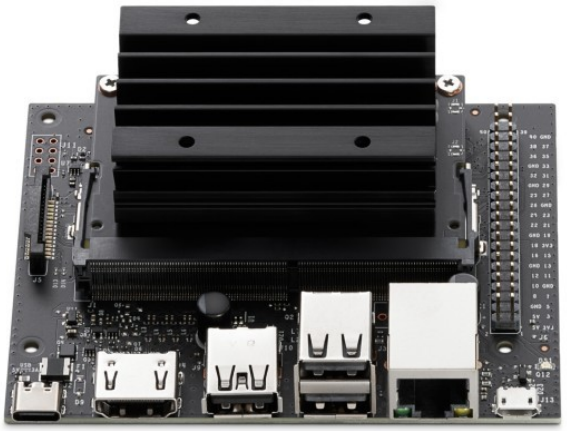
CENG443
Heterogeneous Parallel Programming

Deep Learning Inference with Jetson Nano Developer Kit

Işıl ÖZ, IZTECH, Fall 2023

22 December 2023

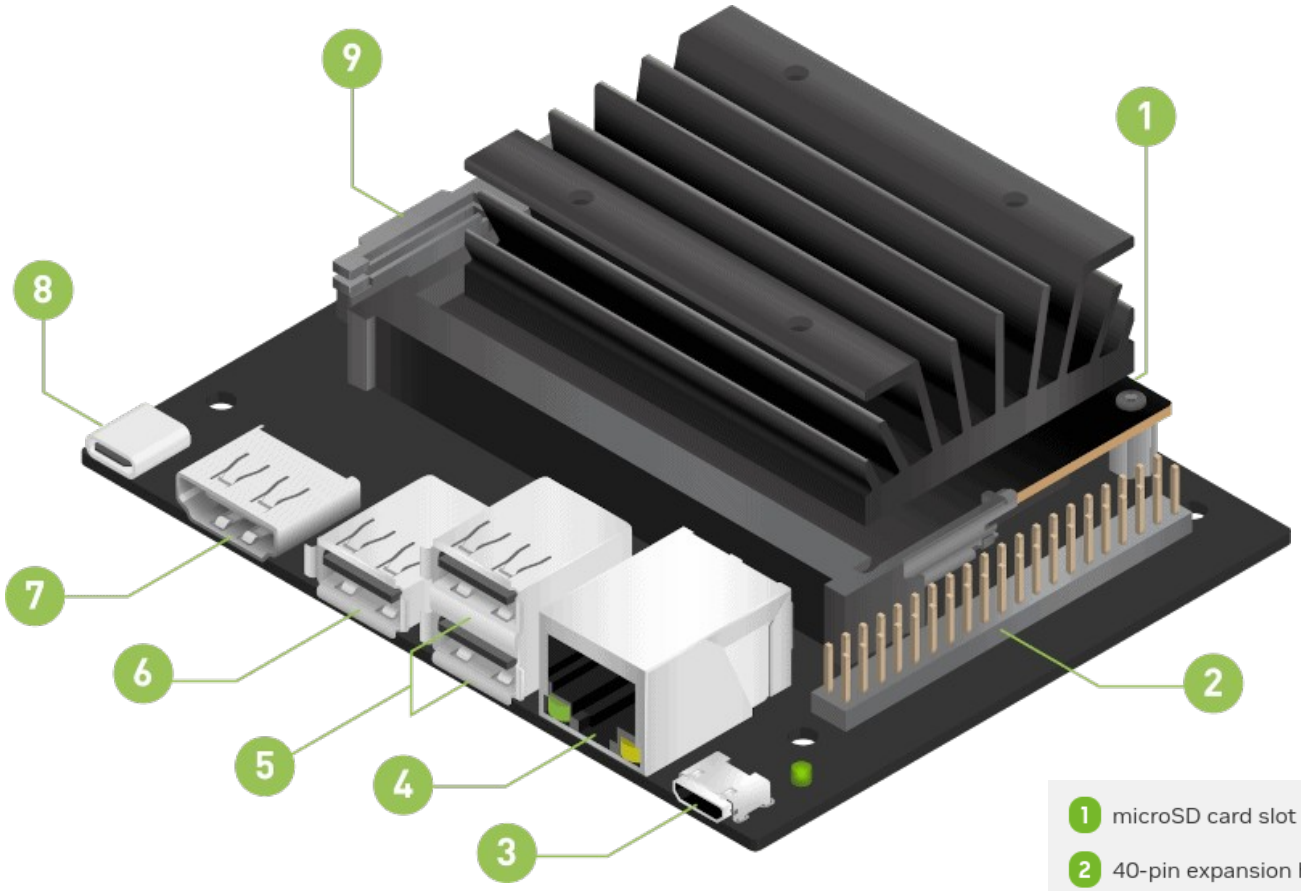
NVIDIA Jetson Nano 2GB Developer Kit



TECHNICAL SPECIFICATIONS

GPU	128-core NVIDIA Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memory	2 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (Card not included)
Video Encode	4Kp30 | 4x 1080p30 | 9x 720p30 (H.264/H.265)
Video Decode	4Kp60 | 2x 4Kp30 | 8x 1080p30 | 18x 720p30 (H.264/H.265)
Connectivity	Gigabit Ethernet 802.11ac wireless†
Camera	1x MIPI CSI-2 connector
Display	HDMI
USB	1x USB 3.0 Type A, 2x USB 2.0 Type A, 1x USB 2.0 Micro-B
Others	40-pin header (GPIO, I²C, I²S, SPI, UART) 12-pin header (Power and related signals, UART) 4-pin Fan header†
