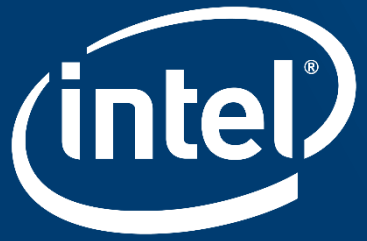


DEPLOYING A DNN MODEL ON LOW-POWER EDGE DEVICES (INTEL® MOVIDIUS™ NEURAL COMPUTE STICK + RASPBERRY PI*)

LEGAL DISCLAIMER

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at intel.com.

This sample source code is released under the Intel Sample Source Code License Agreement.

Intel, the Intel logo, Movidius, Myriad, and OpenVINO are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

AGENDA

Chapter Outcome: Deploy an Object Detection Model on Raspberry Pi*

- Why use a low-powered embedded board
- Compare development mode versus deployment mode of the OpenVINO™ toolkit
- Installing OpenVINO Toolkit on Raspberry Pi
- Running an object detector model on Raspberry Pi

WHY A LOW-POWERED EMBEDDED BOARD?

The Intel® Neural Compute Stick 2 (Intel® NCS2) is a Intel® Movidius™ Myriad™ X Vision Processing Unit (VPU) on a USB stick.

Since the Intel® NCS2 is designed for low-power applications, it makes sense we pair it with a low-power embedded system.

Some options for low-powered systems: Raspberry Pi* (RPi), MinnowBoard* or UP Board.

In this chapter, our focus will be on deploying on **Raspberry Pi*.**

RASPBIAN (OPERATING SYSTEM FOR RPI) BOARD

Raspbian* (operating system for RPi) provides a nice graphical user interface (GUI) so that users can plug in a monitor, keyboard, and mouse to work on the RPi.

While this makes it convenient for users to explore RPi features and do some light development work, it is common to deploy embedded products/projects in headless mode (no monitor, keyboard, or mouse).

For example, if you want to build a RPi-powered ping pong pursuit robot, you can't really put a monitor, keyboard, and mouse on it.

Software for embedded products is typically developed on a development platform, like a laptop, desktop, or server, and the resulting binary files are deployed on the embedded hardware.

DEVELOPMENT VERSUS DEPLOYMENT IN OPENVINO TOOLKIT

The OpenVINO™ toolkit supports both development and deployment of deep learning models

1. **Development Mode—**Installs the complete OpenVINO toolkit package with the Model Optimizer and Inference Engine, with plugins for multiple hardware targets.

   Refer to Chapter 2 for installation.

2. **Deployment on Raspberry Pi*—**Installs only the Inference Engine and Intel® Movidius™ Myriad™ technology plugin on RPi.

   Profiling and compiling networks should be done on a laptop or desktop.

SUPPORTED TARGET PLATFORMS

Hardware

- Raspberry Pi* board with Arm* ARMv7-A CPU architecture. Check that uname -m returns armv7l
- One of Intel® Movidius™ Vision Processing Units (VPU):
  - Intel® Movidius™ Neural Compute Stick
  - Intel® Neural Compute Stick 2

Operating Systems

- Raspbian* Buster, 32 bit
- Raspbian Stretch, 32 bit

Software

- CMake* 3.7.2 or higher
- Python* 3.5, 32 bit

OPENVINO INSTALLATION INSTRUCTIONS FOR RASPBERRY PI*

1. Download the OpenVINO* toolkit for Raspbian* OS to the Downloads directory:

   https://download.01.org/opencv/2019/openvinotoolkit/R3/

2. Open a terminal on Rpi

3. Create an installation folder

   ```
   sudo mkdir -p /opt/intel/openvino
   ```

4. Unpack the archive and install OpenVINO toolkit components

   - `cd ~/Downloads/`
   - `sudo tar -xf  l_openvino_toolkit_runtime_raspbian_p_<version>.tgz --strip 1 -C /opt/intel/openvino`

SET UP THE ENVIRONMENT VARIABLES AND DEPENDENCIES

1. Update environment variables using scripts provided by the OpenVINO™ toolkit package

   ```
   source /opt/intel/openvino/bin/setupvars.sh
   ```

   You will lose the settings when you close the shell

2. (Optional) Permanently set up environment variables

   ```
   echo "source /opt/intel/openvino/bin/setupvars.sh" >> ~/.bashrc
   ```

3. Test environment is set correctly by opening a new terminal. You will see the following message:

   ```
   [setupvars.sh] OpenVINO environment initialized
   ```

4. Install Cmake* to be able to build OpenVINO samples

   ```
   sudo apt install cmake
   ```

ADD USB RULES FOR INTEL® NEURAL COMPUTE STICK 2

1. Add the current Linux* user to the users group

   ```
   sudo usermod -a -G users "$(whoami)"
   ```

   You may be required to log out and log back in.

