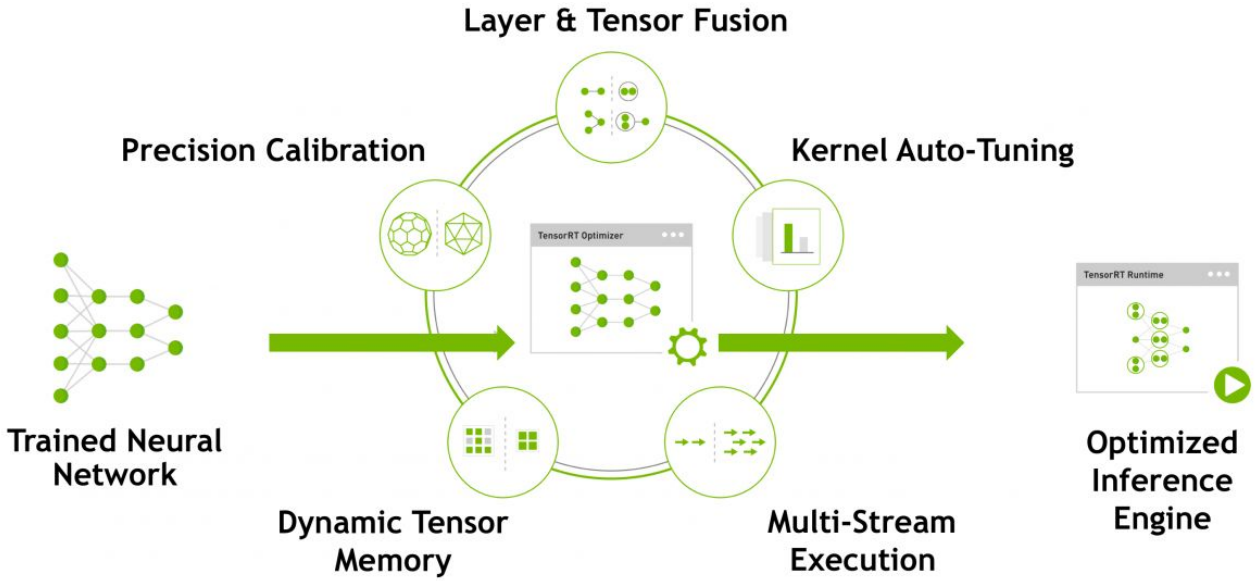
Deployment with TensorRT and DeepStream

Ekapol Chuangsuwanich
Nvidia IVA workshop

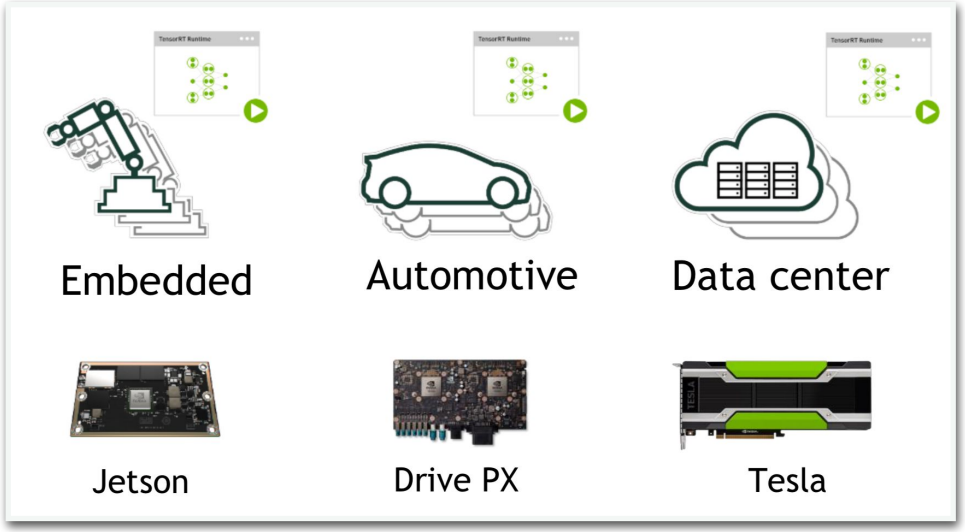
CHULA ENGINEERING
Foundation toward Innovation
COMPUTER

NVIDIA TensorRT 4

High-performance neural network inference optimizer and runtime engine for production deployment



