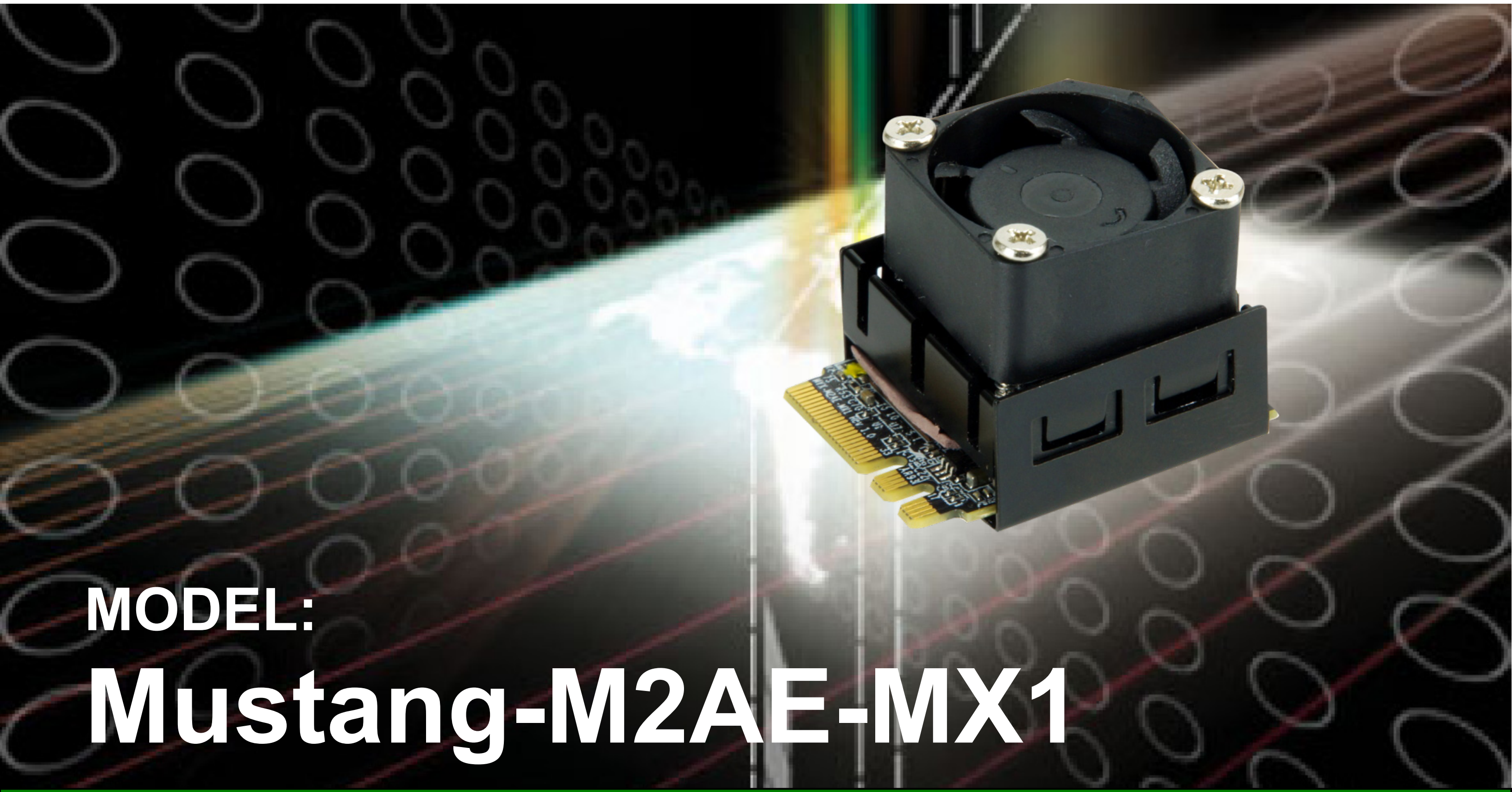
IEI Integration Corp.



MODEL:

# Mustang-M2AE-MX1

## Intel® Vision Accelerator Design with Intel® Movidius™ VPU

**Powered by Open Visual Inference & Neural Network Optimization (OpenVINO™) toolkit**

# User Manual

**Rev. 1.00 - December 9, 2019**

RoHS Lead Free

# Revision

| Date | Version | Changes |
|---|---|---|
| December 9, 2019 | 1.00 | Initial release |

# Copyright

## COPYRIGHT NOTICE

## TRADEMARKS

# Manual Conventions

### WARNING

Warnings appear where overlooked details may cause damage to the equipment or result in personal injury. Warnings should be taken seriously.

### CAUTION

Cautionary messages should be heeded to help reduce the chance of losing data or damaging the product.

### NOTE

These messages inform the reader of essential but non-critical information. These messages should be read carefully as any directions or instructions contained therein can help avoid making mistakes.

