P2M PROJECT:

SUP'COM
Ecole Supérieure des Communications de Tunis

1. Documentation

Intel Neural Compute Stick2 (NCS2) :

NCS2 acts as a USB accelerator

Intel Movidius Myriad X Vision Processing Unit (VPU) a specialized processors for machine learning

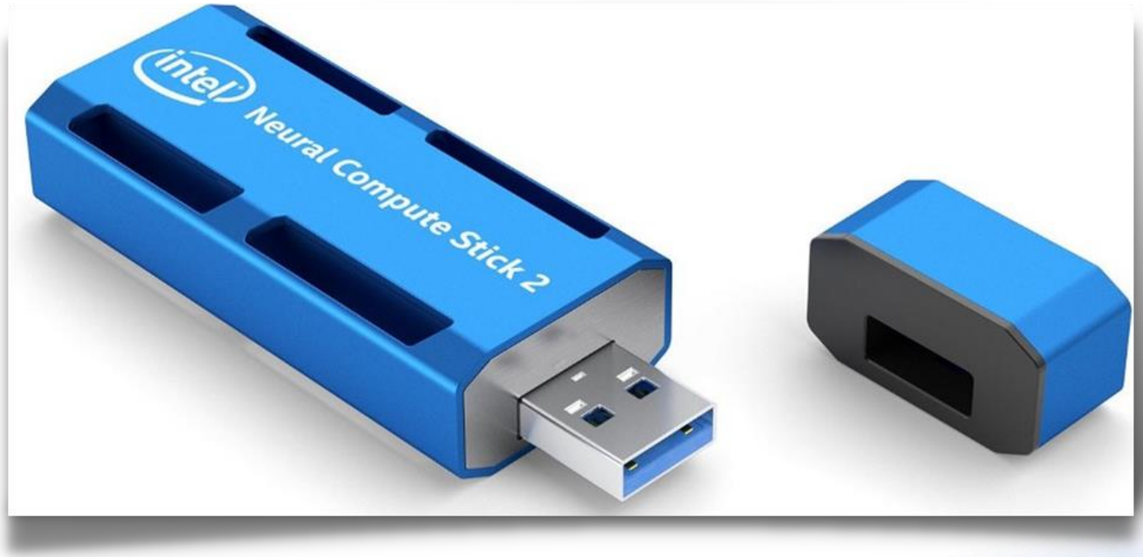
---

Accelerates TensorFlow neural network inferences and improves performances by 10x factor