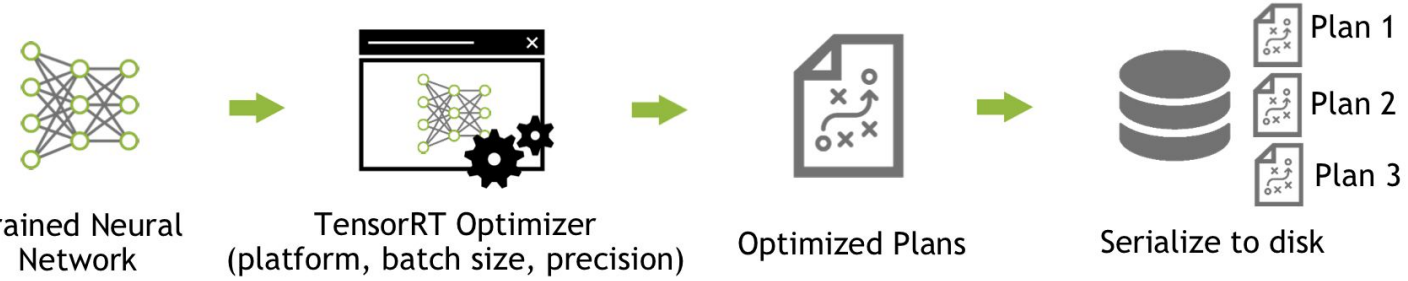
NVIDIA TensorRT 4



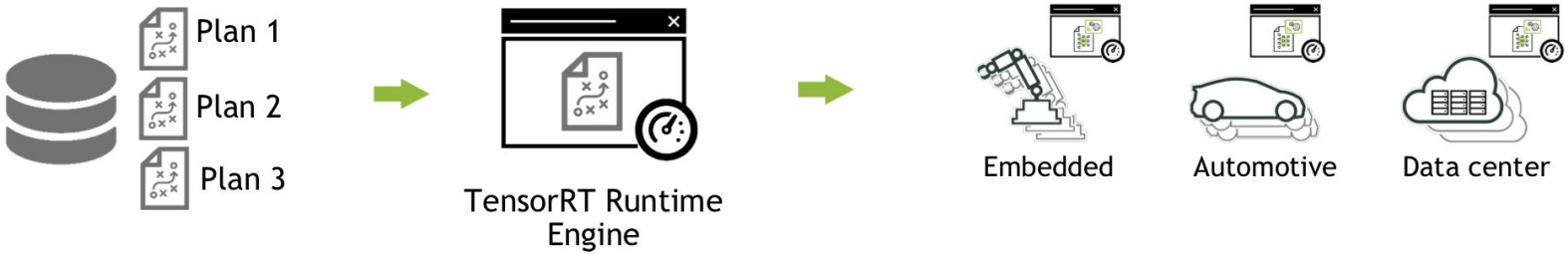
- Maximize inference throughput
- Optimize and deploy TensorFlow and Caffe models
- Deploy faster, more efficient and responsive deep learning applications

TENSORRT DEPLOYMENT WORKFLOW

Step 1: Optimize trained model



Step 2: Deploy optimized plans with runtime

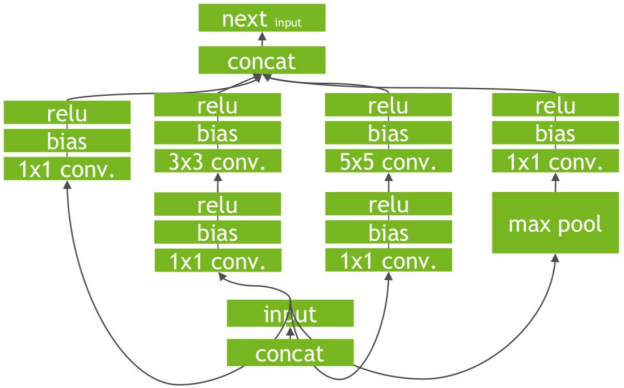


NVIDIA TensorRT
Programmable Inference Accelerator

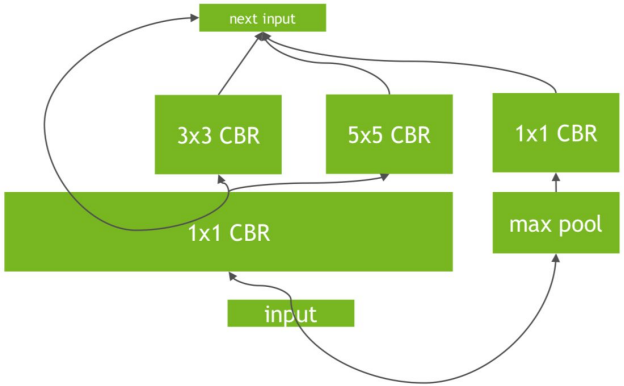
TENSORRT GRAPH OPTIMIZATIONS

TensorRT performs several important transformations and optimizations to the neural network graph

Un-optimized network



TensorRT Optimized Network



Unused output are eliminated to avoid unnecessary computation.

Convolution, Bias, and ReLU layers are fused to form a single layer.

Horizontal layer fusion combine layers that take the same source tensor.

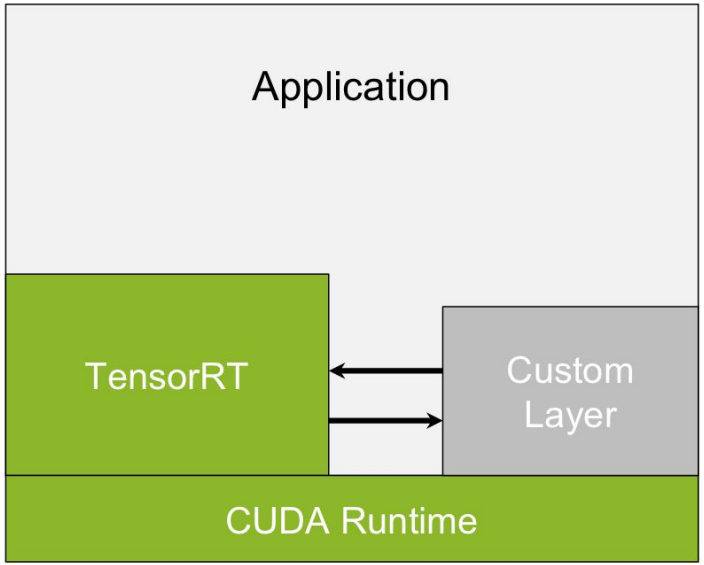
NVIDIA TensorRT
Programmable Inference Accelerator

