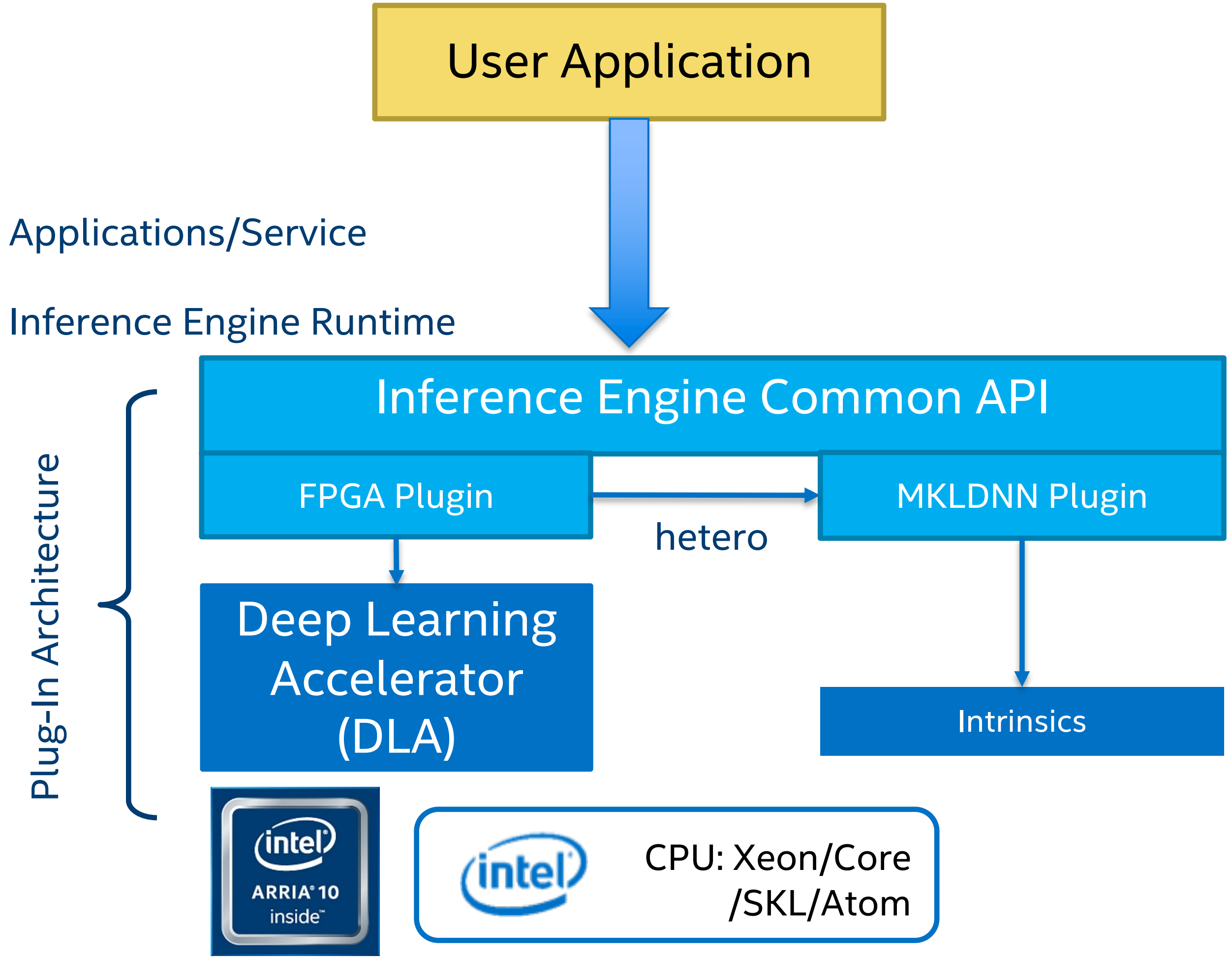
FPGA

Inference Engine Structure



FPGA implementation is "Deep Learning Accelerator" (DLA)