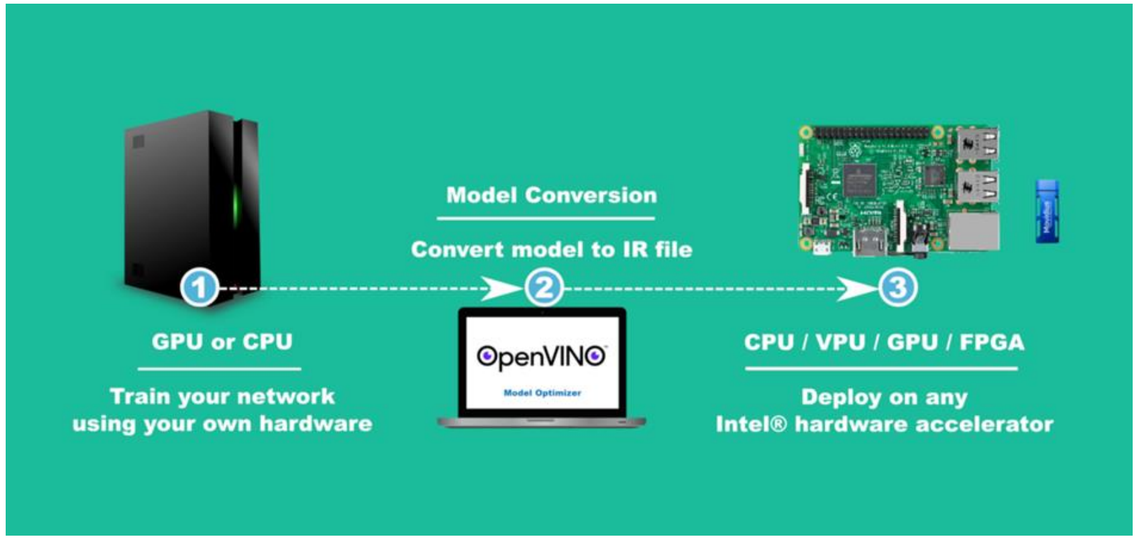
---

Supports multiple softwares (Ubuntu, CentOS, Windows 10...) and multiple algorithms (TensorFlow, Caffe, ApacheMXNet...) via an Open Neural Network Exchange conversion.

Used in our project to achieve end to end learning for self-driving car + traffic sign and pedestrian detection and handling object detection models with deep learning D-CNN (mobilenetv2ssdcoco , mobilenetv2ssdcoco quantized...)



How to connect the NCS2 to the RPi :