# Table of Contents

**1 INTRODUCTION.......................................................................................................... 1**

1.1 INTRODUCTION........................................................................................................ 2

1.2 FEATURES................................................................................................................ 3

1.3 DIMENSIONS............................................................................................................. 3

1.4 TECHNICAL SPECIFICATIONS .................................................................................. 4

1.5 PIN DEFINE................................................................................................................ 4

**2 UNPACKING ............................................................................................................... 6**

2.1 ANTI-STATIC PRECAUTIONS ..................................................................................... 7

2.2 UNPACKING PRECAUTIONS...................................................................................... 7

2.3 PACKING LIST........................................................................................................... 8

**3 HARDWARE INSTALLATION.................................................................................. 9**

3.1 ANTI-STATIC PRECAUTIONS ................................................................................... 10

3.2 INSTALLATION CONSIDERATIONS........................................................................... 10

3.3 HARDWARE INSTALLATION..................................................................................... 12

**4 OPENVINO™ TOOLKIT INSTALLATION (LINUX).......................................... 13**

4.1 OPERATING SYSTEMS ............................................................................................ 14

4.2 ONLINE INSTALLATION GUIDE................................................................................ 14

4.3 INSTALL THE INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT CORE COMPONENTS
............................................................................................................................... 15

4.4 INSTALL EXTERNAL SOFTWARE DEPENDENCIES...................................................... 18

4.5 SET THE ENVIRONMENT VARIABLES........................................................................ 19

4.6 CONFIGURE THE MODEL OPTIMIZER ...................................................................... 20

*4.6.1 Model Optimizer Configuration Steps ........................................................... 20*

4.7 RUN THE VERIFICATION SCRIPTS TO VERIFY INSTALLATION................................... 22

4.8 INSTALL MUSTANG-M2AE-MX1 <INTEL HDDL> DRIVER ......................................... 24

4.9 STEPS FOR VERIFY INTEL® VISION ACCELERATOR DESIGN WITH
MUSTANG-M2AE-MX1 INTEL® MOVIDIUS™ VPU ..................................................... 25

**5 OPENVINO™ TOOLKIT INSTALLATION (WINDOWS 10)............................. 26**

5.1 SYSTEM REQUIREMENT ........................................................................................ 27
5.2 ONLINE INSTALLATION GUIDE ............................................................................. 27
5.3 INSTALLATION STEPS .......................................................................................... 28
*5.3.1 Install the Intel® Distribution of OpenVINO™ toolkit Core Components ..... 28*
*5.3.2 Set the Environment Variables ........................................................................ 32*
5.4 CONFIGURE THE MODEL OPTIMIZER ................................................................... 32
*5.4.1 Model Optimizer Configuration Steps ............................................................ 33*
5.4.1.1 Option 1: Configure the Model Optimizer for all supported frameworks at the same time: ........................................................................................ 34
5.4.1.2 Option 2: Configure the Model Optimizer for each framework separately: ........................................................................................................ 34
5.5 USE VERIFICATION SCRIPTS TO VERIFY YOUR INSTALLATION .................................. 36
*5.5.1 Run the Image Classification Verification Script............................................ 37*
*5.5.2 Run the Inference Pipeline Verification Script................................................ 38*
*5.5.3 Installation Steps for iEi Mustang-M2AE-MX1 the Intel® Vision Accelerator Design with Intel® Movidius™ VPU ....................................................... 40*
**A PERFORMANCE TEST ........................................................................................ 42**
**B REGULATORY COMPLIANCE ............................................................................. 44**
**C PRODUCT DISPOSAL ........................................................................................ 46**
**D HAZARDOUS MATERIALS DISCLOSURE ............................................................ 48**

# List of Figures

Figure 1-1: Mustang-M2AE-MX1 ........................................................................................ 2
Figure 1-2: Dimensions (mm) ............................................................................................. 3

Chapter

1

# Introduction

## 1.1 Introduction

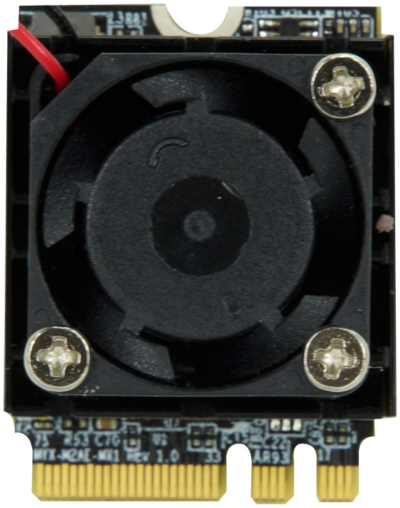


**Figure 1-1: Mustang-M2AE-MX1**

The Mustang-M2AE-MX1 is a deep learning convolutional neural network acceleration card for speeding up AI inference, in a flexible and scalable way. Equipped with Intel® Movidius™ Myriad™ X Vision Processing Unit (VPU), the Mustang-M2AE-MX1 M.2 card can be used with the existing system, enabling high-performance computing without costing a fortune. VPUs can run AI faster, and is well suited for low power consumption applications such as surveillance, retail and transportation. With the advantage of power efficiency and high performance to dedicate DNN topologies, it is perfect to be implemented in AI edge computing device to reduce total power usage, providing longer duty time for the rechargeable edge computing equipment.

"Open Visual Inference & Neural Network Optimization (OpenVINO™) toolkit" is based on convolutional neural networks (CNN), the toolkit extends workloads across Intel® hardware and maximizes performance. It can optimize pre-trained deep learning model such as Caffe, MXNET, Tensorflow into IR binary file then execute the inference engine across Intel®-hardware heterogeneously such as CPU, GPU, Intel® Movidius™ Myriad X VPU, and FPGA.

## 1.2 Features

Mustang-M2AE-MX1 features are listed below:

- One Intel® Movidius™ Myriad™ X Vision Processing Unit
- Interface: M.2 A+E Key
- Dimensions: 22 mm x 30 mm
- Cooling: Active fan
- OpenVINO™ Toolkit, AI edge computing ready device
- Low power consumption, approximate 4.5W.