Additional step: FPGA RTE (aocl) loads bitstream with desired version of DLA to FPGA before running

Supported layers

The following layers are supported by the plugin:

- •Batch_norm (being converted by Model Optimizer to ScaleShift layer)
- •Concat
- •Convolution (dilated convolutions are supported, depthwise are not supported)
- •Eltwise (operation sum is supported)
- •Fully Connected
- •LRN Normalization
- •Pooling
- •Power (scale and offset parameters are supported)
- •ReLu (with negative slope)
- •ScaleShift

Note Support is limited to the specific parameters (depending on the bitstream).

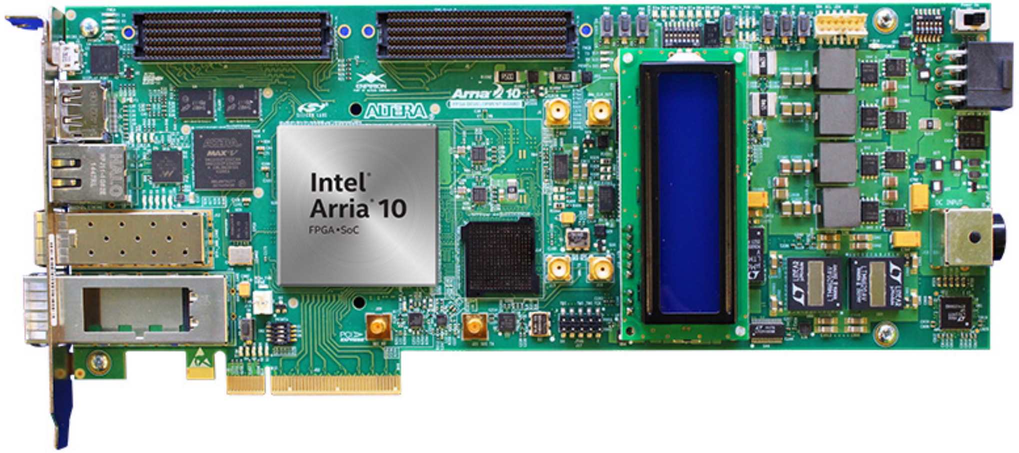
Heterogeneous execution

In case when topology contains layers not supported on FPGA, you need to use **Heterogeneous plugin** with dedicated fallback device.

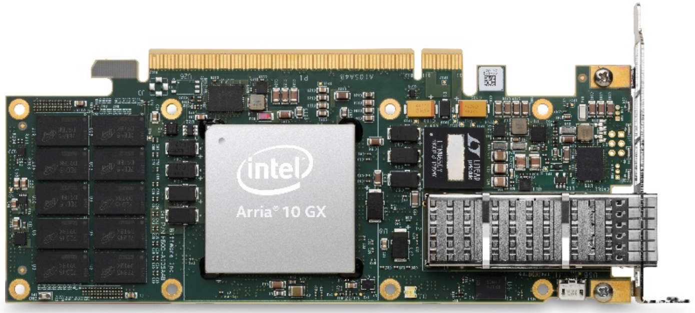
Setup the FPGA Card

You have three options, more are coming up…

Intel® Arria® 10 GX FPGA Development Kit



PCIe Gen3 x 8, full size card
(We are using this card in this session)