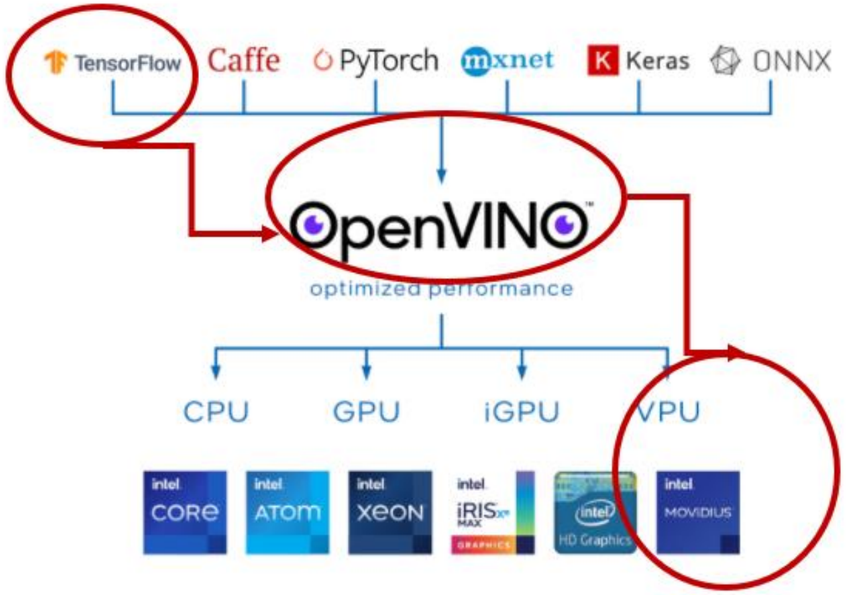


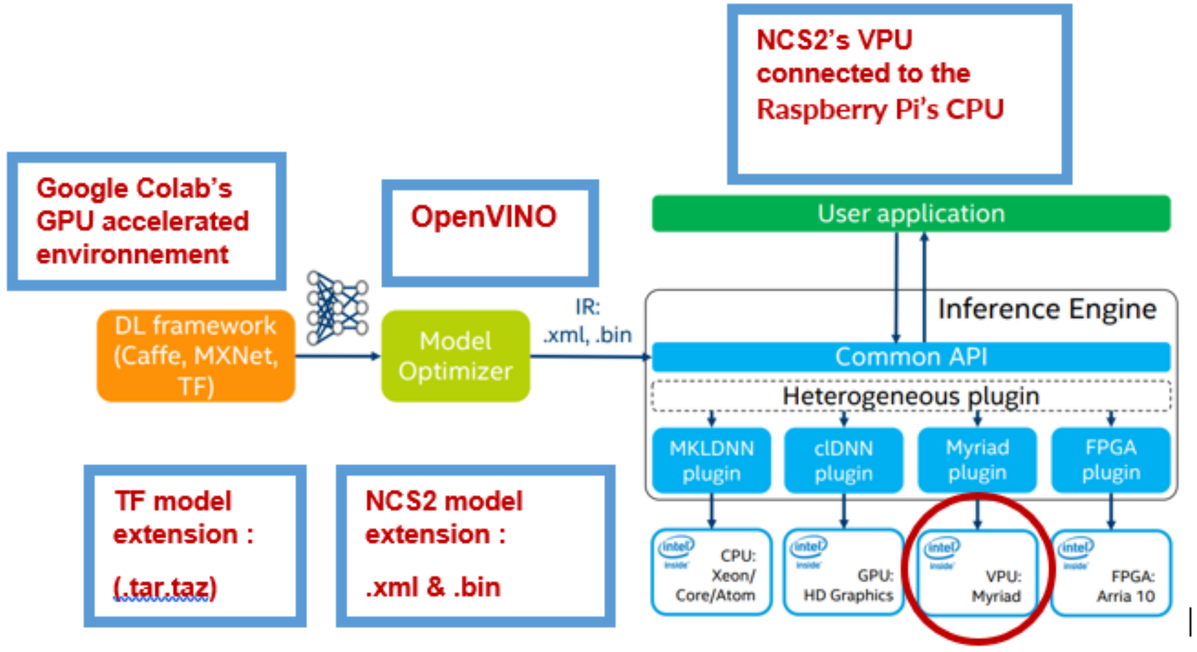
To connect the NCS2 to the RPi, the procedure required is:

1. Convert a TensorFlow model to NCS compatible one, using Open VINO Toolkit by Intel (all of this work is done on your pc's CPU or GPU, for TensorFlow's model we use Google Colab)

2. Install a light version of Open VINO on Raspberry, to run inferences onboard

3. Test and deploy the converted model on Raspberry





SUP'COM
Ecole Supérieure des Communications de Tunis

2.Testing the NCS2

2.1 Demo_Security_Barrier_Camera :

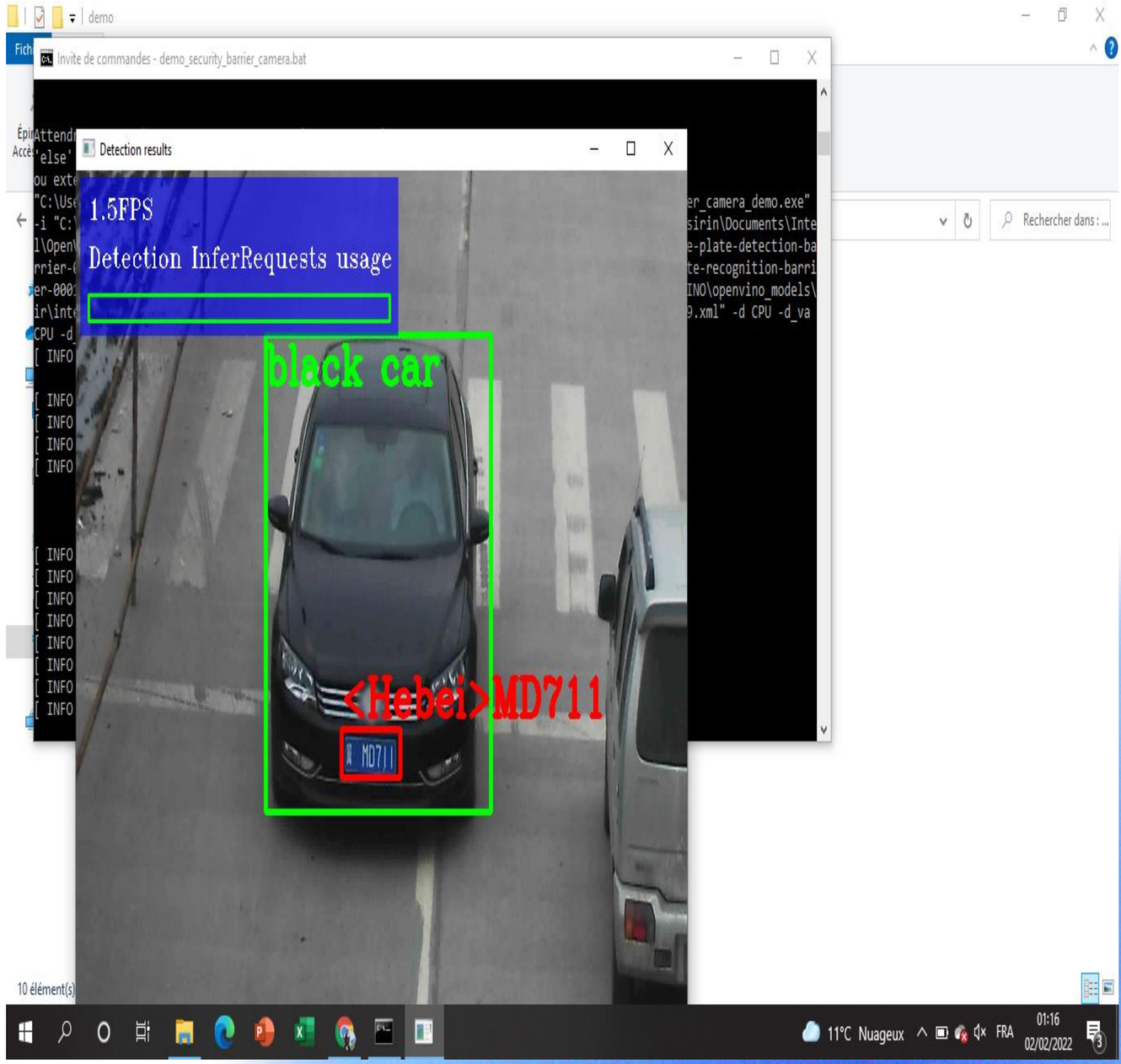
Using the deplyement tools offered by openvino