TENSORRT LAYERS

Built-in support

- Convolution, Deconvolution
- Activation: ReLU, tanh, sigmoid
- Pooling: max and average
- Scaling
- Element wise operations
- LRN
- Fully-connected
- SoftMax
- Gather
- TopK
- Const

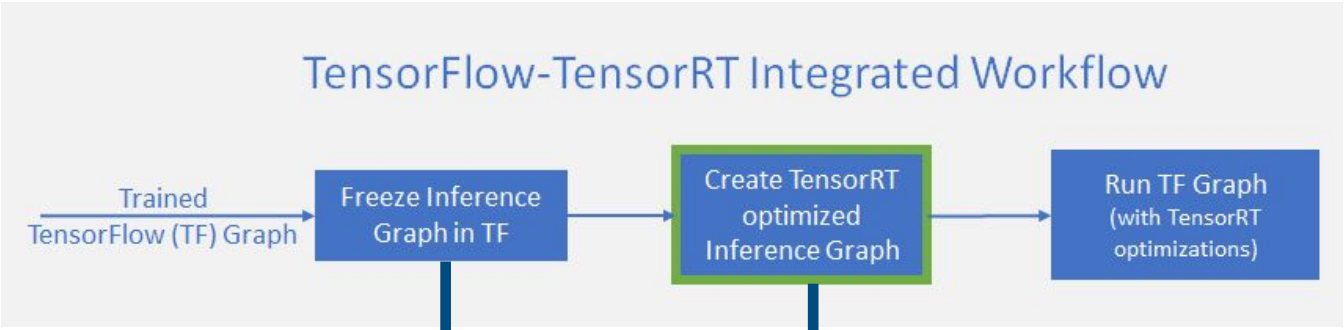
Custom Layer API



NVIDIA TensorRT
Programmable Inference Accelerator

Applying TensorRT optimizations to Tensorflow

TensorRT builds an optimized inference graph
from a frozen TensorFlow graph. **(highlighted in green)**



convert any model format into .pb

optimize graph -> FP32 , FP16

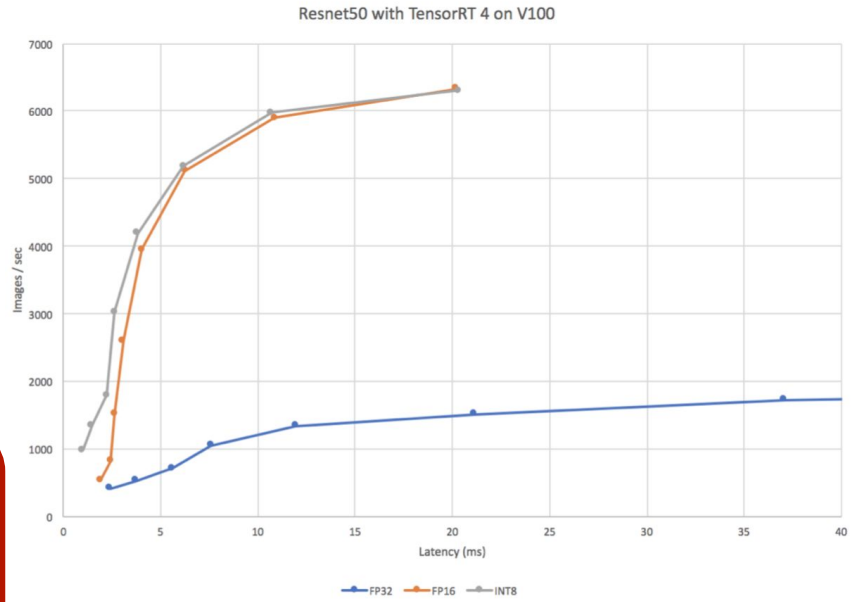
Using New TensorFlow APIs

Speed up TensorFlow inference with TensorRT

8x higher throughput in TensorRT

- sub-graph optimisation TensorRT
- use custom TensorFlow ops

Available in TensorFlow 1.7



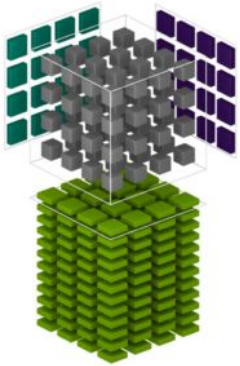
from tensor flow.contrib import tensor as trt