Mechanical	100 mm x 80 mm x 29 mm

† Not initially available in all regions



1. microSD card slot for main storage
2. 40-pin expansion header
3. Micro-USB port for Device Mode
4. Gigabit Ethernet port
5. USB 2.0 ports (x2)
6. USB 3.0 port (x1)
7. HDMI output port
8. USB-C for 5V power input
9. MIPI CSI-2 camera connector

Jetson Nano

Jetson Nano is a small, powerful computer for embedded applications and AI IoT that delivers the power of modern AI

NVIDIA JetPack SDK powering the Jetson modules is the most comprehensive solution and provides full development environment for building end-to-end accelerated AI applications and shortens time to market

JetPack 4.6.1

CUDA 10.2

cuDNN 8.2.1

TensorRT 8.2.1

NVIDIA Nsight Systems 2021.5

Other utility software

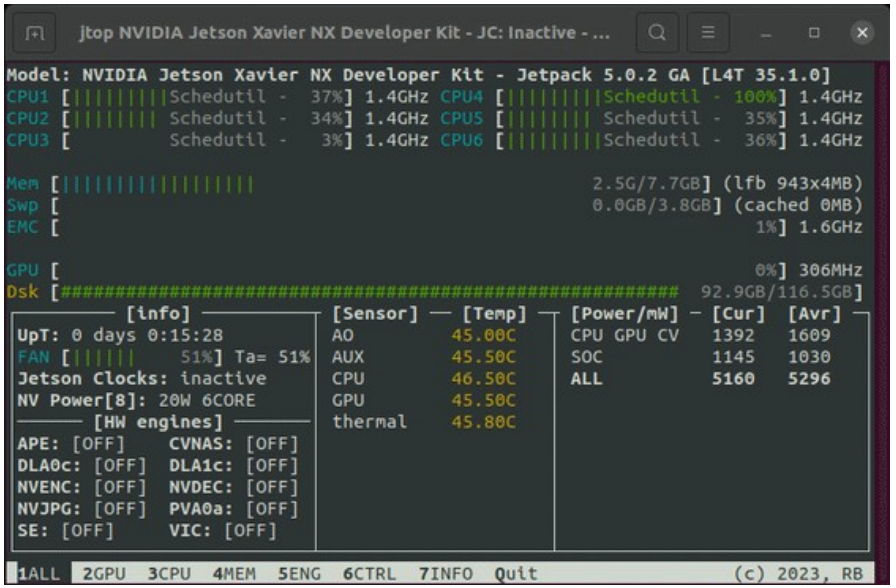
```
$ sudo apt-get install nvidia-jetpack
```

