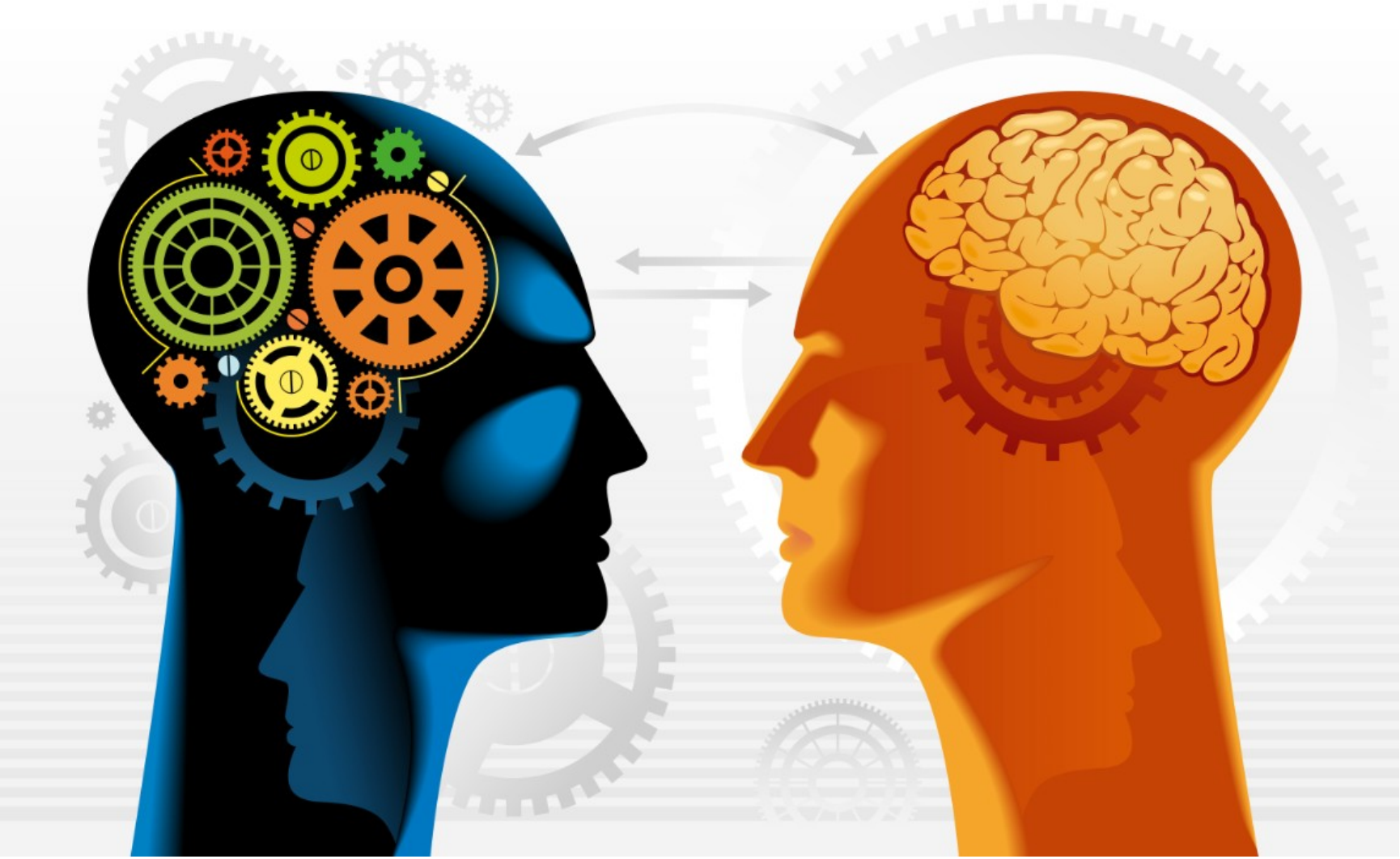
Machine Learning



credit istock

The Intel Neural Compute Stick 2

06.12.2018

Bogdan Vulpescu
LPC+

The edge computing is:

[en] ... a distributed computing paradigm in which computation is largely or completely performed on distributed device nodes known as smart devices or edge devices as opposed to primarily taking place in a centralized cloud environment (en.wikipedia)

[fr] ... une méthode d'optimisation employée dans le cloud computing qui consiste à traiter les données à la périphérie du réseau, près de la source des données (fr.wikipedia)

Primarily it can be used to:

- ingest
- store
- filter
- send

data to cloud systems.