## 1.3 Dimensions

The dimensions of the board are listed below:

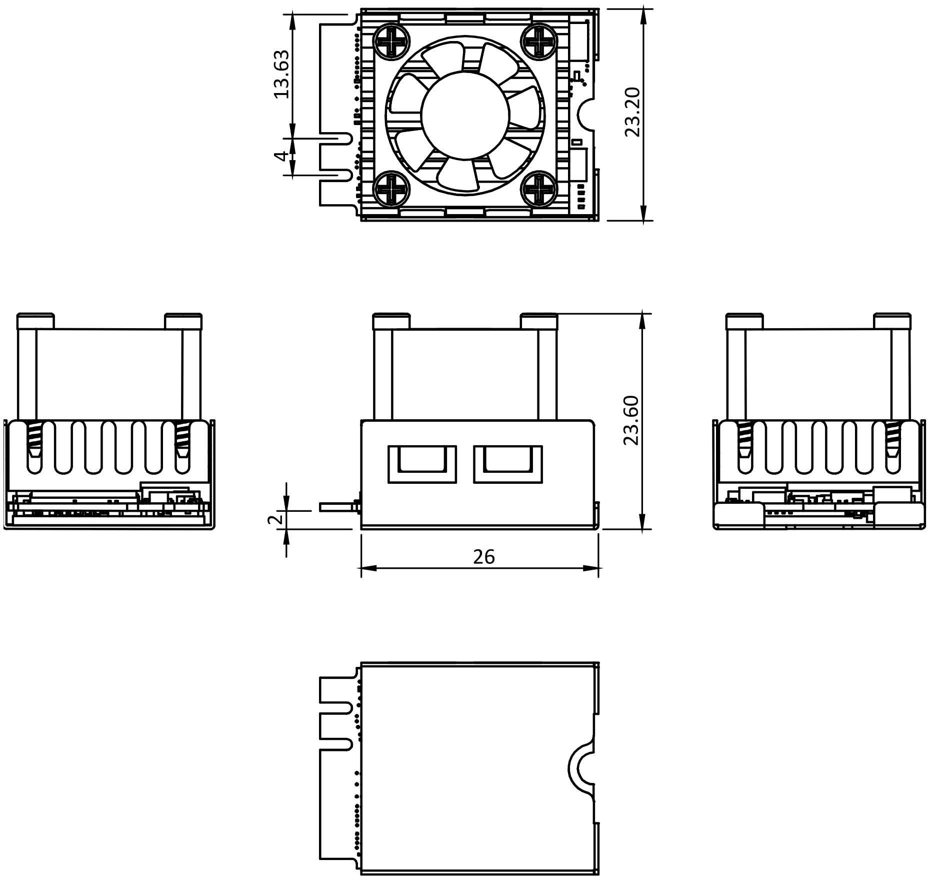


**Figure 1-2: Dimensions (mm)**

## 1.4 Technical Specifications

Mustang-M2AE-MX1 technical specifications are listed below.

| Specification | Mustang-M2AE-MX1 |
|---|---|
| Main Chip | One Intel® Movidius™ Myriad™ X MA2485 VPU |
| Operating Systems | Ubuntu 18.04 (LTS), 64-bit<br>Ubuntu 16.04 (LTS), 64-bit<br>CentOS 64-bit<br>Windows10 64-bit |
| Interface | M.2 A+E Key |
| Fan | Active fan |
| Power Consumption | Approximate 4.5W |
| Operating Temperature | -20°C~60°C |
| Storage Temperature | 5% ~ 90% |
| Dimensions | 22 mm x 30 mm |

**Table 1-1: Technical Specifications**

## 1.5 Pin Define

| Pin | Description | Pin | Description |
|---|---|---|---|
| 1 | GND | 2 | 3.3V |
| 3 | USB_D+ | 4 | 3.3V |
| 5 | USB_D- | 6 | N/C |
| 7 | GND | 8 | Module Key |
| 9 | Module Key | 10 | Module Key |
| 11 | Module Key | 12 | Module Key |
| 13 | Module Key | 14 | Module Key |
| 15 | Module Key | 16 | N/C |
| 17 | N/C | 18 | GND |
| 19 | N/C | 20 | N/C |

| | | | |
|---|---|---|---|
| 21 | N/C | 22 | N/C |
| 23 | N/C | 24 | Module Key |
| 25 | Module Key | 26 | Module Key |
| 27 | Module Key | 28 | Module Key |
| 29 | Module Key | 30 | Module Key |
| 31 | Module Key | 32 | N/C |
| 33 | GND | 34 | N/C |
| 35 | PERp0 | 36 | N/C |
| 37 | PERn0 | 38 | N/C |
| 39 | GND | 40 | N/C |
| 41 | PETp0 | 42 | N/C |
| 43 | PETn0 | 44 | N/C |
| 45 | GND | 46 | N/C |
| 47 | REFCLKp0 | 48 | N/C |
| 49 | REFCLKn0 | 50 | N/C |
| 51 | GND | 52 | PERST0#(I)(0/3.3V) |
| 53 | CLKREQ0#(I/O)(0/3.3V) | 54 | N/C |
| 55 | PEWAKE0#(I/O)(0/3.3V) | 56 | N/C |
| 57 | GND | 58 | N/C |
| 59 | N/C | 60 | N/C |
| 61 | N/C | 62 | N/C |
| 63 | GND | 64 | N/C |
| 65 | N/C | 66 | N/C |
| 67 | N/C | 68 | N/C |
| 69 | GND | 70 | N/C |
| 71 | N/C | 72 | 3.3V |
| 73 | N/C | 74 | 3.3V |
| 75 | GND | | |

# Chapter 2

# Unpacking

## 2.1 Anti-static Precautions

⚠ **WARNING!**

Static electricity can destroy certain electronics. Make sure to follow the ESD precautions to prevent damage to the product, and injury to the user.