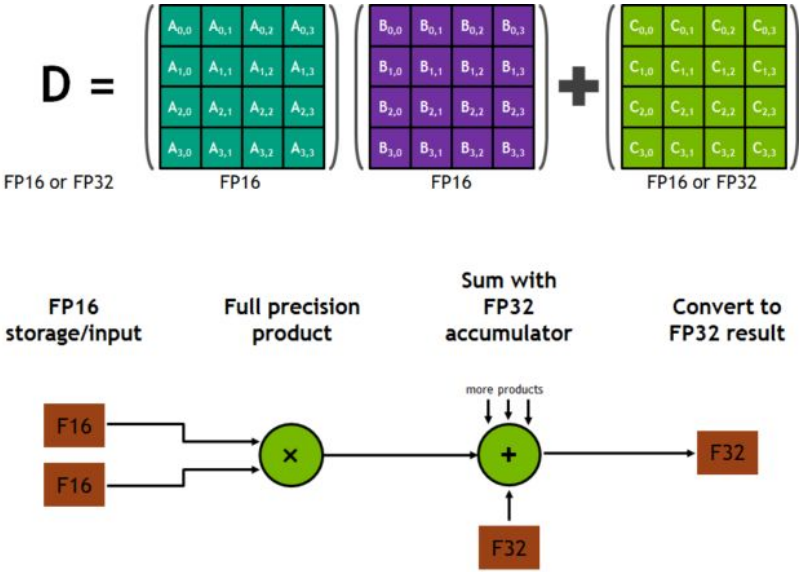
```
trt_graph = trt.create_inference_graph(frozen_graph_def,
                output_node_name,
                max_batch_size=batch_size,
                max_workspace_size_bytes=workspace_size,
                precision_mode=precision)
```

NVIDIA TensorRT
Programmable Inference Accelerator

Using Tensor Cores on Volta GPUs

Tensor Cores provide a 4x4x4 matrix processing array which performs the operation D = A * B + C, where A, B, C and D are 4×4 matrices.





NVIDIA's Volta architecture incorporates hardware matrix math accelerators known as Tensor Cores.

Using New TensorFlow APIs

Speed up TensorFlow inference with TensorRT

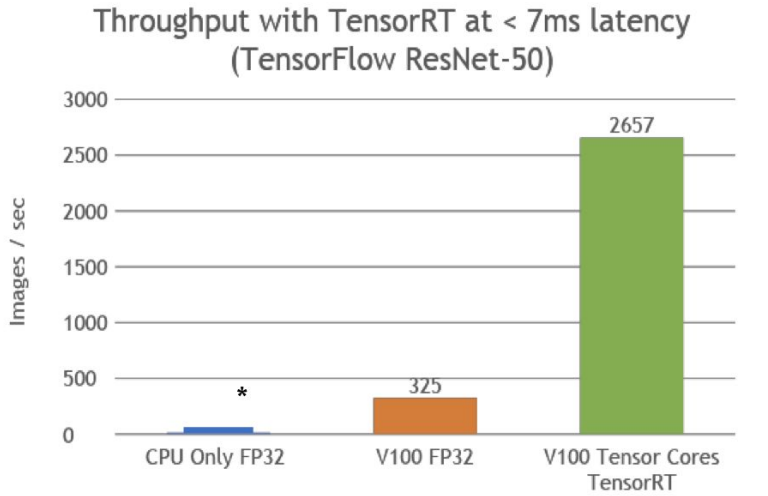
8x higher throughput in TensorRT

- sub-graph optimisation TensorRT
- use custom TensorFlow ops

Available in TensorFlow 1.7

```
from tensor flow.contrib import tensor as trt

trt_graph = trt.create_inference_graph(frozen_graph_def,
            output_node_name,
            max_batch_size=batch_size,
            max_workspace_size_bytes=workspace_size,
            precision_mode=precision)
```



* Min CPU latency measured was 70 ms. It is not < 7 ms.
CPU: Skylake Gold 6140, 2.5GHz, Ubuntu 16.04; 18 CPU threads. Volta V100 SXM;
CUDA (384.111; v9.0.176);
Batch sizes: CPU=1, V100_FP32=2, V100_TensorFlow_TensorRT=16 w/ latency=6ms

NVIDIA TensorRT
Programmable Inference Accelerator

FP16 API FOR CUSTOM LAYERS

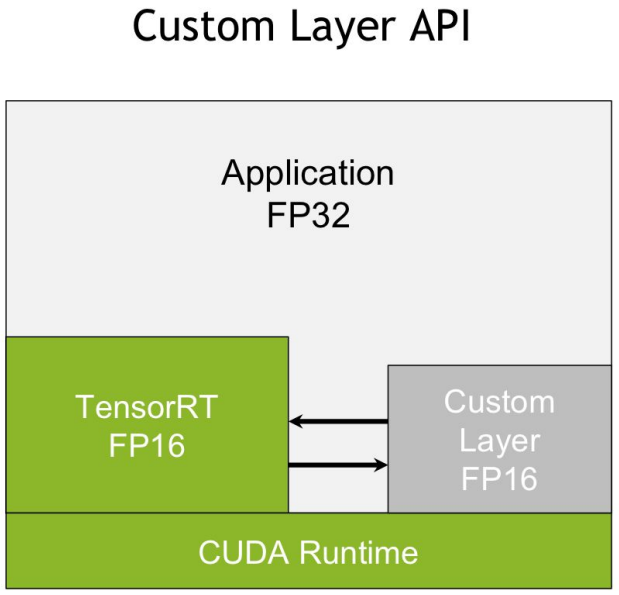
Define, optimize and deploy apps with FP16 custom layers on Tensor Cores