Needs (advantages): low connectivity, low energy consumption, ...

Intel Neural Compute Stick 2 : de l'intelligence offline

Par Bastien Contreras
Le 15 novembre 2018

© Intel

Intel a dévoilé son nouveau Neural Compute Stick 2 (NCS2) lors d'une conférence destinée aux développeurs. Cet outil permet de tester des algorithmes d'intelligence artificielle, sans avoir besoin d'une connexion à Internet.

Les 14 et 15 novembre 2018, Intel tient sa première conférence pour développeurs, dédiée à l'intelligence artificielle, à Pékin. Pour bien commencer, l'entreprise a présenté une nouvelle version de son **Neural Compute Stick**.


INTEL® NEURAL COMPUTE STICK
Home
Intel® Neural Compute Stick 2
MORE CORES.
MORE AI INFERENCE.
Powered by the Intel® Movidius™ Myriad™ X Vision Processing Unit (VPU)
Buy Now
Get Started
Neural Compute Stick 2
Code Samples
Documentation
Forum
Tutorials

Install the Intel® Distribution of OpenVINO™ toolkit with FPGA Support

By Deanne Deuermeyer (Intel), Andrey Z. (Intel), AMY R. (Intel), Fritz B. (Intel), published on May 8, 2018, updated November 14, 2018 Translate

Introduction

The Intel® Distribution of OpenVINO™ toolkit quickly deploys applications and solutions that emulate human vision. Based on Convolutional Neural Networks (CNN), the toolkit extends computer vision (CV) workloads across Intel® hardware, maximizing performance. The Intel Distribution of OpenVINO toolkit includes the Intel® Deep Learning Deployment Toolkit (Intel® DLDT).

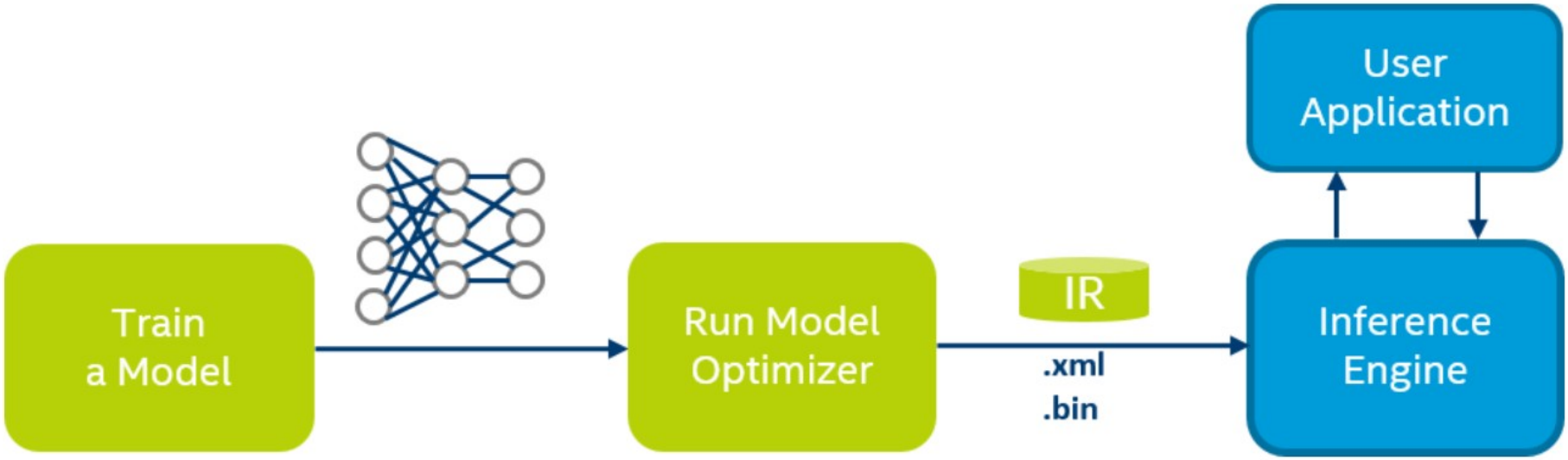
The Intel Distribution of OpenVINO toolkit for Linux* with FPGA Support:

- Enables CNN-based deep learning inference on the edge
- Supports heterogeneous execution across Intel® CPU, Intel® Integrated Graphics, Intel® FPGA, Intel® Movidius™ Neural Compute Stick and Intel® Neural Compute Stick 2
- Speeds time-to-market via an easy-to-use library of computer vision functions and pre-optimized kernels
- Includes optimized calls for computer vision standards including OpenCV*, OpenCL™, and OpenVX*

Model Optimizer Developer Guide

Model Optimizer is a cross-platform command-line tool that facilitates the transition between the training and deployment environment, performs static model analysis, and adjusts deep learning models for optimal execution on end-point target devices.

Model Optimizer process assumes you have a network model trained using a supported deep learning framework. The scheme below illustrates the typical workflow for deploying a trained deep learning model:



Model Optimizer produces an Intermediate Representation (IR) of the network, which can be read, loaded, and inferred with the Inference Engine. The Inference Engine API offers a unified API across a number of supported Intel® platforms. The Intermediate Representation is a pair of files describing the model:

- `.xml` - Describes the network topology
- `.bin` - Contains the weights and biases binary data.

Model	Samples supported on the model	CPU	GPU	HETERO:FPGA,CPU	MYRIAD
face-detection-adas-0001	**Interactive Face Detection Demo**	Supported	Supported	Supported	Supported
age-gender-recognition-retail-0013	**Interactive Face Detection Demo**	Supported	Supported	Supported	Supported
head-pose-estimation-adas-0001	**Interactive Face Detection Demo**	Supported	Supported	Supported	Supported
emotions-recognition-retail-0003	**Interactive Face Detection Demo**	Supported	Supported	Supported	Supported
facial-landmarks-35-adas-0001	**Interactive Face Detection Demo**	Supported	Supported	Supported	
vehicle-license-plate-detection-barrier-0106	**Security Barrier Camera Demo**	Supported	Supported	Supported	Supported
vehicle-attributes-recognition-barrier-0039	**Security Barrier Camera Demo**	Supported	Supported	Supported	Supported
license-plate-recognition-barrier-0001	**Security Barrier Camera Demo**	Supported	Supported	Supported	Supported
person-detection-retail-0001	**Object Detection Demo**	Supported	Supported	Supported	
person-vehicle-bike-detection-crossroad-0078	**Crossroad Camera Demo**	Supported	Supported	Supported	Supported
person-attributes-recognition-crossroad-0031	**Crossroad Camera Demo**	Supported	Supported	Supported	Supported
person-reidentification-retail-0031	**Crossroad Camera Demo** **Pedestrian Tracker Demo**	Supported	Supported	Supported	Supported
person-reidentification-retail-0076	**Crossroad Camera Demo**	Supported	Supported	Supported	Supported
person-reidentification-retail-0079	**Crossroad Camera Demo**	Supported	Supported	Supported	Supported
road-segmentation-adas-0001	**Image Segmentation Demo**	Supported	Supported		
semantic-segmentation-adas-0001	**Image Segmentation Demo**	Supported	Supported		

Supported Devices

The Inference Engine can infer models in different formats with various input and output formats. This section provides supported and optimal configurations per device.

The Inference Engine provides unique capabilities to infer deep learning models on the following device types with corresponding plugins:

Plugin	Device types
GPU plugin	Intel® Processor Graphics, including Intel® HD Graphics and Intel® Iris® Graphics
CPU plugin	Intel® Xeon® with Intel® AVX2 and AVX512, Intel® Core™ Processors with Intel® AVX2, Intel® Atom® Processors with Intel® SSE
FPGA plugin	Intel® Arria® 10 GX FPGA Development Kit, Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA, Intel® Vision Accelerator Design with Intel® Arria® 10 FPGA
VPU plugins	Intel® Movidius™ Myriad™ 2 Vision Processing Unit
GNA plugin	Intel® Speech Enabling Developer Kit, Amazon Alexa* Premium Far-Field Developer Kit, Intel® Pentium® Silver processor J5005, Intel® Celeron® processor J4005, Intel® Core™ i3-8121U processor
Heterogeneous plugin	Heterogeneous plugin enables computing for inference on one network on several Intel® devices.