2. Set up environment variables

   If not permanently set using bashrc, run the following command again:

   ```
   source /opt/intel/openvino/bin/setupvars.sh
   ```

3. Run the install_NCS_udev_rules.sh script to install USB rules. This is **required** to deploy using the Intel Neural Compute Stick 2 (Intel® NCS2)

   Plug in the Intel NCS2

   ```
   sh /opt/intel/openvino/install_dependencies/install_NCS_udev_rules.sh
   ```

DEPLOY PRETRAINED FACE DETECTION EXAMPLE ON RASPBERRY PI* USING INTEL® NEURAL COMPUTE STICK 2

Create a samples build directory

- `mkdir build && cd build`

Download pretrained face detection model (weights file .bin and network file .xml)

- `wget --no-check-certificate https://download.01.org/opencv/2019/open_model_zoo/R1/models_bin/face-detection-adas-0001/FP16/face-detection-adas-0001.bin`
- `wget --no-check-certificate https://download.01.org/opencv/2019/open_model_zoo/R1/models_bin/face-detection-adas-0001/FP16/face-detection-adas-0001.xml`

DEPLOY PRETRAINED FACE DETECTION EXAMPLE ON RASPBERRY PI* USING INTEL® NEURAL COMPUTE STICK 2 (CONTINUED)

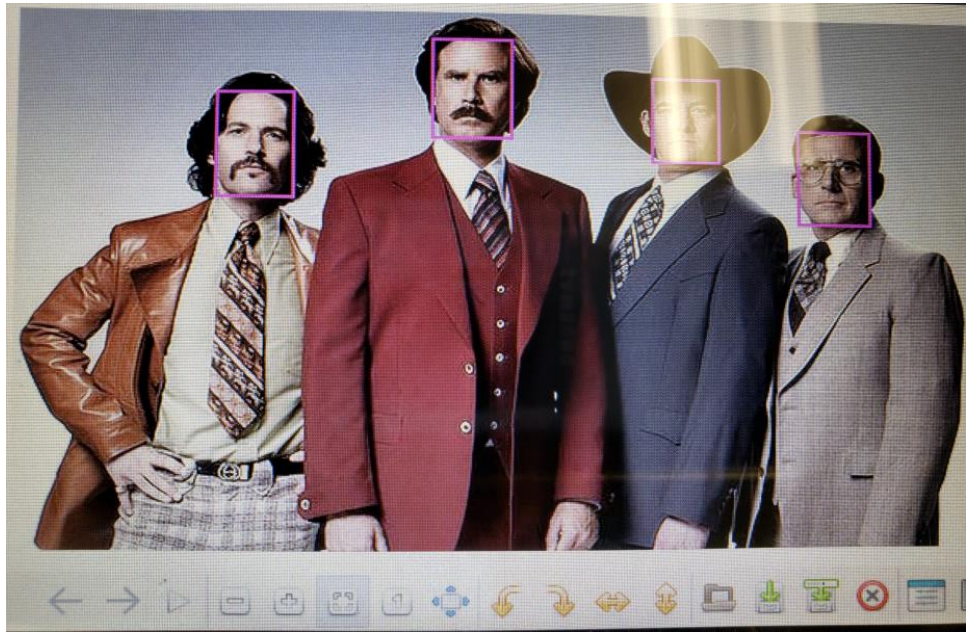
Build the Object Detection sample application. We will use this app to call the Intel® Movidius™ Myriad™ technology plugin

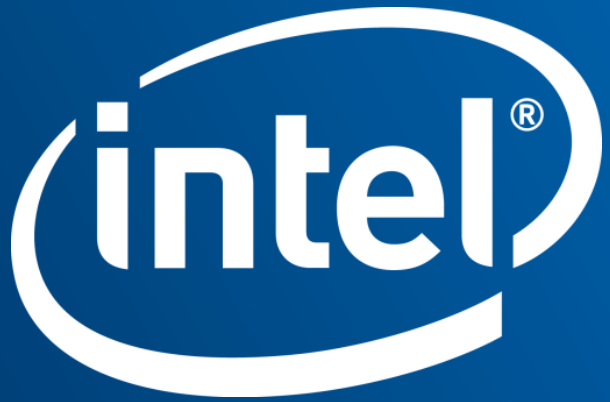
- `cmake -DCMAKE_BUILD_TYPE=Release -DCMAKE_CXX_FLAGS="-march=armv7-a" /opt/intel/openvino/deployment_tools/inference_engine/samples`
- `make -j2 object_detection_sample_ssd`

Run the sample with an example input image and the trained model

- `./armv7l/Release/object_detection_sample_ssd -m face-detection-adas-0001.xml -d MYRIAD -i <path_to_image>`

OUTPUT




intel