Automotive, Robotics, Video Analytics

Custom layers can now be used in performance critical sections

FP16 is 1.8x faster P100, TX1 or
3-4x faster V100

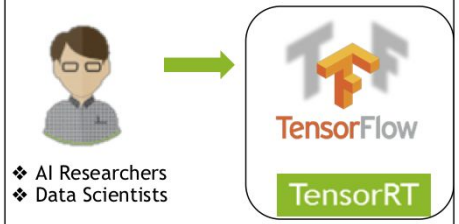
No need to convert Tensors back to FP32



Announcing NVIDIA TensorRT 4 RC

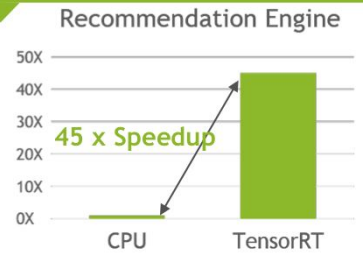
TensorFlow Integration • RNN and MLP • ONNX Import

Speed Up Inference in TensorFlow



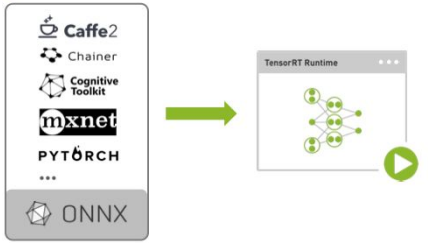
Inference runs 8x faster in TensorFlow on Tesla V100 because they have integrated TensorRT

Translation, Speech and Recommenders



Speed up speech, audio and recommender app inference performance through new layers and optimizations

Optimize & Deploy ONNX Models



User can now easily deploy to GPU. 50x faster ONNX model throughput with TensorRT vs. CPU

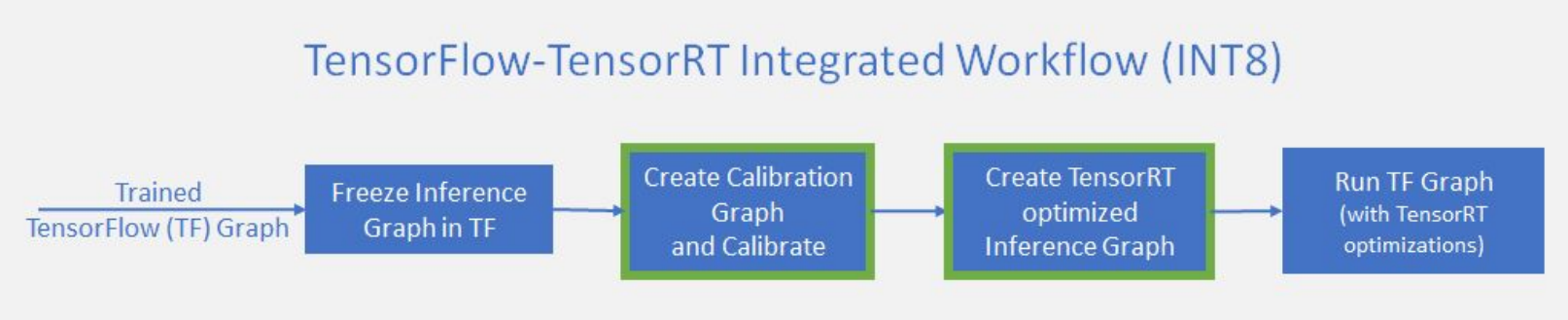
NVIDIA TensorRT
Programmable Inference Accelerator

Inference using INT 8 precision

Performing inference using INT8 precision further improves computation speed and places lower requirements on bandwidth.

	Dynamic Range	Minimum Positive Value
FP32	-3.4×10^{38} ~ $+3.4\times10^{38}$	1.4×10^{-45}
FP16	65504 ~ +65504	5.96×10^{-8}
INT8	-128 ~ +127	1

Inference using INT 8 precision



Converting models for deployment with INT8 requires calibrating the trained FP32 model before applying the TensorRT optimizations described earlier.

NEW RNN AND MLP LAYERS

Maximize Translation, Speech and Recommender Inference Throughput on GPUs

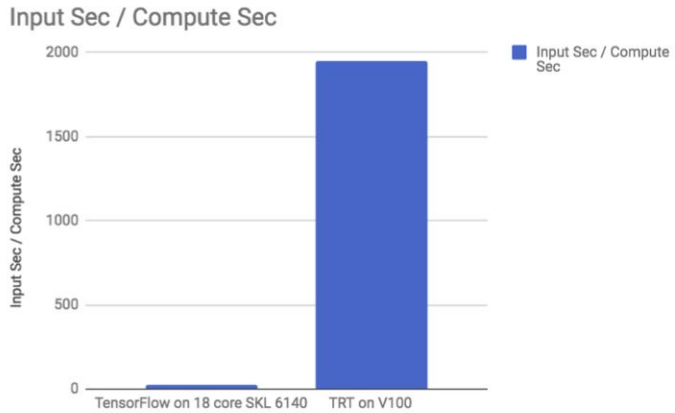
GNMT	Deep Speech 2	MLP
50x	60x	45x

New operators for Gather/Embedding, Top-k, LSTM with Projection, Constant, Softmax and Batch GEMM