Demo: image classification

Runs inference for one SqueezeNet model:

- 66 layers

Classifies in 1000 classes (not only cars), top 10 are shown:

- sport car
- convertible
- car wheel
- racing car ...

```
817 0.8295898 label sport car
511 0.0961304 label convertible
479 0.0439453 label car wheel
751 0.0101318 label racing car
...
```

```
654 0.8525391 label minibus
656 0.1436768 label minivan
...
7 0.0000000 label cock
3 0.0000000 label tiger shark
8 0.0000000 label hen
```

```
511 0.7343750 label convertible
817 0.2171631 label sport car
...
754 0.0006294 label radio, wireless
674 0.0004752 label mousetrap
```

“galaxy” is not in the list of labels !

```
971 0.4194336 label bubble
111 0.1340332 label nematode, roundworm
107 0.0965576 label jellyfish
909 0.0517273 label wok
818 0.0415344 label spotlight, spot
...
```

```
260 0.9184570 label chow, chow chow
261 0.0754395 label keeshond
259 0.0060005 label Pomeranian
154 0.0001284 label Pekinese, Pekingese, Peke
231 0.0000688 label collie
...
```

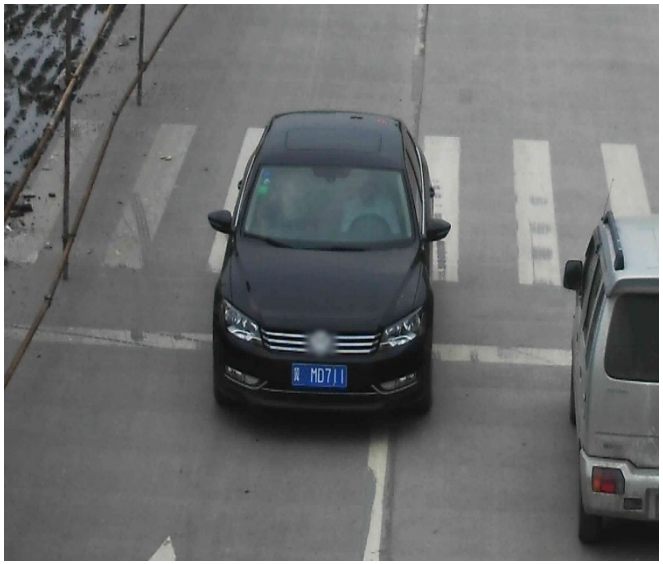
Demo: security barrier camera

Runs inferences for 3 (pre-trained) models:

- vehicle and license plate recognition
  - 143 layers
- vehicle attributes model
  - 31 layers
- license plate recognition
  - 159 layers