Make sure to adhere to the following guidelines:

- ***Wear an anti-static wristband***: Wearing an anti-static wristband can prevent electrostatic discharge.
- ***Self-grounding***: Touch a grounded conductor every few minutes to discharge any excess static buildup.
- ***Use an anti-static pad***: When configuring any circuit board, place it on an anti-static mat.
- ***Only handle the edges of the PCB***: Don't touch the surface of the motherboard. Hold the motherboard by the edges when handling.

## 2.2 Unpacking Precautions

When the Mustang-M2AE-MX1 is unpacked, please do the following:

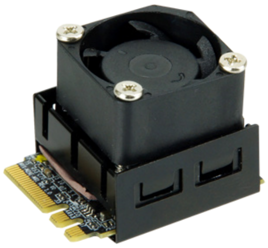
- Follow the antistatic guidelines above.
- Make sure the packing box is facing upwards when opening.
- Make sure all the packing list items are present.

## 2.3 Packing List

**NOTE:**

If any of the components listed in the checklist below are missing, do not proceed with the installation. Contact the IEI reseller or vendor the Mustang-M2AE-MX1 was purchased from or contact an IEI sales representative directly by sending an email to sales@ieiworld.com.

The Mustang-M2AE-MX1 is shipped with the following component:

| Quantity | Item and Part Number | Image |
|---|---|---|
| 1 | Mustang-M2AE-MX1 AI acceleration card |  |

Chapter

3

# Hardware Installation

## 3.1 Anti-static Precautions

**WARNING:**

Failure to take ESD precautions during the installation of the Mustang-M2AE-MX1 may result in permanent damage to the Mustang-M2AE-MX1 and severe injury to the user.

Electrostatic discharge (ESD) can cause serious damage to electronic components, including the Mustang-M2AE-MX1. Dry climates are especially susceptible to ESD. It is therefore critical that whenever the Mustang-M2AE-MX1 or any other electrical component is handled, the following anti-static precautions are strictly adhered to.

- ***Wear an anti-static wristband***: Wearing a simple anti-static wristband can help to prevent ESD from damaging the board.
- ***Self-grounding*** Before handling the board, touch any grounded conducting material. During the time the board is handled, frequently touch any conducting materials that are connected to the ground.
- ***Use an anti-static pad***: When configuring the Mustang-M2AE-MX1, place it on an anti-static pad. This reduces the possibility of ESD damaging the Mustang-M2AE-MX1.
- ***Only handle the edges of the PCB***: When handling the PCB, hold the PCB by the edges.

## 3.2 Installation Considerations

**NOTE:**

The following installation notices and installation considerations should be read and understood before installation. All installation notices must be strictly adhered to. Failing to adhere to these precautions may lead to severe damage and injury to the person performing the installation.

**WARNING:**

The installation instructions described in this manual should be carefully followed in order to prevent damage to the Mustang-M2AE-MX1, its components and injury to the user.

Before and during the installation please **DO** the following:

- Read the user manual:
  The user manual provides a complete description of the Mustang-M2AE-MX1 installation instructions and configuration options.
- Wear an electrostatic discharge cuff (ESD):
  Electronic components are easily damaged by ESD. Wearing an ESD cuff removes ESD from the body and helps prevent ESD damage.
- Turn off system:
  When installing the Mustang-M2AE-MX1, make sure that the system to be connected is disconnected from all power supplies and that no electricity is being fed into the system.

Before and during the installation of the Mustang-M2AE-MX1 **DO NOT:**

- Remove any of the stickers on the PCB board. These stickers are required for warranty validation.
- Use the product before verifying all the cables and power connectors are properly connected.
- Allow screws to come in contact with the PCB circuit, connector pins, or its components.

## 3.3 Hardware Installation

To install the Mustang-M2AE-MX1, please follow the steps below.

**Step 1:** **Prepare the computer.** Turn off the computer, and remove the power cord from the rear of the power supply.

**WARNING:**

Disconnect the computer from the power supply and from any networks to which you will install the Mustang-M2AE-MX1, or you risk damaging the system or experiencing electrical shock.

**Step 2:** **Remove the cover from the chassis.**

**Step 3:** **Locate available M.2 2280 slot.** The Mustang-M2AE-MX1 is compatible with M.2 2280 A+E key slot.

**Step 4:** **Insert into the socket.** Line up the notch on the module with the notch on the slot. Slide the Mustang-M2AE-MX1 into the socket at an angle of about 20º.