Fused Kernel for Stacks of FC + Bias + Activation used in MLP

Easy-to-use APIs (Python/C++) and samples demonstrating attention and beam search

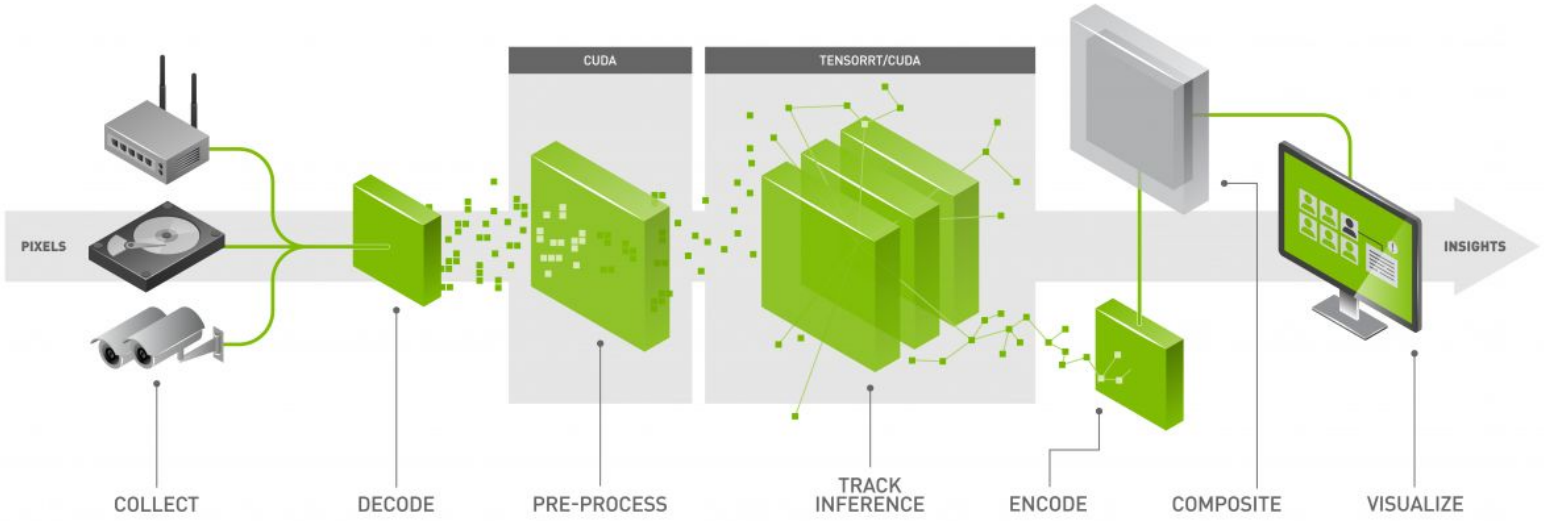
Automatic Speech Recognition
Deep Speech 2