GPU Activities on Jetson Nano

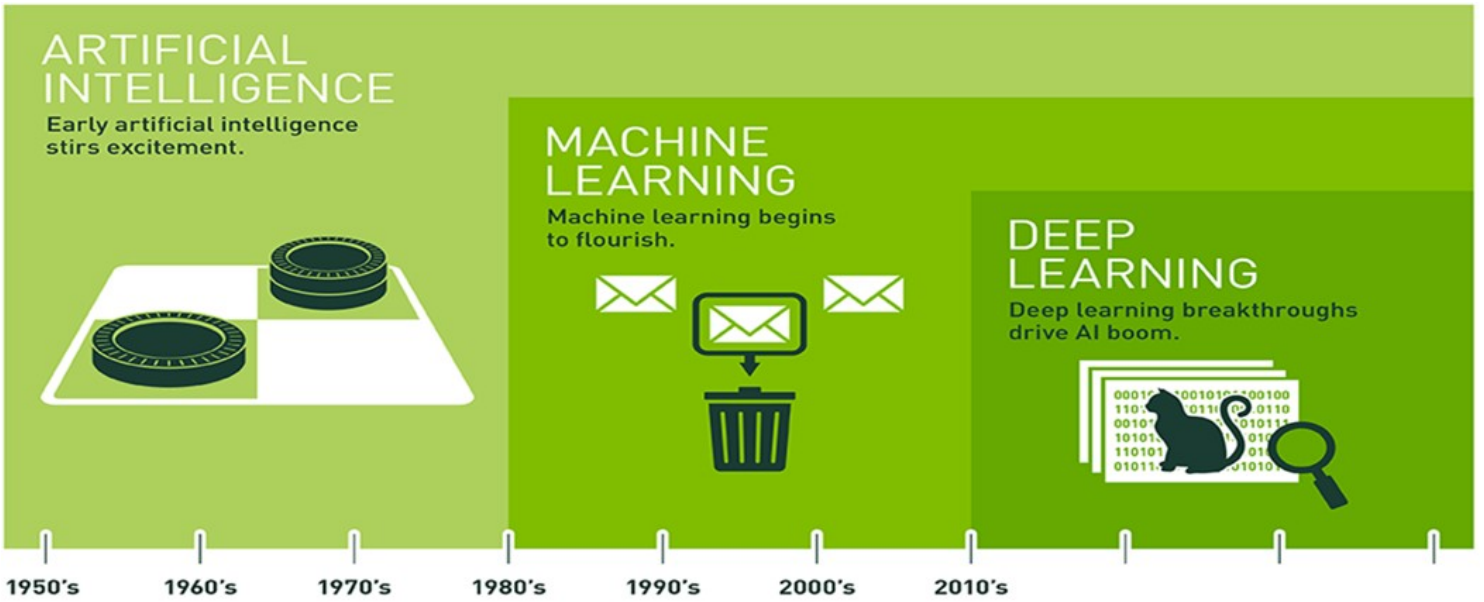
Jetson Stats

package for monitoring and controlling your NVIDIA Jetson

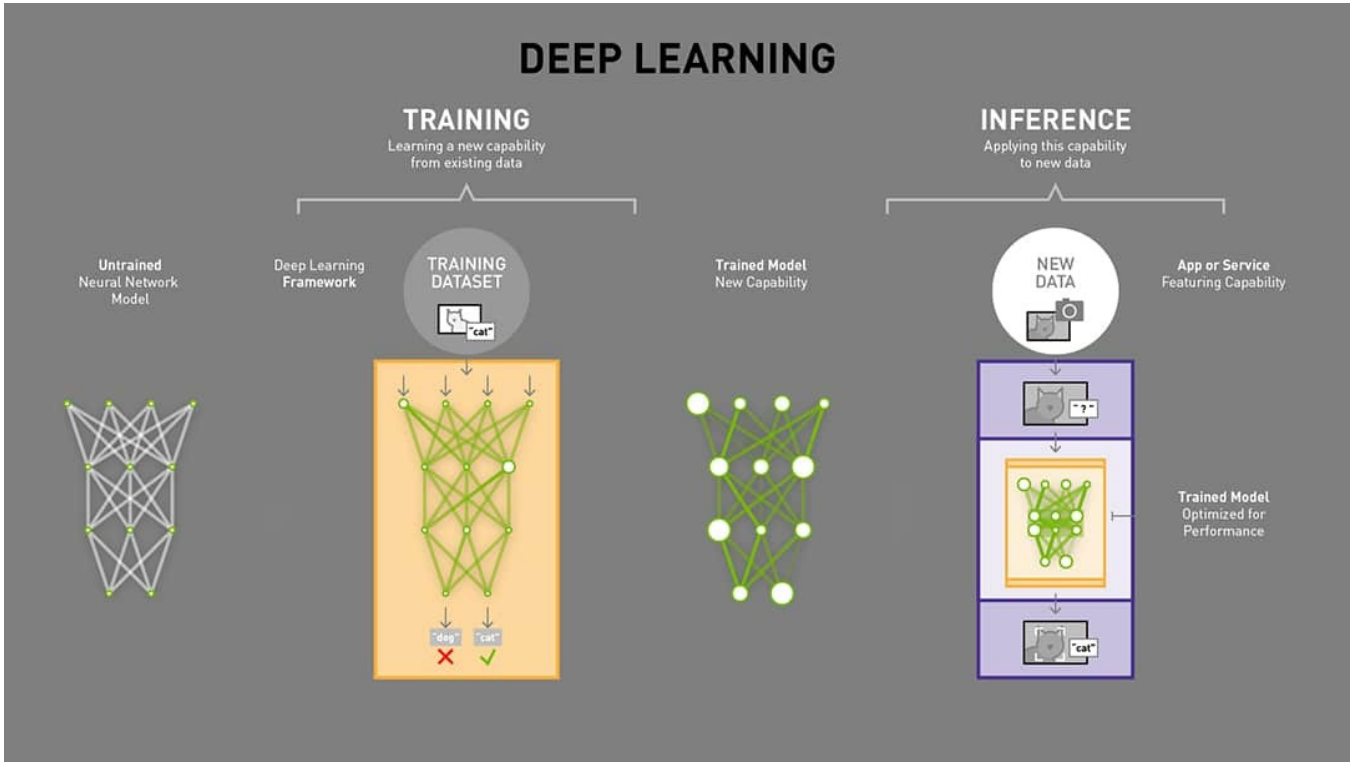
```
$ sudo apt install python-pip -y
# install jetson-stats
$ sudo -H pip install jetson-stats
# reboot jetson
$ sudo reboot