**Step 5:** **Secure the Mustang-M2AE-MX1** with a retention screw.

**Step 6:** **Re-install the cover of the chassis**.

**Step 7:** **Reconnect any power cords and any network cables to the system.** Power up the system.

Chapter

4

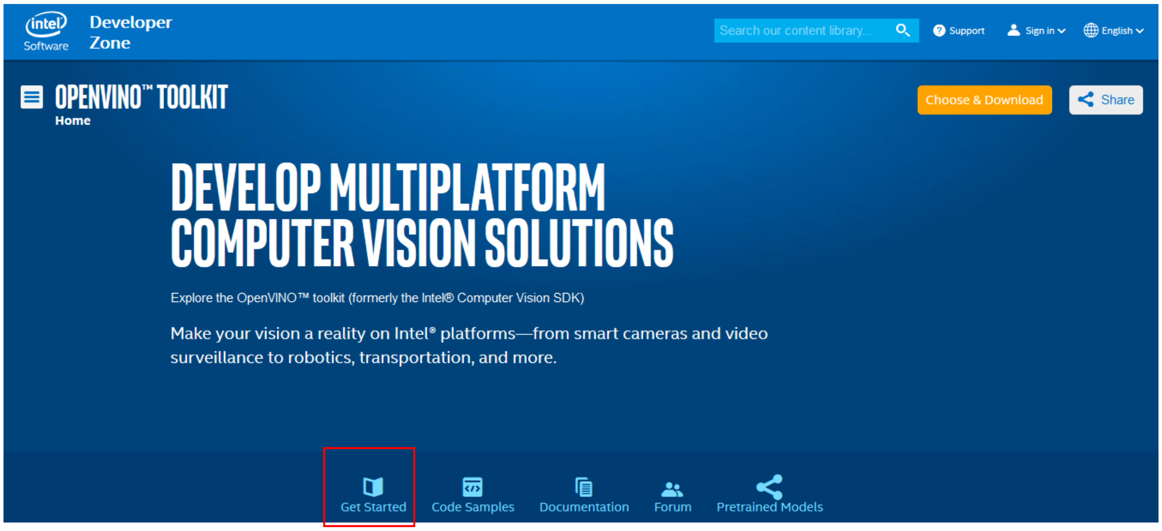
# OpenVINO™ Toolkit Installation (Linux)

## 4.1 Operating Systems

- Ubuntu 18.04 LTS 64-bit
- Ubuntu 16.04 LTS 64-bit
- CentOS 64-bit

## 4.2 Online Installation Guide

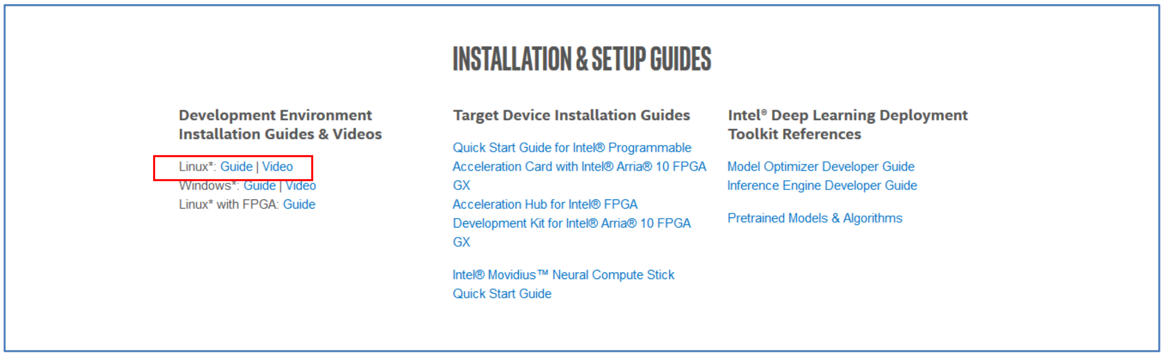
Go to https://software.intel.com/en-us/openvino-toolkit. Click "Get Started" then choose your configuration from "Development Environment Installation Guides & Videos". Follow the instruction to complete the installation procedure.



DEVELOP MULTIPLATFORM COMPUTER VISION SOLUTIONS

Explore the OpenVINO™ toolkit (formerly the Intel® Computer Vision SDK)

Make your vision a reality on Intel® platforms—from smart cameras and video surveillance to robotics, transportation, and more.



INSTALLATION & SETUP GUIDES

**Development Environment Installation Guides & Videos**

Linux*: Guide | Video
Windows*: Guide | Video
Linux* with FPGA: Guide

**Target Device Installation Guides**

Quick Start Guide for Intel® Programmable Acceleration Card with Intel® Arria® 10 FPGA GX
Acceleration Hub for Intel® FPGA Development Kit for Intel® Arria® 10 FPGA GX

Intel® Movidius™ Neural Compute Stick Quick Start Guide

**Intel® Deep Learning Deployment Toolkit References**

Model Optimizer Developer Guide
Inference Engine Developer Guide

Pretrained Models & Algorithms

## 4.3 Install the Intel® Distribution of OpenVINO™ Toolkit Core Components

Download the Intel® Distribution of OpenVINO™ toolkit package file from Intel® Distribution of OpenVINO™ toolkit for Linux*. Select the Intel® Distribution of OpenVINO™ toolkit for Linux package from the dropdown menu.

1. Open a command prompt terminal window.
2. Change directories to where you downloaded the Intel Distribution of OpenVINO toolkit for Linux* package file.
   If you downloaded the package file to the current user's `Downloads` directory:

   ```
   cd ~/Downloads/
   ```

   By default, the file is saved as `l_openvino_toolkit_p_<version>.tgz`.
3. Unpack the .tgz file:

   ```
   tar -xvzf l_openvino_toolkit_p_<version>.tgz
   ```

   The files are unpacked to the `l_openvino_toolkit_p_<version>` directory.
4. Go to the `l_openvino_toolkit_p_<version>` directory:

   ```
   cd l_openvino_toolkit_p_<version>
   ```

   If you have a previous version of the Intel Distribution of OpenVINO toolkit installed, rename or delete these two directories:

   - `/home/<user>/inference_engine_samples_build`
   - `/home/<user>/openvino_models`

**Installation Notes:**

- Choose an installation option and run the related script as root.
- You can use either a GUI installation wizard or command-line instructions (CLI).