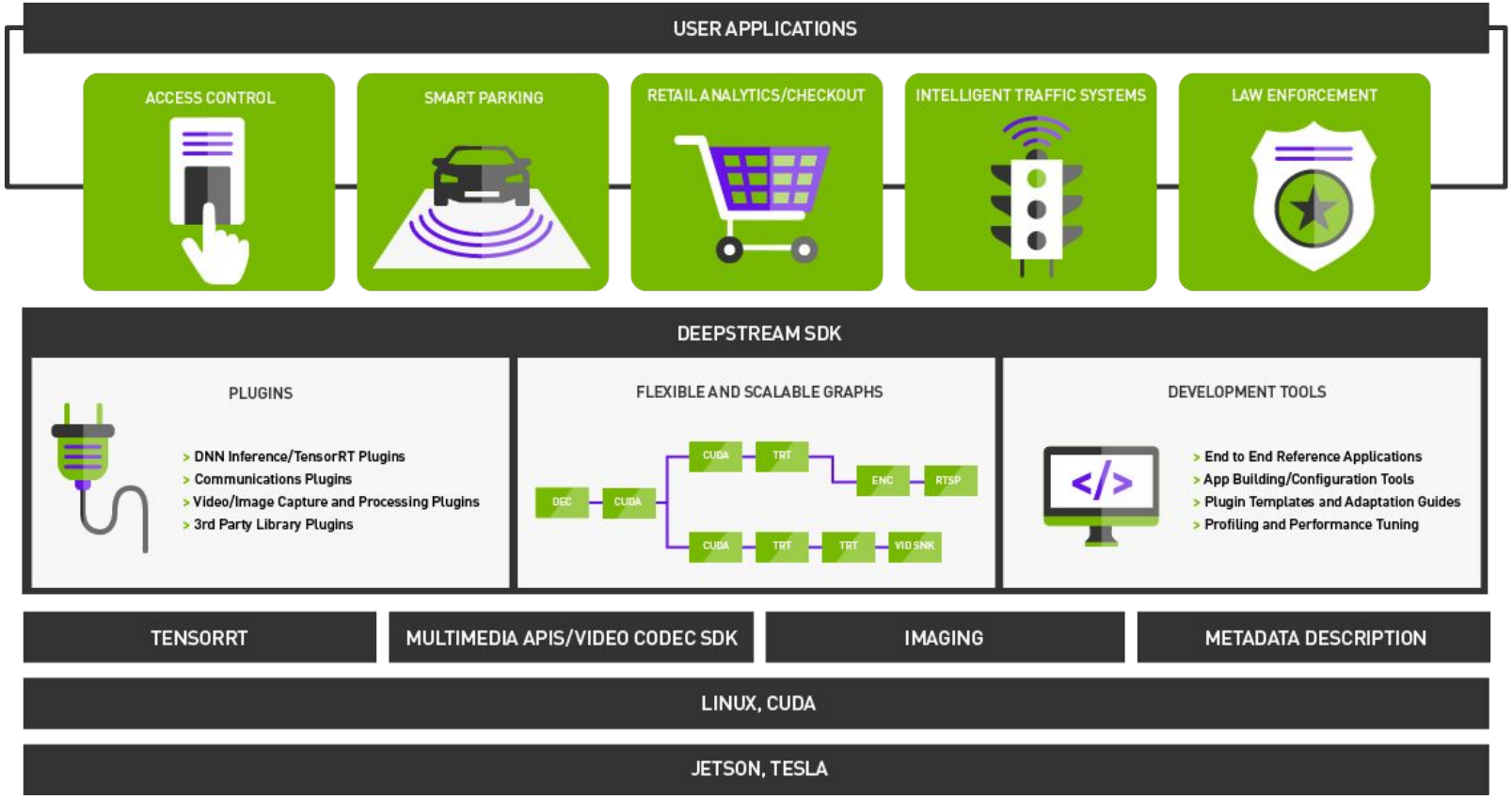
Software	Platform	Seconds of Data processed / Elapsed Second	Speed UP	Batch
TensorFlow	CPU (FP32)	24.7	1X	256
TRT	GPU (FP32)	383	15X	128
TRT	GPU (FP16)	1948	78X	128

The NVIDIA DeepStream SDK

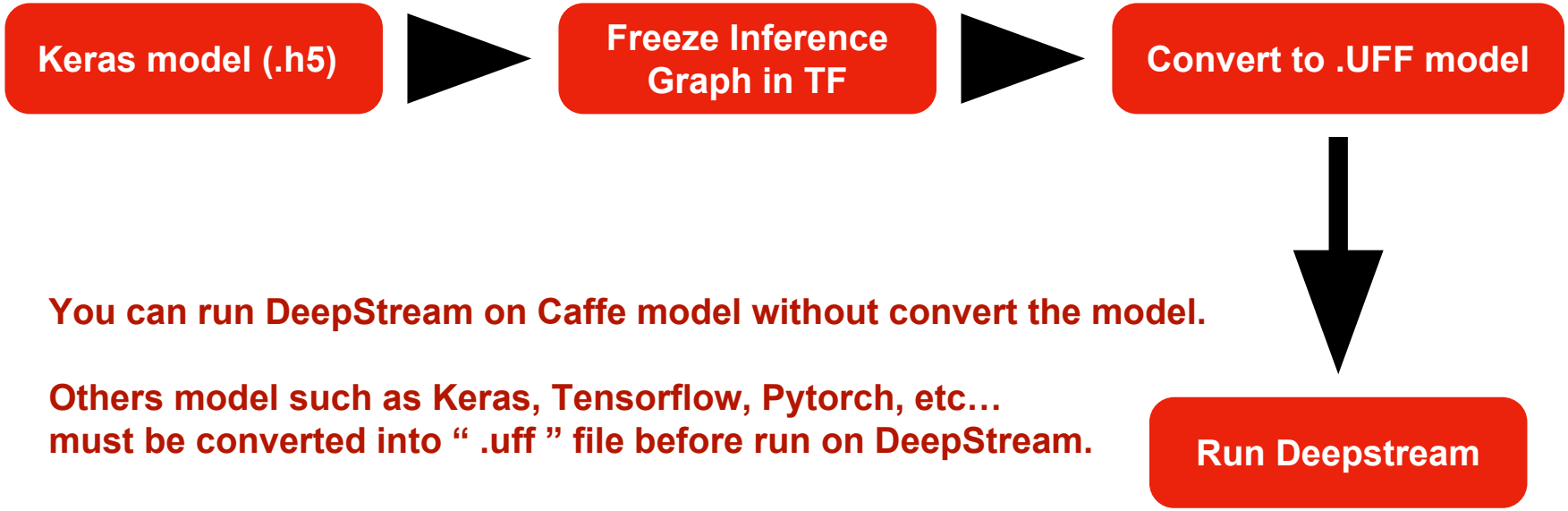
- Creating and deploying AI-based solutions for video analytics applications at scale.
- Offering a complete framework and all the essential building blocks.
- It lets you focus on the core deep learning networks and IP you care about, rather than designing an end-to-end solution from scratch.