Intel® Programmable Acceleration Card with
Intel Arria® 10 GX FPGA

PCIe Gen3 x 8, Half-height half-length card

Deep Learning Acceleration Card with
Intel Arria® 10 GX FPGA

PCIe Gen3 x 8, Half-height half-length card

Optimization Notice

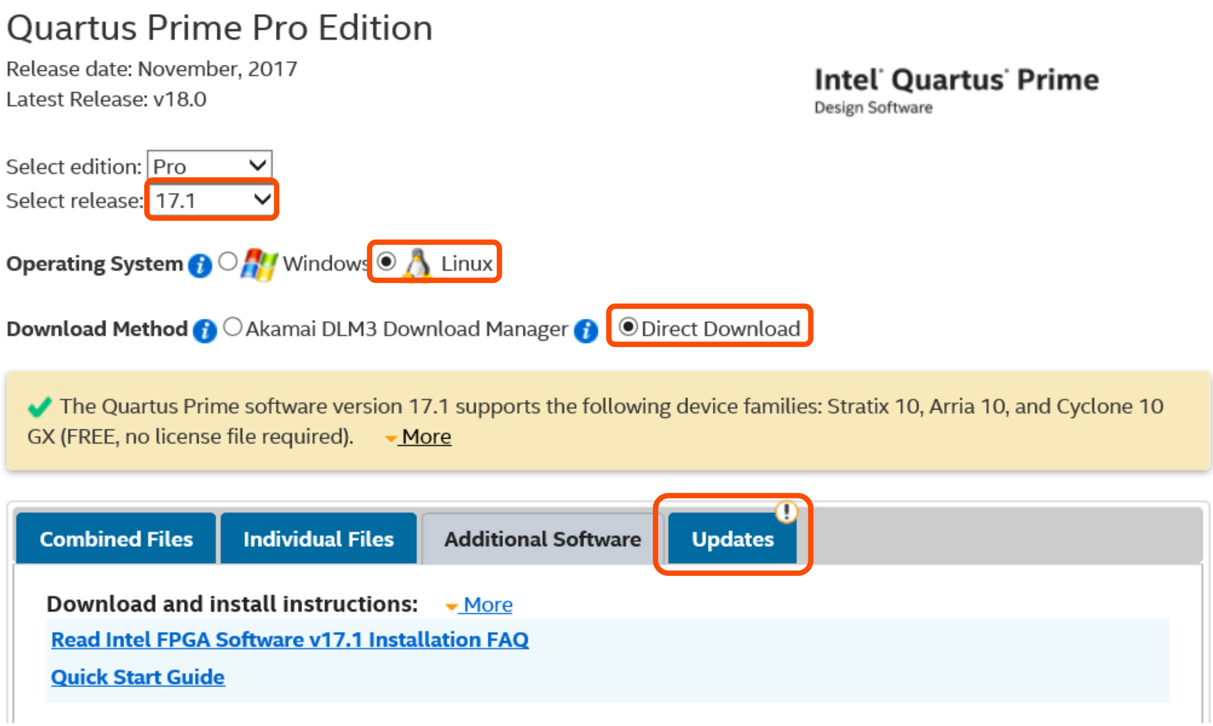
Setup the OS and Software

Setup a clean Ubuntu 16.04, upgrade the library and kernel to 4.13
Until it shows no further update is available…

```
$ sudo apt-get upgrade
[sudo] password for intel: xxxx
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
$ █
```