- Screenshots are provided for the GUI, but not for CLI. The following information also applies to CLI and will be helpful to your installation where you will be presented with the same choices and tasks.

1. Choose your installation option:

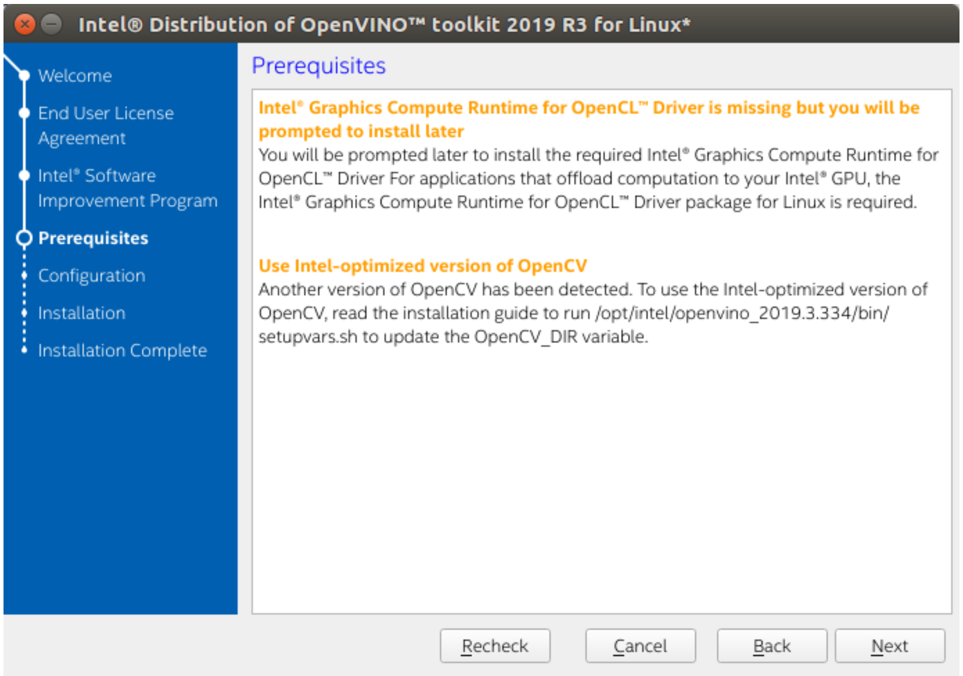
    - **Option 1:** GUI Installation Wizard:

```
sudo ./install_GUI.sh
```

    - **Option 2:** Command-Line Instructions:

```
sudo ./install.sh
```

2. Follow the instructions on your screen. Watch for informational messages such as the following in case you must complete additional steps:


Intel® Distribution of OpenVINO™ toolkit 2019 R3 for Linux*
Welcome
End User License Agreement
Intel® Software Improvement Program
Prerequisites
Configuration
Installation
Installation Complete
Prerequisites
Recheck
Cancel
Back
Next

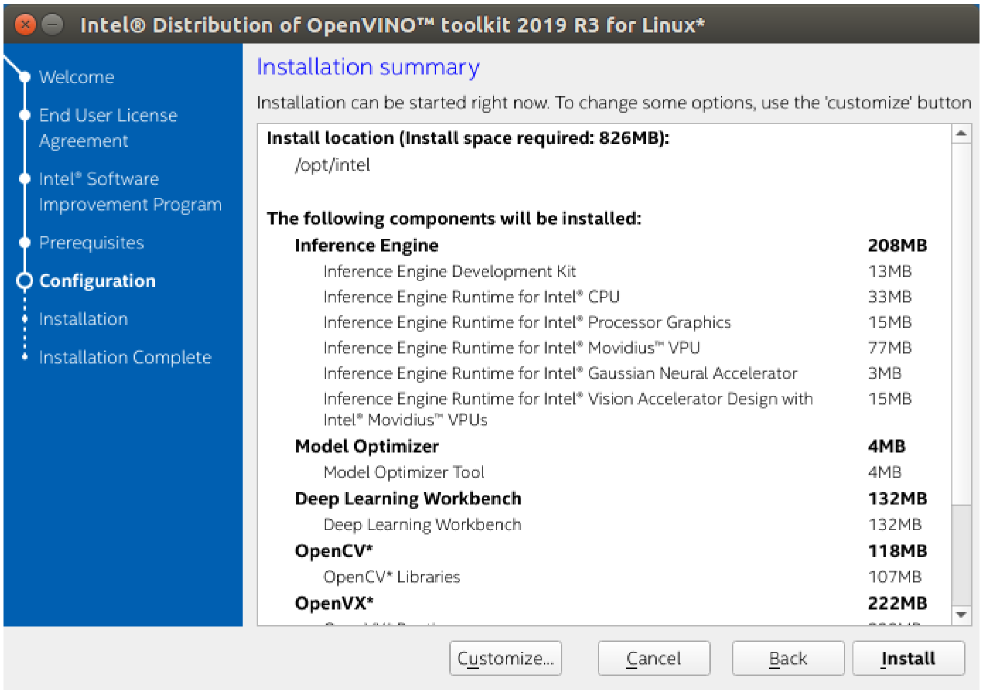
Intel® Graphics Compute Runtime for OpenCL™ Driver is missing but you will be prompted to install later

You will be prompted later to install the required Intel® Graphics Compute Runtime for OpenCL™ Driver For applications that offload computation to your Intel® GPU, the Intel® Graphics Compute Runtime for OpenCL™ Driver package for Linux is required.