It uses SqueezeNet for the Computer Vision and Pattern Recognition (object detection : balck car and the licence numbers)

The inference is run in the VPU of the neural compute stick (we must specify the command line -d MYRIAD when we run the Demo)

Software (my host machine Windows 10 with a lite version of intel openvino_2021.4.752 installed and configured,Squeezenet DNN) Hardware(NCS plugged in my PC’s USB port).

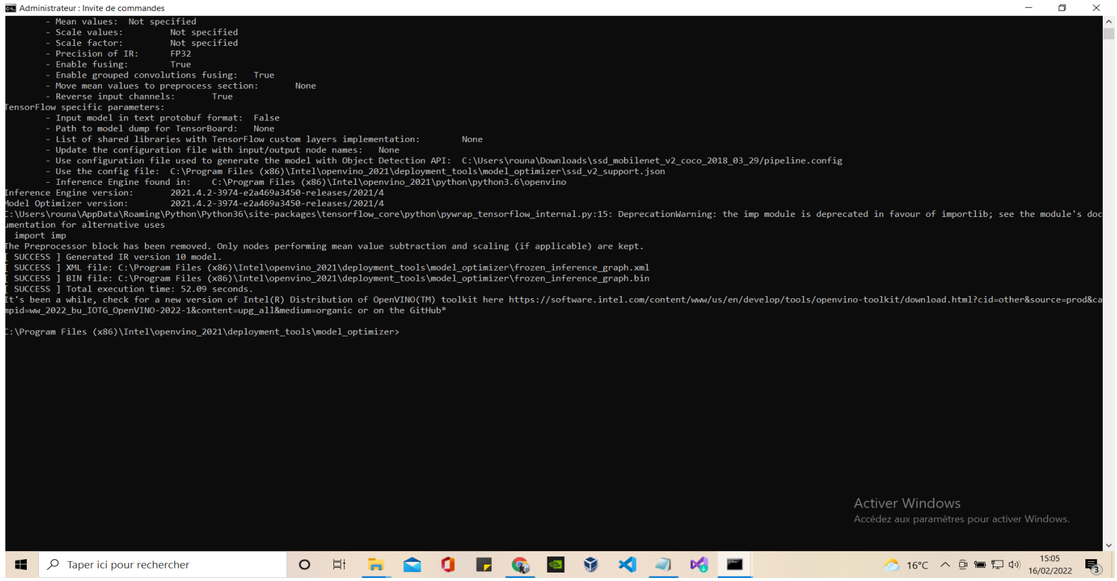
This Demo downloads a SqueezeNet model, uses the Model Optimizer to convert the model to the .bin and .xml Intermediate Representation (IR) files.The Inference Engine requires this model conversion so it can use the IR as input and achieve optimum performance on Intel hardware.