Setup Quartus tools and OpenCL RT

Select, download and install Quartus Prime Pro Edition v17.1 Update 2



Quartus Prime Pro Edition

Release date: November, 2017
Latest Release: v18.0

Intel® Quartus® Prime
Design Software

Select edition: Pro
Select release: 17.1

Operating System ○ Windows ● Linux

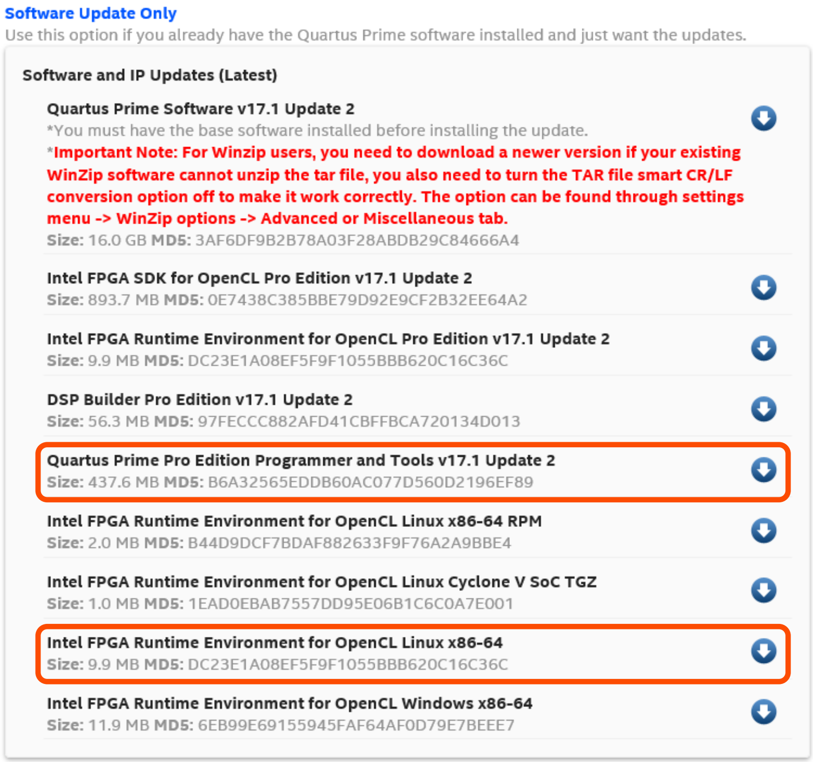
Download Method ○ Akamai DLM3 Download Manager ● Direct Download

✓ The Quartus Prime software version 17.1 supports the following device families: Stratix 10, Arria 10, and Cyclone 10 GX (FREE, no license file required). More

Download and install instructions: More

Read Intel FPGA Software v17.1 Installation FAQ

Quick Start Guide



Software Update Only

Use this option if you already have the Quartus Prime software installed and just want the updates.

Software and IP Updates (Latest)

Quartus Prime Software v17.1 Update 2
*You must have the base software installed before installing the update.
***Important Note: For Winzip users, you need to download a newer version if your existing WinZip software cannot unzip the tar file, you also need to turn the TAR file smart CR/LF conversion option off to make it work correctly. The option can be found through settings menu -> WinZip options -> Advanced or Miscellaneous tab.**
Size: 16.0 GB **MD5:** 3AF6DF9B2B78A03F28ABDB29C84666A4

Intel FPGA SDK for OpenCL Pro Edition v17.1 Update 2
Size: 893.7 MB **MD5:** 0E7438C385BBE79D92E9CF2B32EE64A2

Intel FPGA Runtime Environment for OpenCL Pro Edition v17.1 Update 2
Size: 9.9 MB **MD5:** DC23E1A08EF5F9F1055BBB620C16C36C

DSP Builder Pro Edition v17.1 Update 2
Size: 56.3 MB **MD5:** 97FECCC882AFD41CBFFBCA720134D013

Quartus Prime Pro Edition Programmer and Tools v17.1 Update 2
Size: 437.6 MB **MD5:** B6A32565EDDB60AC077D560D2196EF89

Intel FPGA Runtime Environment for OpenCL Linux x86-64 RPM
Size: 2.0 MB **MD5:** B44D9DCF7BDAF882633F9F76A2A9BBE4

Intel FPGA Runtime Environment for OpenCL Linux Cyclone V SoC TGZ
Size: 1.0 MB **MD5:** 1EAD0EBAB7557DD95E06B1C6C0A7E001

Intel FPGA Runtime Environment for OpenCL Linux x86-64
Size: 9.9 MB **MD5:** DC23E1A08EF5F9F1055BBB620C16C36C

Intel FPGA Runtime Environment for OpenCL Windows x86-64
Size: 11.9 MB **MD5:** 6EB99E69155945FAF64AF0D79E7BEEE7

Setup the OS and Software

Initializing the FPGA card, and soft reboot…

```
$ export PATH=/opt/intelFPGA_pro/17.1/qprogrammer/bin:$PATH
$ quartus_pqm -c 1 -m JTAG -o “p;top.sof”
*******************************************************************
Running Quartus Prime Programmer
Version 17.1.2 Build 304 01/31/2018 SJ Pro Edition
...
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
    Info: Peak virtual memory: 1628 megabytes
    Info: Processing ended: Fri May 11 08:07:51 2018
    Info: Elapsed time: 00:00:21
    Info: Total CPU time (on all processors): 00:00:10
$ sudo reboot
```