$ jtop
```



AI And Deep Learning



Deep Learning Models



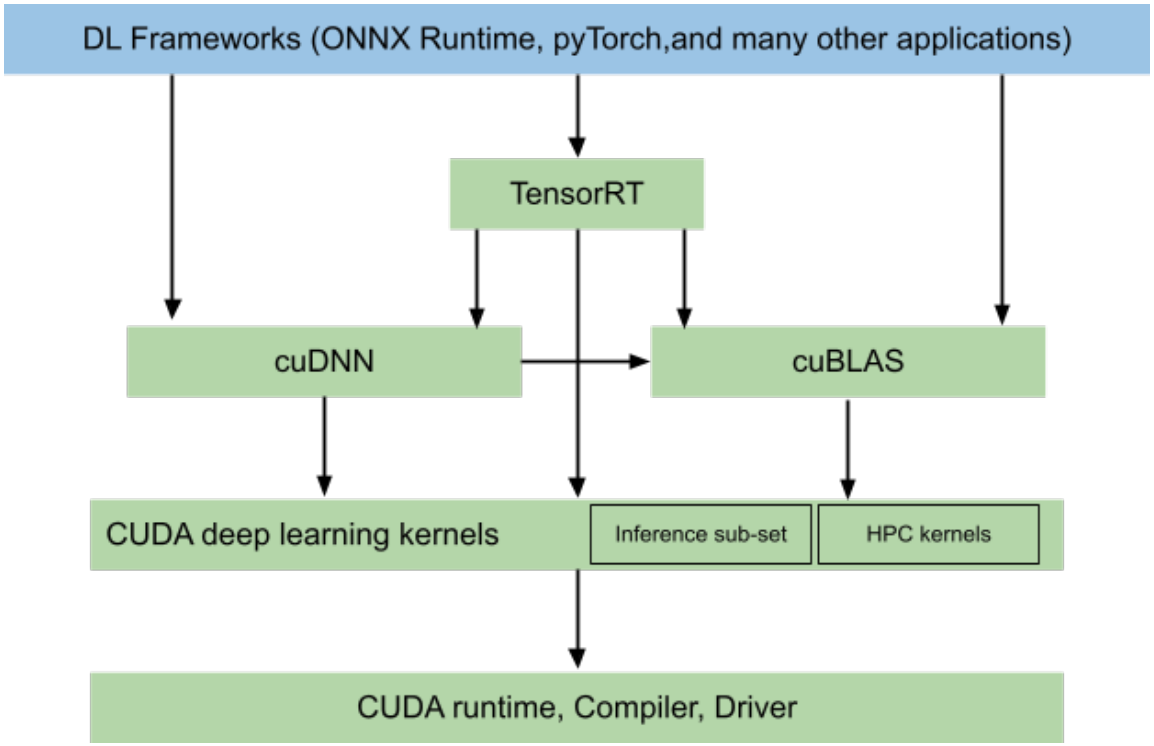
Accelerating DNNs Using GPUs

The extensive calculations required for training DNN models and running inference through trained models can be quite large in number, requiring intensive compute resources and time