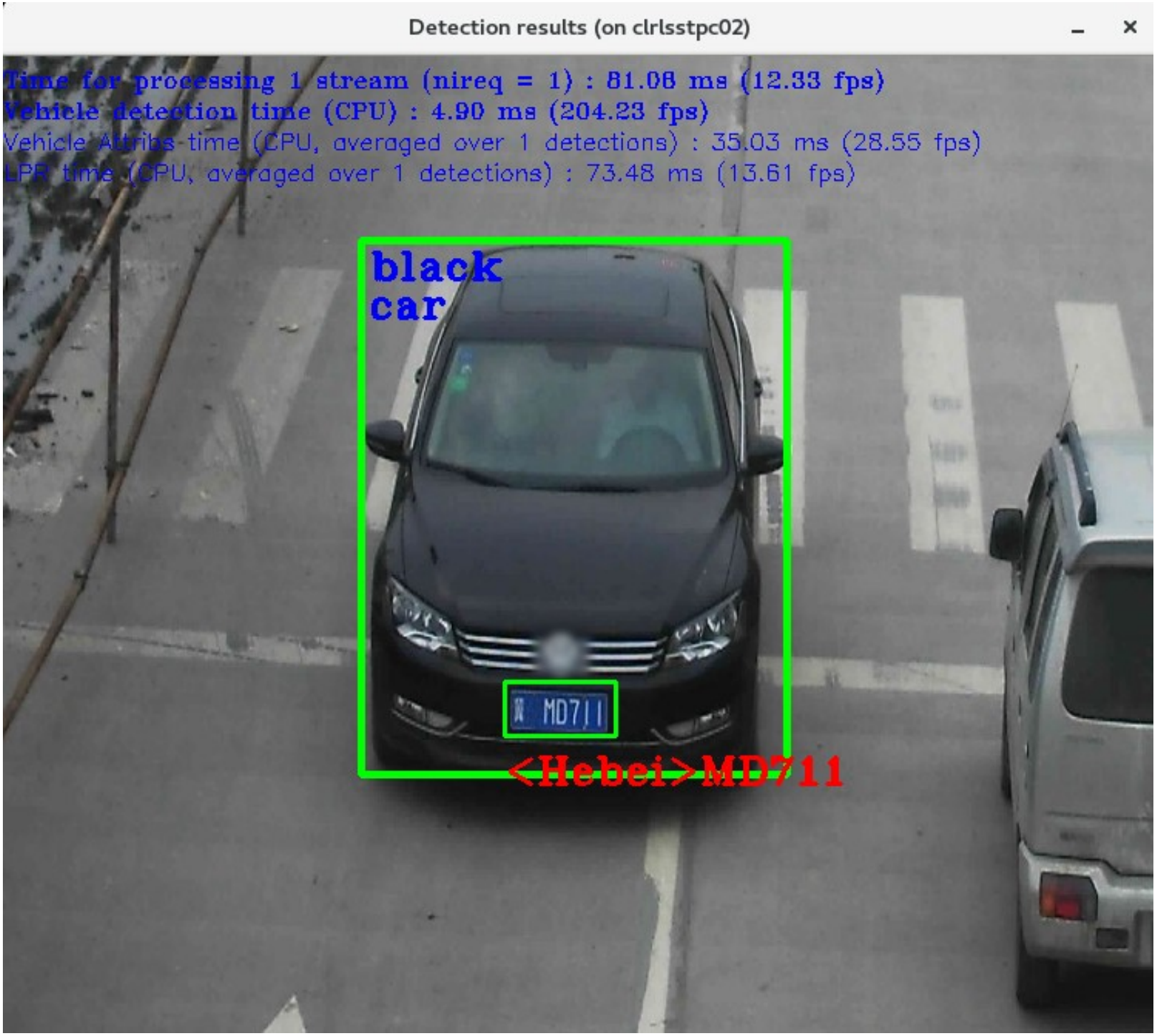
Source framework: TensorFlow

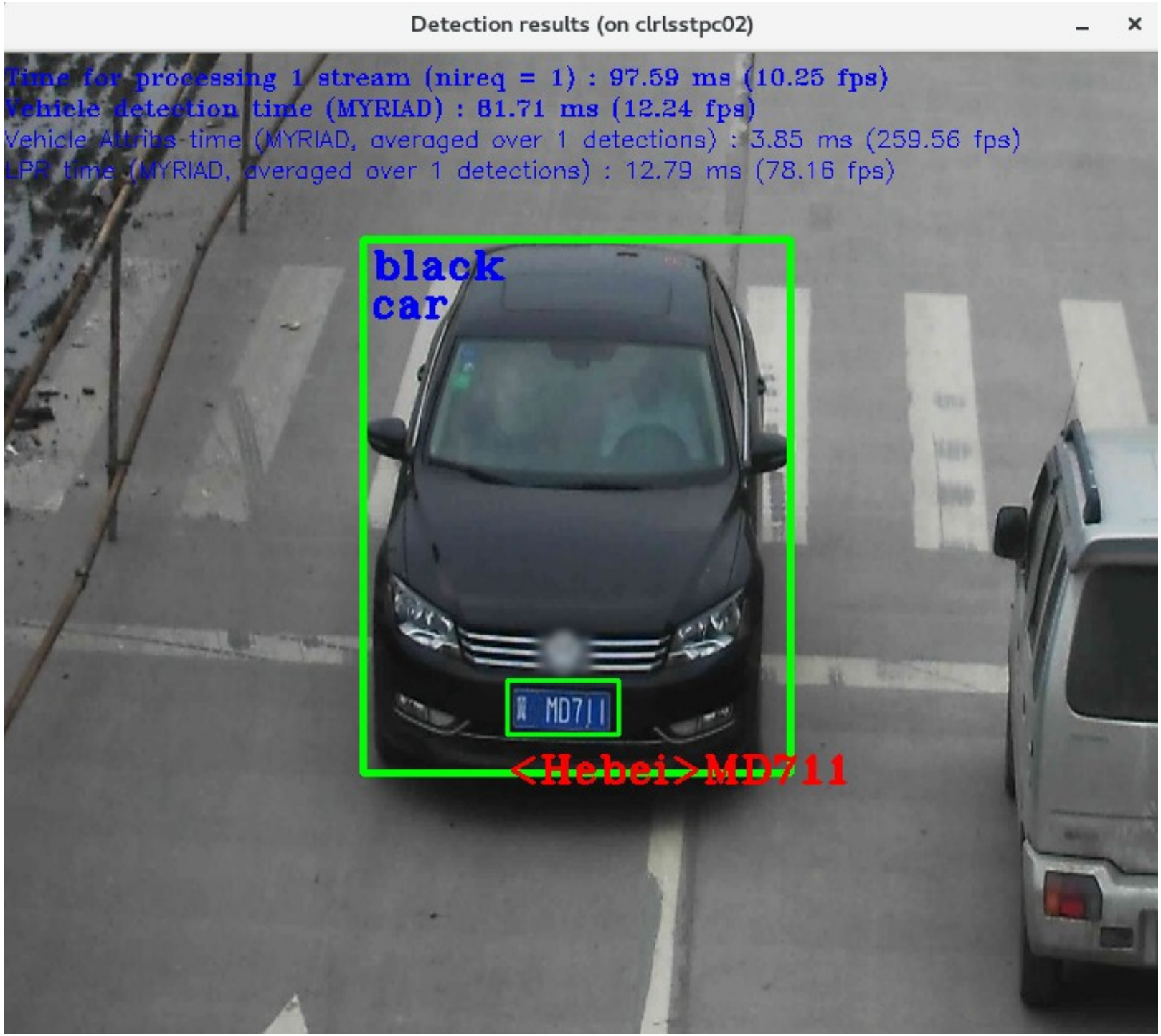
Input = 1x3x300x300 image (BxCxHxW)




Detection results
Time for processing 1 stream (nireq = 1) : 71.05 ms (14.07 fps)
Vehicle detection time (CPU) : 14.01 ms (71.36 fps)
Vehicle Attribs time (CPU, averaged over 1 detections) : 17.56 ms (56.94 fps)
LPR time (CPU, averaged over 1 detections) : 54.98 ms (18.19 fps)
black
car
MD711
<Hebei>MD711


Detection results
Time for processing 1 stream (nireq = 1) : 117.58 ms (8.50 fps)
Vehicle detection time (MYRIAD) : 98.53 ms (10.15 fps)
Vehicle Attribs time (MYRIAD, averaged over 1 detections) : 15.32 ms (65.26 fps)
LPR time (MYRIAD, averaged over 1 detections) : 13.82 ms (72.38 fps)
black
car
MD711
<Hebei>MD711


Detection results (on clrlsstpc02)
Time for processing 1 stream (nireq = 1) : 81.08 ms (12.33 fps)
Vehicle detection time (CPU) : 4.90 ms (204.23 fps)
Vehicle Attribs time (CPU, averaged over 1 detections) : 35.03 ms (28.55 fps)
LPR time (CPU, averaged over 1 detections) : 73.48 ms (13.61 fps)
black
car
MD711
<Hebei>MD711


Detection results (on clrlsstpc02)
Time for processing 1 stream (nireq = 1) : 97.59 ms (10.25 fps)
Vehicle detection time (MYRIAD) : 81.71 ms (12.24 fps)
Vehicle Attribs time (MYRIAD, averaged over 1 detections) : 3.85 ms (259.56 fps)
LPR time (MYRIAD, averaged over 1 detections) : 12.79 ms (78.16 fps)
black
car
冀 MD711
<Hebei>MD711

from google images

Demo 1) total inference time:

aliceport06		lsstpc02
CPU	VPU(USB2)	VPU(USB3)
6.1042681	14.1264470	9.5164888

Demo 2) average inference time:

aliceport06		lsstpc02
CPU	VPU(USB2)	VPU(USB3)
71.054 ms	117.585 ms	97.5866 ms