Setup the OS and Software

Setup OpenCL RT for FPGA…

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$INTELFPGAOCLSDKROOT/host/
  linux64/lib:$AOCL_BOARD_PACKAGE_ROOT/linux64/lib
$ aocl install
aocl install: Running install from
/tools/aclboardpkg/altera_a10pciedk/16.0/linux64/libexec
...
  CC      /tmp/opencl_driver_x6GjWS/aclpci_a10_ref_drv.mod.o
  LD [M]  /tmp/opencl_driver_x6GjWS/aclpci_a10_ref_drv.ko.unsigned
  NO SIGN [M] /tmp/opencl_driver_x6GjWS/aclpci_a10_ref_drv.ko
make: Leaving directory `/usr/src/kernels/2.6.32-358.el6.x86_64'
$ 
```

Diagnostics

After reboot, run diagnosis and it should show PASSED status

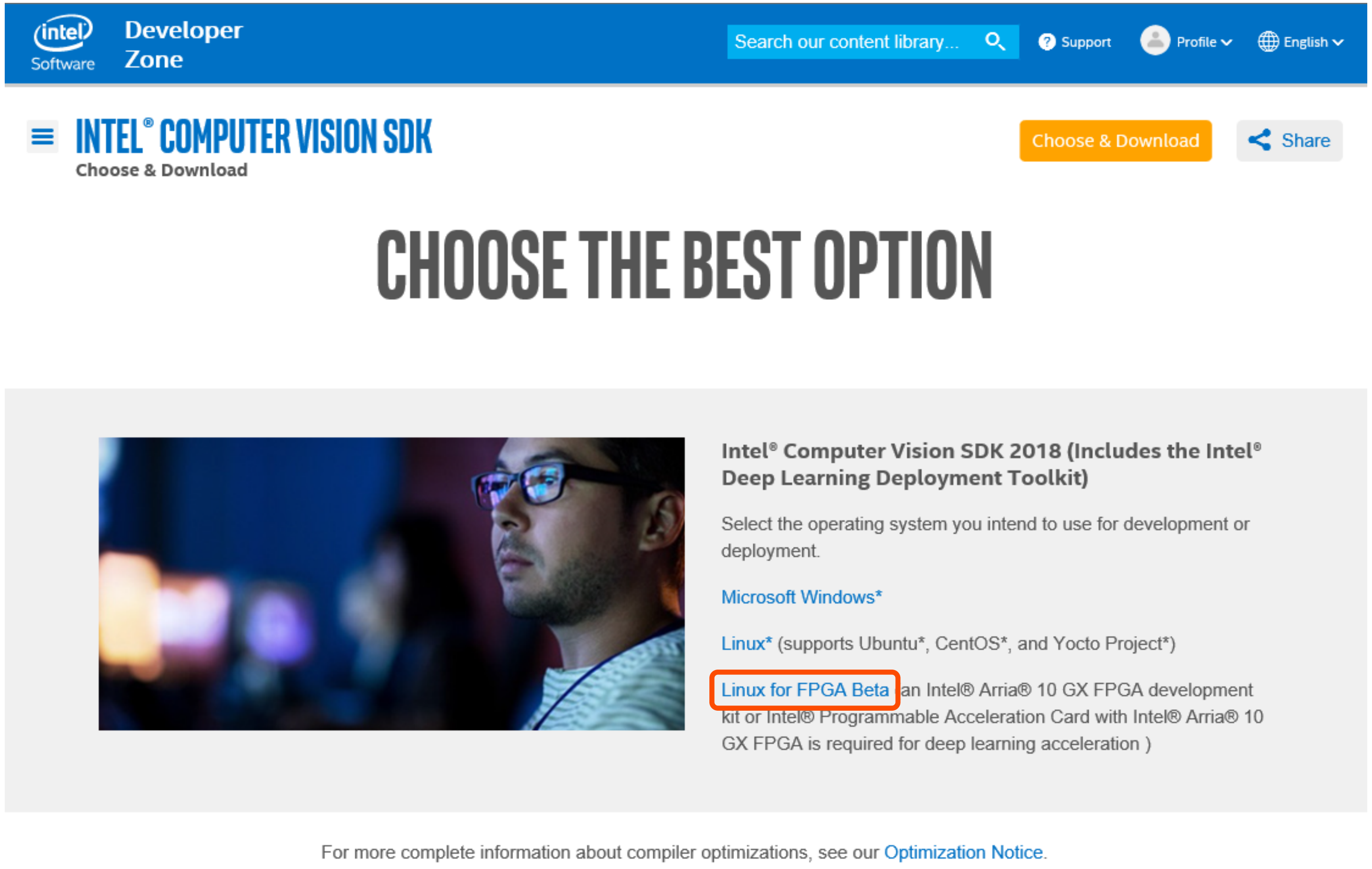
```
$ aocl diagnose
--------------------------------------------------------
...
Vendor: Intel Corporation