Deep learning frameworks such as Caffe, TensorFlow, and PyTorch, are optimized to run faster on GPUs

The Jetson Nano includes a 128-core NVIDIA Maxwell GPU, it can accelerate both training and inference

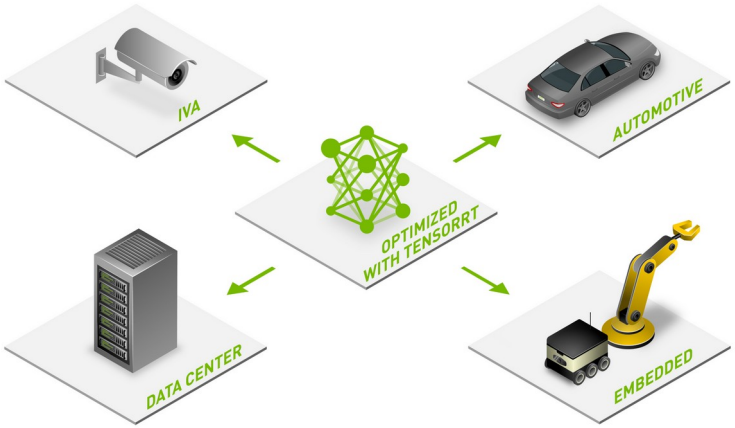
NVIDIA Inference Stack



TensorRT

SDK for high-performance deep learning inference, includes a deep learning inference optimizer and runtime that delivers low latency and high throughput for inference applications

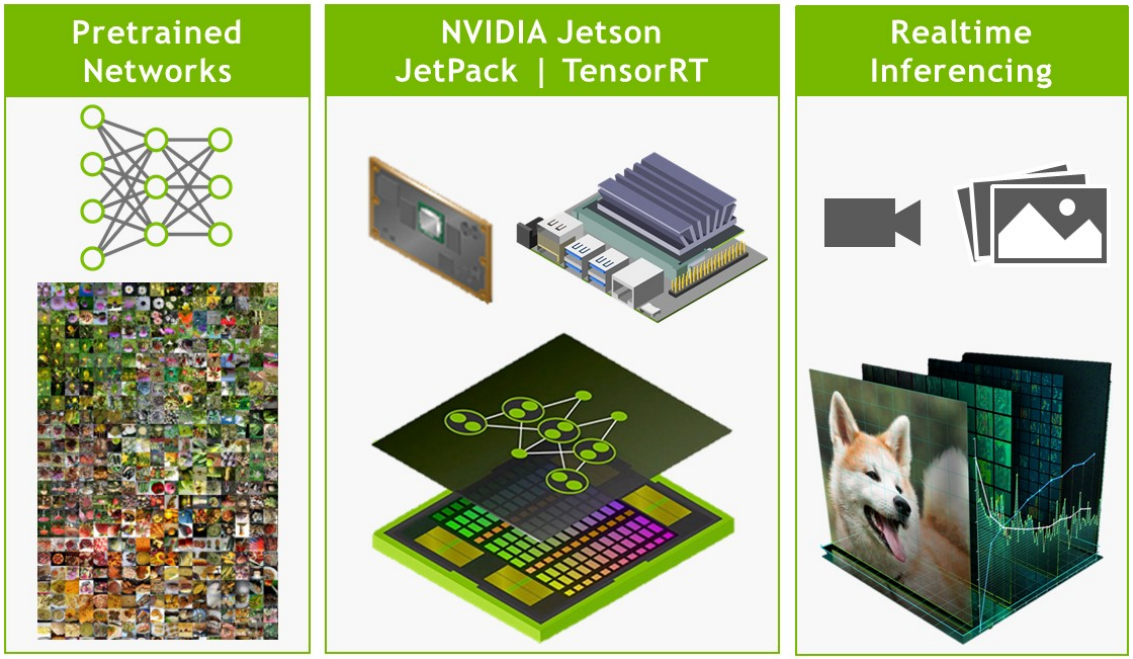
Built on the CUDA parallel programming model



https://github.com/NVIDIA/TensorRT

Hello AI World Guide

Deploying deep-learning inference networks and deep vision primitives with TensorRT and NVIDIA Jetson