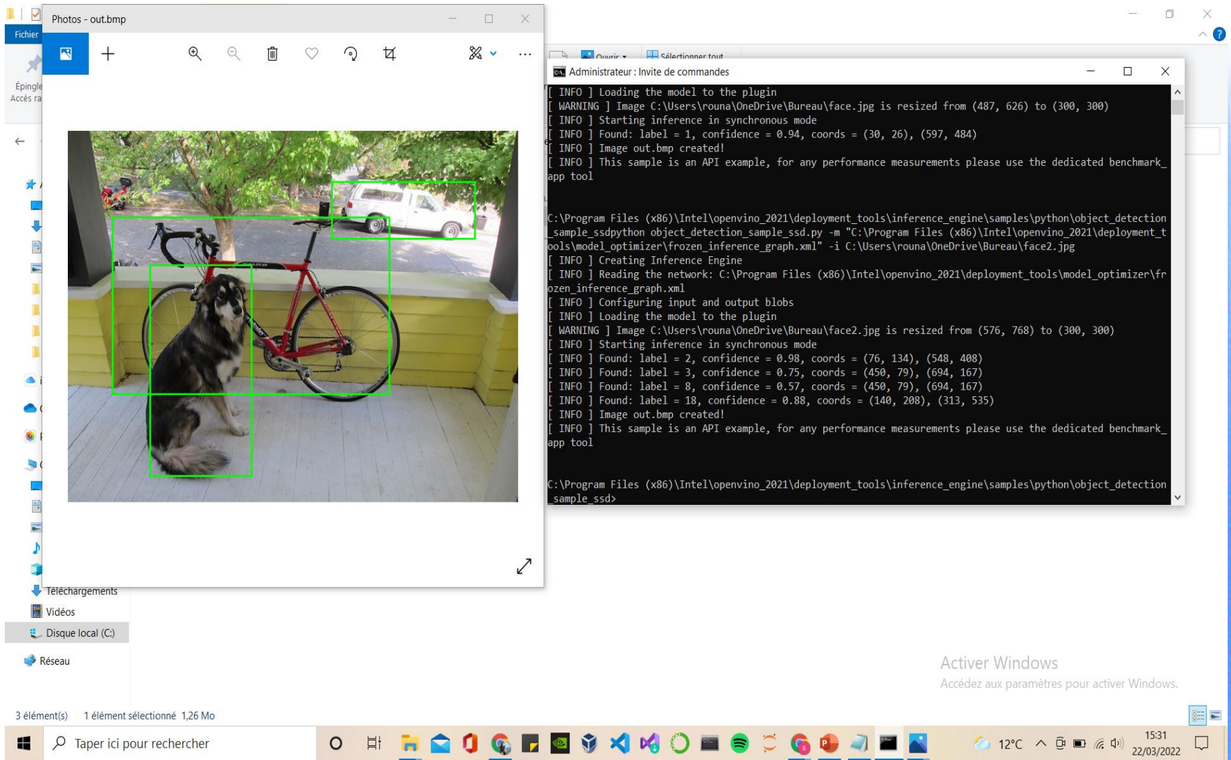
2.2 Demo with ssd_mobilenet_v2_coco_2018_03_29 on images :

First we download the pretrained model ssd_mobilenet_v2_coco from model ZOO

Then we use Openvino's model optimizer to do the IR convertion of the model and generate from it's frozen_inference_grapg.pb both XML file(frozen_inference_grapg.xml) and BIN file (frozen_inference_grapg.bin)



Then we run the XML file using Openvino's inference engine That uses object_detection_sample_ssd.py offred along in the python samples of Openvino



The output of this Demo is the input image with a boundary box showing the face detection.