DEEPSTREAM SDK



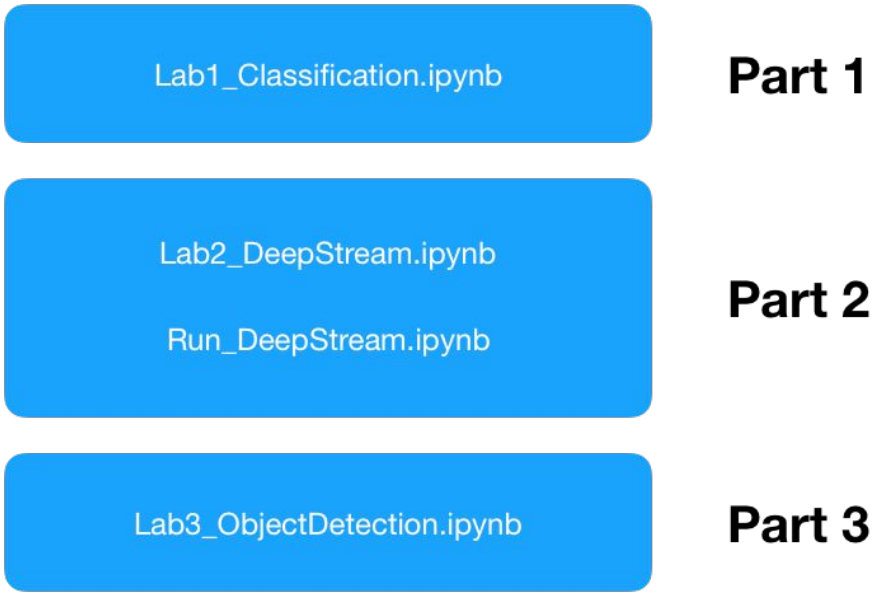
DEEPSTREAM DEPLOYMENT WORKFLOW

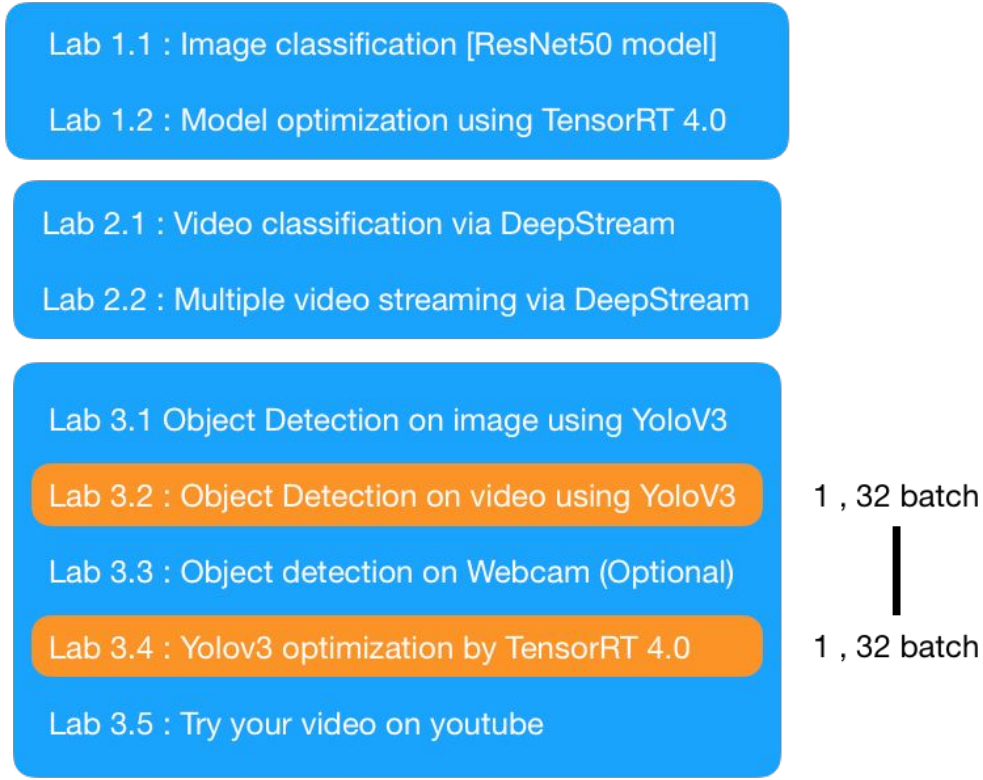


You can run DeepStream on Caffe model without convert the model.

Others model such as Keras, Tensorflow, Pytorch, etc…
must be converted into “ .uff ” file before run on DeepStream.

Lab 2 : Deployment --> consist of 3 parts




Lab 1.1 : Image classification [ResNet50 model]
Lab 1.2 : Model optimization using TensorRT 4.0
Lab 2.1 : Video classification via DeepStream
Lab 2.2 : Multiple video streaming via DeepStream
Lab 3.1 Object Detection on image using YoloV3
Lab 3.2 : Object Detection on video using YoloV3
Lab 3.3 : Object detection on Webcam (Optional)
Lab 3.4 : Yolov3 optimization by TensorRT 4.0
Lab 3.5 : Try your video on youtube
1 , 32 batch
1 , 32 batch