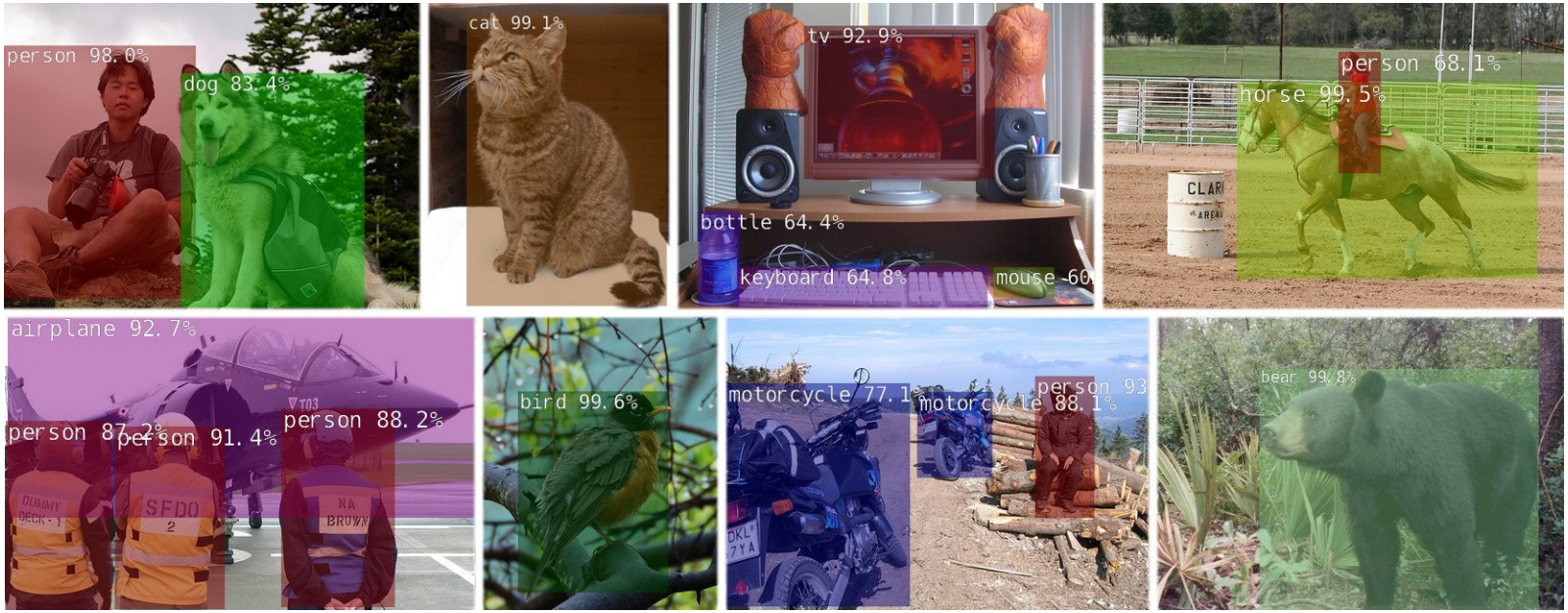
https://github.com/dusty-nv/jetson-inference

Object Detection - Inference

Inference with TensorRT

Detecting objects from images

/jetson-inference-master/examples/detectnet
/jetson-inference-master/c/detectNet.cu
/jetson-inference-master/c/detectNet.cpp
/jetson-inference-master/c/tensorNet.cpp