Software (my host machine Windows 10 with a lite version of intel openvino_2021.4.752 installed and configured,ssd_mobilenet_v2_coco_2012_03_29 DNN) Hardware(NCS plugged in my PC's USB port).

2.3 Demo with ssd_mobilenet_v2_coco_2018_03_29 on videos for real-time object detection :

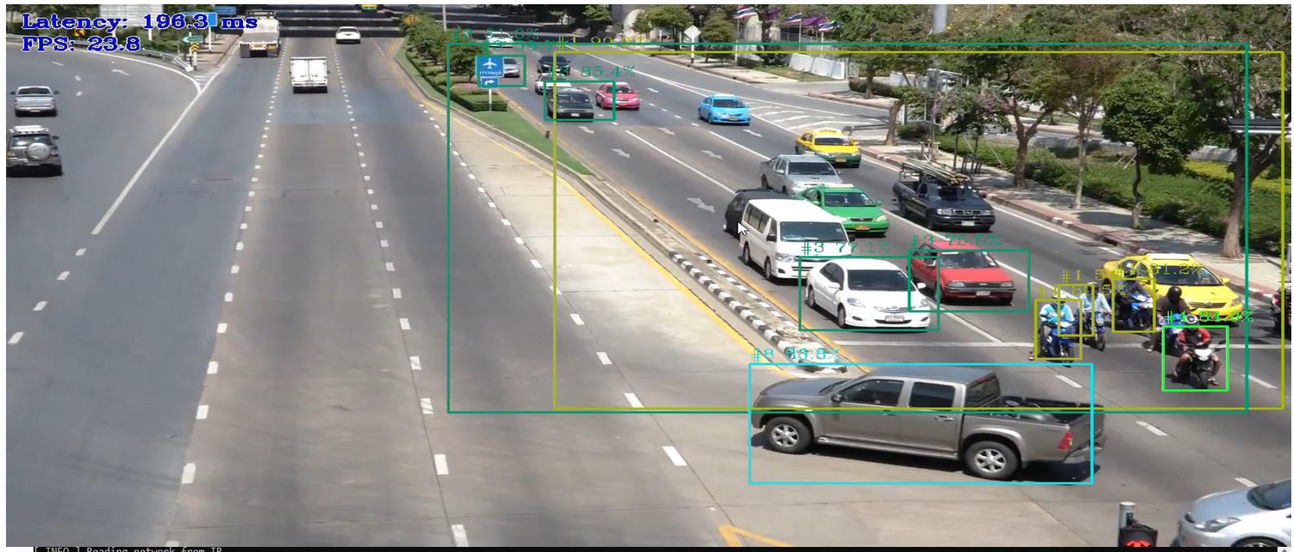
First we download the pretrained model ssd_mobilenet_v2_coco from model ZOO