Phys Dev Name Status  Information
acla10_ref0   Passed   Arria 10 Reference Platform (acla10_ref0)
                       PCIe dev_id = 2494, bus:slot.func = 04:00.00,
                       Gen3 x8
                       FPGA temperature = 53.3438 degrees C.

DIAGNOSTIC_PASSED
--------------------------------------------------------
$ █
```

Install the OpenVINO™

Follow the instructions to install Open Visual Inference & Neural network Optimization (OpenVINO™)



INTEL® COMPUTER VISION SDK

Choose & Download

CHOOSE THE BEST OPTION

Intel® Computer Vision SDK 2018 (Includes the Intel® Deep Learning Deployment Toolkit)

Select the operating system you intend to use for development or deployment.

Microsoft Windows*

Linux* (supports Ubuntu*, CentOS*, and Yocto Project*)

Linux for FPGA Beta an Intel® Arria® 10 GX FPGA development kit or Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA is required for deep learning acceleration)

For more complete information about compiler optimizations, see our Optimization Notice.

Optimization Notice

Run the Demo!

Targeting vehicle detection and classification workloads to CPU only

```
$ /opt/intel/computer_vision_sdk_fpga_2018.0.234/deployment_tools/inference_
  engine/samples/build/intel64/Release/security_barrier_camera_sample
  -d CPU
  -i /opt/intel/computer_vision_sdk_fpga_2018.0.234/deployment_tools/demo/
     test_video.mp4
  -m /opt/intel/computer_vision_sdk_fpga_2018.0.234/deployment_tools/demo/
     intel_models/vehicle-license-plate-detection-barrier-0007/FP32/vehicle-
     license-plate-detection-barrier-0007.xml
  -m_va opt/intel/computer_vision_sdk_fpga_2018.0.234/deployment_tools/demo/
        intel_models/vehicle-attributes-recognition-barrier-0010/FP32/
        vehicle-attributes-recognition-barrier-0010.xml
$ █
```

Setup the Environment

Before start deploying your workloads to FPGA, setup the environment…

```
$ source /opt/intel/computer_vision_sdk_fpga_2018.0.234/bin/setupvars.sh
$ export LD_LIBRARY_PATH=/opt/intel/computer_vision_sdk_fpga/deployment_
  tools/inference_engine/lib/ubuntu_16.04/intel64:$LD_LIBRARY_PATH
$ export LD_LIBRARY_PATH=/opt/altera/aocl-pro-rte/aclrte-linux64/host/
  linux64/lib:$LD_LIBRARY_PATH
$ █
```

Program the Bitstream to the FPGA

Setup bitstream variables, and load the bitstream to FPGA…

