Codec SDK](https://developer.nvidia.com/nvidia-video-codec-sdk).

The pre-processing stage converts the color format from NV12 to BGR, and also resizes the frames to the resolution required by the neural network being used. Color space conversion and scaling uses a combination of custom CUDA kernels and [NVIDIA Performance Primitives](https://developer.nvidia.com/npp) (NPP) library functions. The inference module processes the frames in batches using a neural network imported and executed using TensorRT. DeepStream currently supports Caffe and TensorFlow based neural networks.



**Figure 5:***. DeepStream video analytics workflow (Img src: Jeremey Furtek’s Deepstream presentation)*

### 4.3 DeepStream API (Source: [Link](https://devblogs.nvidia.com/deepstream-video-analytics-smart-cities/))

DeepStream provides a C++ API, at the core of which are three entities: modules that serve as individual building blocks of the pipeline, the device worker that encapsulates the whole pipeline itself, and tensors to communicate between modules. Modules represent stages in the pipeline, and can be categorized as either pre-defined (decoder, pre-processing, and inference) or user-defined modules that add custom logic. Following is the pseudo code for pipeline creating using DeepStream APIs.

**Step-1: DeviceWorker creation**

IDeviceWorker \*pDeviceWorker = createDeviceWorker(g\_nChannels, g\_devID);

**Step-2: Adding modules to the DeviceWorker**

// Add decode task, the parameter is the format of codec.

pDeviceWorker->addDecodeTask(cudaVideoCodec\_H264);

// Add post processing task and define color format for inference

pDeviceWorker->addColorSpaceConvertorTask(BGR\_PLANAR);

// Enable inference module and define model details

pDeviceWorker->addInferenceTask(preModule\_infer, deployFile, modelFile, meanFile, inputLayerName, outputLayerNames);

**Step-3: Starting the DeviceWorker**

//finally, start the pipeline!

pDeviceWorker->start();

### 4.4 DeepStream installlation

DeepStream is delivered as a dynamic library libdeepstream.so along with the required headers and samples. The application has to link to this library to use the DeepStream APIs. There seems to be no special installation steps required beyond it.

### 4.5 Running DeepStream samples

Steps to run sample applications provided as a part of DeepStream SDK can be found at section 4.2.4 of the [document](https://drive.google.com/file/d/1UZF7YuBsAOz4xuqf7C46iZJBhp3-Mjo_/view?usp=sharing)

### 4.6 Documents on DeepStream SDK

1. [DeepStream API’s and workflow](https://drive.google.com/file/d/10Ad_0dlVhIHfz-NDxpmlRzyM-gd1B_t3/view?usp=sharing)
2. [DeepStream installation and user guide](https://drive.google.com/file/d/1UZF7YuBsAOz4xuqf7C46iZJBhp3-Mjo_/view?usp=sharing)

## 5 REFERENCES

|  |  |
| --- | --- |
| [ 1  ] | <https://developer.nvidia.com/deepstream-jetson> |
| [  2  ] | <https://devblogs.nvidia.com/accelerate-video-analytics-deepstream-2> |
| [  3  ] | <https://devblogs.nvidia.com/deepstream-video-analytics-smart-cities/> |
| [  4  ] | <https://devblogs.nvidia.com/deploying-deep-learning-nvidia-tensorrt/> |
| [  5  ] | <https://software.intel.com/en-us/articles/a-closer-look-at-object-detection-recognition-and-tracking> |
| [  6  ] | <https://meetshah1995.github.io/projects> |
| [  7  ] | <https://www.cse.unr.edu/~bebis/CRCD/> |
| [  8  ] | <https://en.wikipedia.org/wiki/Video_content_analysis> |
| [  9  ] | <https://en.wikipedia.org/wiki/Object_detection> |
| [  10  ] | <https://www.quora.com/What-is-image-segmentation> |