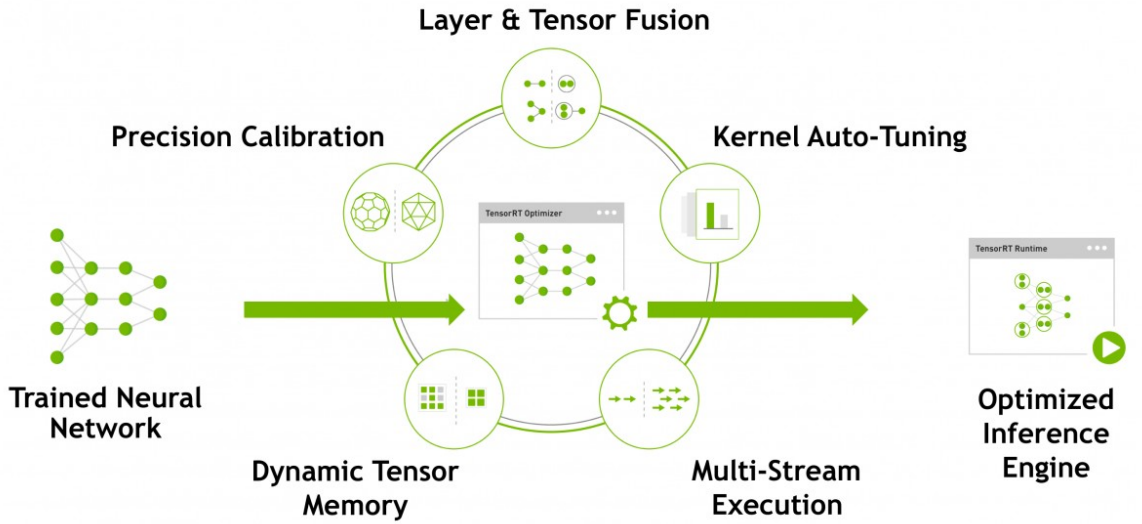
Code/Execution Flow

Load network model

Load engine

Process network

/jetson-inference-master/c/tensorNet.cpp



Pretrained Detection Models

Model	CLI argument	NetworkType enum	Object classes
SSD-Mobilenet-v1	`ssd-mobilenet-v1`	`SSD_MOBILENET_V1`	91 (COCO classes)
SSD-Mobilenet-v2	`ssd-mobilenet-v2`	`SSD_MOBILENET_V2`	91 (COCO classes)
SSD-Inception-v2	`ssd-inception-v2`	`SSD_INCEPTION_V2`	91 (COCO classes)
TAO PeopleNet	`peoplenet`	`PEOPLENET`	person, bag, face
TAO PeopleNet (pruned)	`peoplenet-pruned`	`PEOPLENET_PRUNED`	person, bag, face
TAO DashCamNet	`dashcamnet`	`DASHCAMNET`	person, car, bike, sign
TAO TrafficCamNet	`trafficcamnet`	`TRAFFICCAMNET`	person, car, bike, sign
TAO FaceDetect	`facedetect`	`FACEDETECT`	face

./detectnet --network=ssd-inception-v2 input.jpg output.jpg

Build Project from Source

Quick Reference

If you aren't using the Docker container, here's a condensed form of the commands to build/install the project directly on your Jetson:

```
sudo apt-get update
sudo apt-get install git cmake libpython3-dev python3-numpy
git clone --recursive --depth=1 https://github.com/dusty-nv/jetson-inference
cd jetson-inference
mkdir build
cd build
cmake ../
make -j$(nproc)
sudo make install
sudo ldconfig
```

Execute and Profile on Jetson Nano

```
$ cd /jetson-inference-master/build/aarch64/bin
$ ./detectnet --network=ssd-mobilenet-v2 images/peds_0.jpg images/test/output.jpg
$ ./detectnet "images/humans_*.jpg" images/test/humans_output_%i.jpg
```

Profile nsys or nvprof

```
$ nsys –profile --stats=true ./detectnet ...
```

```
$ sudo /usr/local/cuda/bin/nvprof --csv --log-file output.csv --metrics sm_efficiency ./detectnet ...
```

Project Description

Install JetPack

Build the inference project

https://github.com/dusty-nv/jetson-inference/blob/master/docs/building-repo-2.md

Run and profile (nvprof) detectNet with different networks for all available images in the project

ssd-mobilenet-v2 (default)

ssd-inception-v2

peoplenet

Report the results (execution time, profile results, image outputs)

References

Self-Paced DLI Course: Getting Started with AI on Jetson Nano

https://courses.nvidia.com/courses/course-v1:DLI+S-RX-02+V2/

Deploying Deep Learning project

https://developer.nvidia.com/embedded/twodaystoademo

https://github.com/dusty-nv/jetson-inference/

https://www.youtube.com/watch?v=bcM5AQSAzUY

NVIDIA TensorRT

https://developer.nvidia.com/tensorrt