```
$ export CL_CONTEXT_COMPILER_MODE_INTELFPGA=3
$ export DLA_AOCX=/opt/intel/computer_vision_sdk_fpga_2018.0.234/a10_devkit_
  bitstreams/0-8-1_a10dk_fp16_8x48_arch06.aocx
$ source /opt/altera/aocl-pro-rte/aclrte-linux64/init_opencl.sh
$ aocl diagnose
$ aocl program acl0 $DLA_AOCX
Programming device: a10gx : Arria 10 Reference Platform (acla10_ref0)
Reprogramming device [0] with handle 1
Program succeed.
$ █
```

Inference Engine, few FPGA tips

- First iteration with FPGA is always significantly slower than the rest, make sure you run multiple iterations (“-ni”)
- If CPU utilization of concern, minimizes the busy wait time when OMP threads loop in between parallel regions:
  - set KMP_BLOCKTIME=0
  - consider playing OMP_NUM_THREADS
- FPGA performance heavily depends on the bitstream

Run the Demo in the Heterogeneous Mode

Setup the environment properly for heterogeneous mode

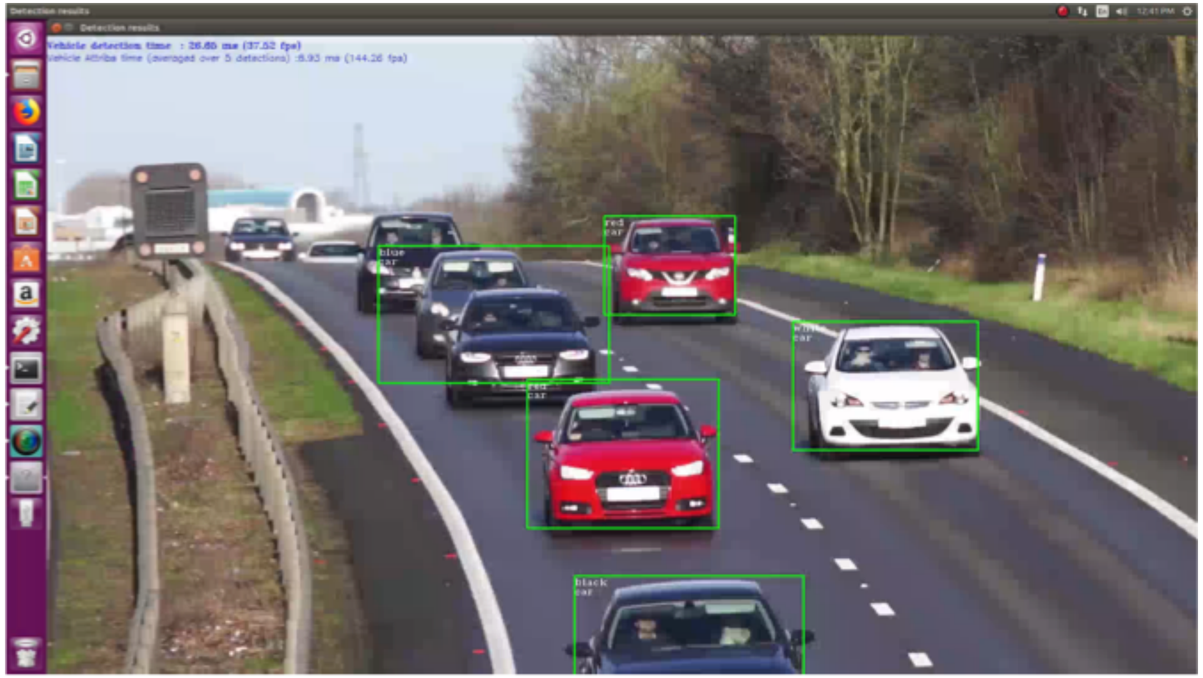
```
$ export KMP_BLOCKTIME=1
$ export OMP_NUM_THREADS=4
$ █
```

Run the Demo on FPGA!

Targeting vehicle detection and classification workloads to FPGA and CPU