Use Intel-optimized version of OpenCV

Another version of OpenCV has been detected. To use the Intel-optimized version of OpenCV, read the installation guide to run /opt/intel/openvino_2019.3.334/bin/setupvars.sh to update the OpenCV_DIR variable.

3. If you select the default options, the **Installation summary** GUI screen looks like this:


Intel® Distribution of OpenVINO™ toolkit 2019 R3 for Linux*
Welcome
End User License Agreement
Intel® Software Improvement Program
Prerequisites
Configuration
Installation
Installation Complete
Customize...
Cancel
Back
Install

Installation summary

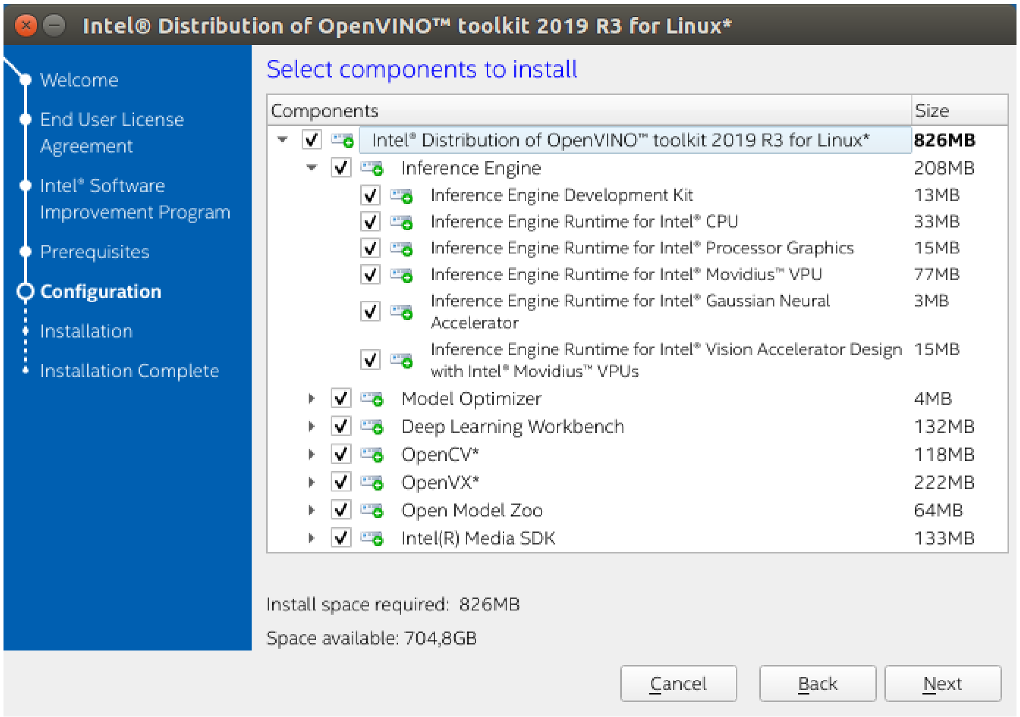
Installation can be started right now. To change some options, use the 'customize' button

**Install location (Install space required: 826MB):**
/opt/intel

**The following components will be installed:**

| | |
|---|---|
| **Inference Engine** | **208MB** |
| Inference Engine Development Kit | 13MB |
| Inference Engine Runtime for Intel® CPU | 33MB |
| Inference Engine Runtime for Intel® Processor Graphics | 15MB |
| Inference Engine Runtime for Intel® Movidius™ VPU | 77MB |
| Inference Engine Runtime for Intel® Gaussian Neural Accelerator | 3MB |
| Inference Engine Runtime for Intel® Vision Accelerator Design with Intel® Movidius™ VPUs | 15MB |
| **Model Optimizer** | **4MB** |
| Model Optimizer Tool | 4MB |
| **Deep Learning Workbench** | **132MB** |
| Deep Learning Workbench | 132MB |
| **OpenCV*** | **118MB** |
| OpenCV* Libraries | 107MB |
| **OpenVX*** | **222MB** |

**Optional:** You can choose **Customize** to change the installation directory or the components you want to install:


Intel® Distribution of OpenVINO™ toolkit 2019 R3 for Linux*
Welcome
End User License Agreement
Intel® Software Improvement Program
Prerequisites
Configuration
Installation
Installation Complete
Cancel
Back
Next

Select components to install

| Components | Size |
|---|---|
| ☑ Intel® Distribution of OpenVINO™ toolkit 2019 R3 for Linux* | **826MB** |
| ☑ Inference Engine | 208MB |
| ☑ Inference Engine Development Kit | 13MB |
| ☑ Inference Engine Runtime for Intel® CPU | 33MB |
| ☑ Inference Engine Runtime for Intel® Processor Graphics | 15MB |
| ☑ Inference Engine Runtime for Intel® Movidius™ VPU | 77MB |
| ☑ Inference Engine Runtime for Intel® Gaussian Neural Accelerator | 3MB |
| ☑ Inference Engine Runtime for Intel® Vision Accelerator Design with Intel® Movidius™ VPUs | 15MB |
| ☑ Model Optimizer | 4MB |
| ☑ Deep Learning Workbench | 132MB |
| ☑ OpenCV* | 118MB |
| ☑ OpenVX* | 222MB |
| ☑ Open Model Zoo | 64MB |
| ☑ Intel(R) Media SDK | 133MB |