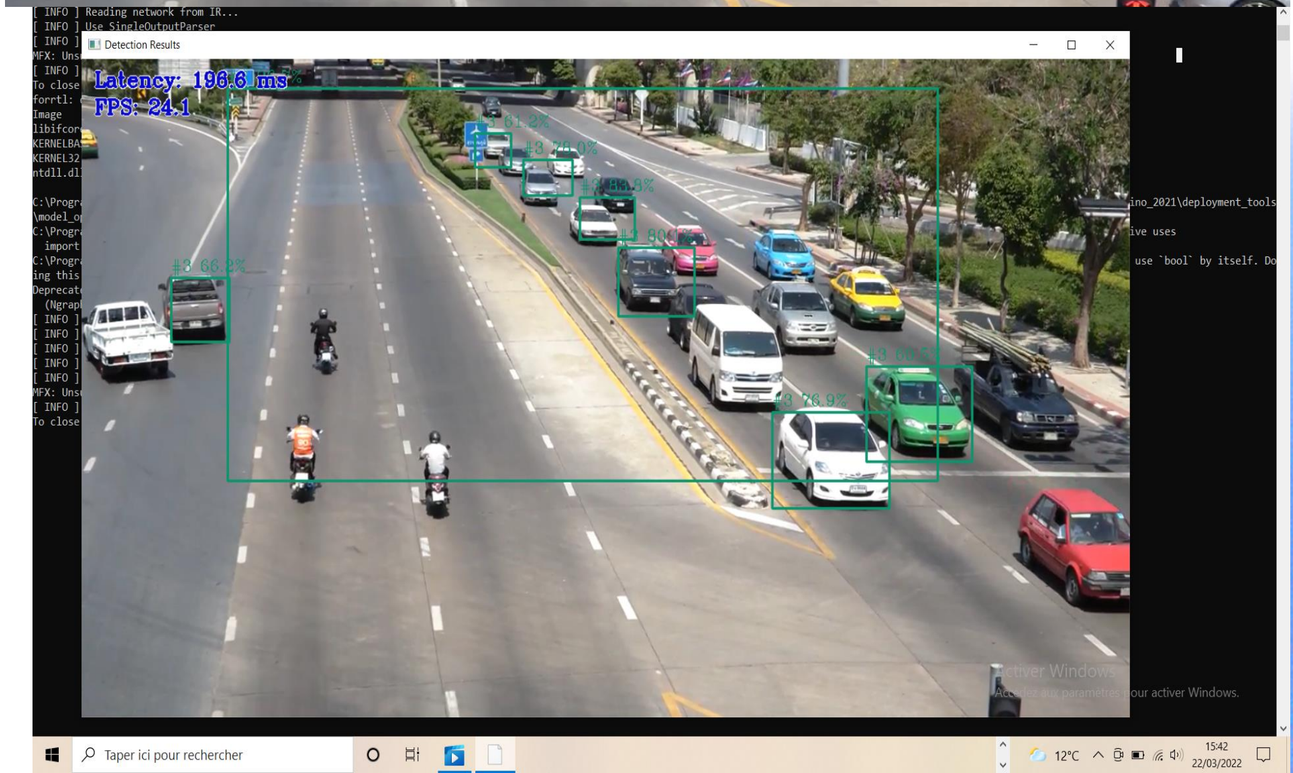
Then we use Openvino's model optimizer to do the IR convertion of the model and generate from it's frozen_inference_grapg.pb both XML file(frozen_inference_grapg.xml) and BIN file (frozen_inference_grapg.bin)

Then we run the XML file using Openvino's inference engine That uses object_detection_demo.py offred along in the open_model_zoo of Openvino's deployement tools.

The output of this Demo is the input video with a real time object detection boundary boxes.

Software (my host machine Windows 10 with a lite version of intel openvino_2021.4.752 installed and configured,ssd_mobilenet_v2_coco_2012_03_29 DNN) Hardware(NCS plugged in my PC's USB port).





P2M PROJECT:

SUP'COM
Ecole Supérieure des Communications de Tunis

3. End-to-End Lane Navigation via Nvidia’s Deep Learning Model (using Colab’s GPUs)

SUP'COM
Ecole Supérieure des Communications de Tunis

4.Next Steps

Convert our Deep Learning model to the .bin and .xml Intermediate Representation (IR) files to run them over the NCS2 and RPI3 :

After generation the deep learning model for autonomous lane navigation (Lane_navigation_finale.h5 and Lane_navigation_check.h5)

We will use intel's **DL Workbench** that converts Keras H5 models to the Saved Model format and then to the OpenVINO™ format (XML and BIN files) with the Model Optimizer.

The inference can then be run in the VPU of the neural compute stick and the Raspbarry Pi3

DL Workbench combines OpenVINO™ tools to assist you with the most commonly used tasks: import a model, analyze its performance and accuracy, visualize the outputs, optimize and prepare the model for deployment in a matter of minutes.