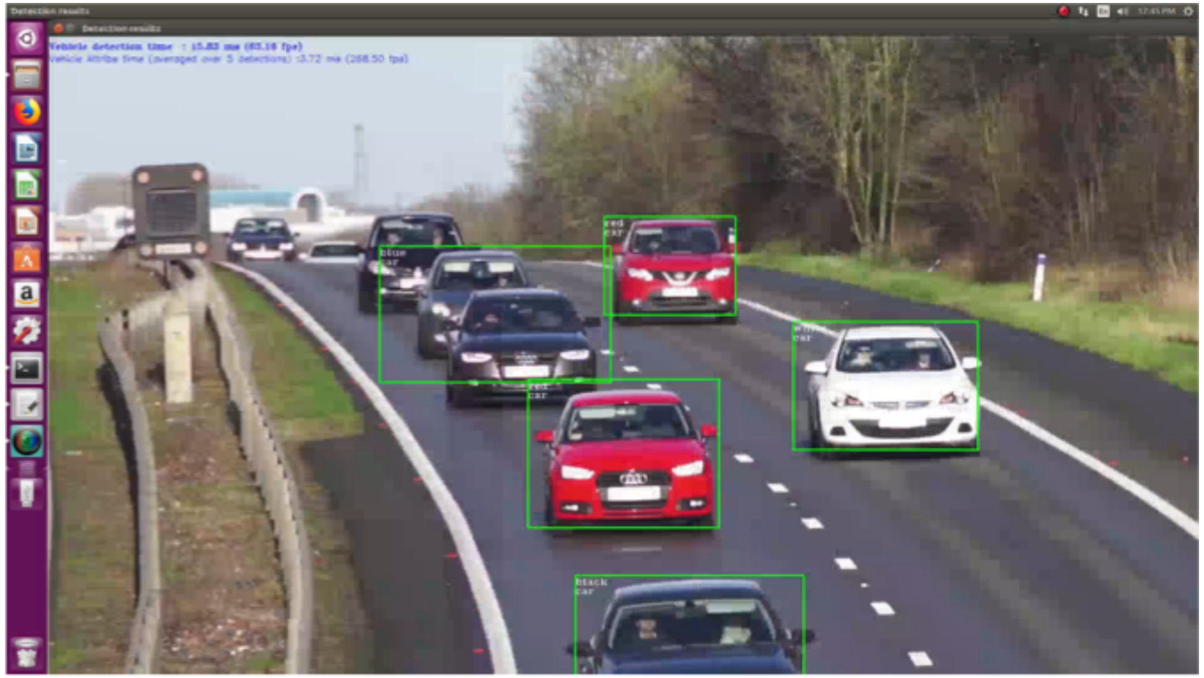
```
$ /opt/intel/computer_vision_sdk_fpga_2018.0.234/deployment_tools/inference_
  engine/samples/build/intel64/Release/security_barrier_camera_sample
  -d HETERO:FPGA,CPU
  -i /opt/intel/computer_vision_sdk_fpga_2018.0.234/deployment_tools/demo/
     test_video.mp4
  -m /opt/intel/computer_vision_sdk_fpga_2018.0.234/deployment_tools/demo/
     intel_models/vehicle-license-plate-detection-barrier-0007/FP32/vehicle-
     license-plate-detection-barrier-0007.xml
  -m_va opt/intel/computer_vision_sdk_fpga_2018.0.234/deployment_tools/demo/
        intel_models/vehicle-attributes-recognition-barrier-0010/FP32/
        vehicle-attributes-recognition-barrier-0010.xml
$ █
```

Contrast


Detection results

Vehicle detection time : 26.65 ms (37.52 fps)
Vehicle Attribs time (averaged over 5 detections) :6.93 ms (144.26 fps)

CPU only without OpenMP tweak


Detection results

Vehicle detection time : 15.83 ms (63.16 fps)
Vehicle Attribs time (averaged over 5 detections) :3.72 ms (268.50 fps)

Heterogeneous mode (FPGA + CPU) with OpenMP tweak

Optimization Notice