Install space required: 826MB

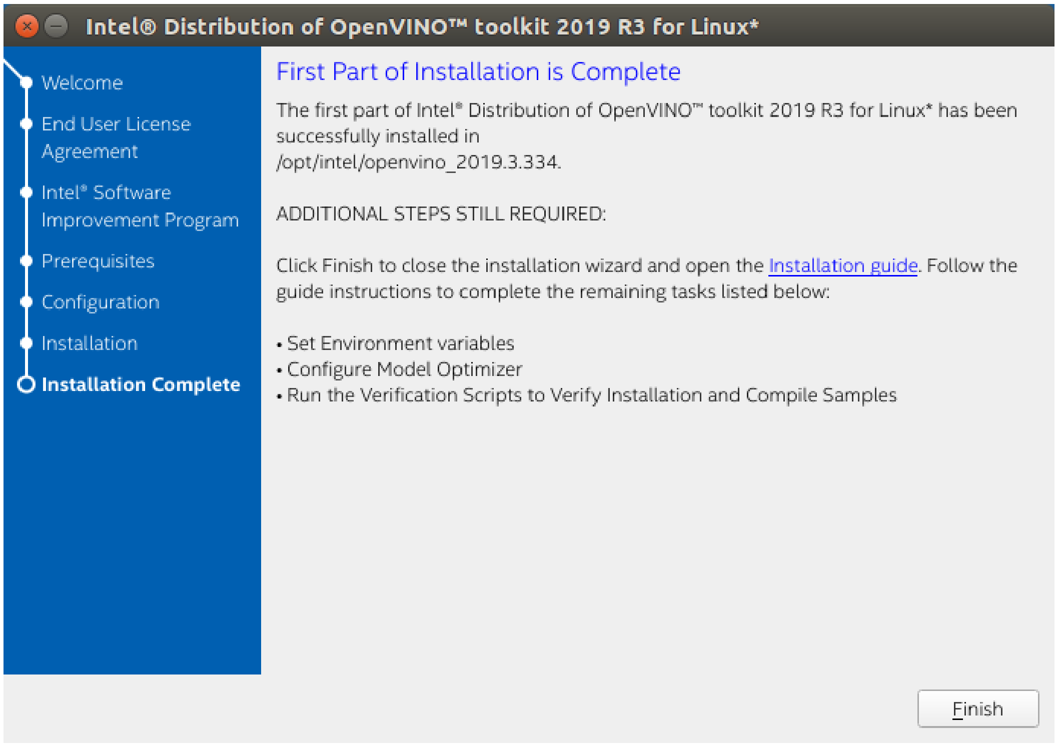
Space available: 704,8GB

When installed as **root** the default installation directory for the Intel Distribution of OpenVINO is `/opt/intel/openvino_<version>/`.

For simplicity, a symbolic link to the latest installation is also created: `/opt/intel/openvino/`.

**NOTE**: The Intel® Media SDK component is always installed in the `/opt/intel/mediasdk` directory regardless of the OpenVINO installation path chosen.

4. A Complete screen indicates that the core components have been install.



Intel® Distribution of OpenVINO™ toolkit 2019 R3 for Linux*
Welcome
End User License Agreement
Intel® Software Improvement Program
Prerequisites
Configuration
Installation
Installation Complete
Finish

First Part of Installation is Complete

The first part of Intel® Distribution of OpenVINO™ toolkit 2019 R3 for Linux* has been successfully installed in
/opt/intel/openvino_2019.3.334.

ADDITIONAL STEPS STILL REQUIRED:

Click Finish to close the installation wizard and open the Installation guide. Follow the guide instructions to complete the remaining tasks listed below:

- Set Environment variables
- Configure Model Optimizer
- Run the Verification Scripts to Verify Installation and Compile Samples

The first core components are installed. Continue to the next section to install additional dependencies.

## 4.4 Install External Software Dependencies

**NOTE**: If you installed the Intel® Distribution of OpenVINO™ to the non-default install directory, replace `/opt/intel` with the directory in which you installed the software.

These dependencies are required for:

- Intel-optimized build of OpenCV library
- Deep Learning Inference Engine
- Deep Learning Model Optimizer tools

1. Change to the `install_dependencies` directory:

```
cd /opt/intel/openvino/install_dependencies
```

2. Run a script to download and install the external software dependencies:

```
sudo -E ./install_openvino_dependencies.sh
```

The dependencies are installed. Continue to the next section to set your environment variables.

## 4.5 Set the Environment Variables

You must update several environment variables before you can compile and run OpenVINO™ applications. Run the following script to temporarily set your environment variables:

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

**Optional:** The OpenVINO environment variables are removed when you close the shell. As an option, you can permanently set the environment variables as follows:

1. Open the `.bashrc` file in `<user_directory>`:

```
vi <user_directory>/.bashrc
```

2. Add this line to the end of the file:

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

3. Save and close the file: press the **Esc** key and type `:wq`.

4. To test your change, open a new terminal. You will see `[setupvars.sh] OpenVINO environment initialized`.

The environment variables are set. Continue to the next section to configure the Model Optimizer.

## 4.6 Configure the Model Optimizer

The Model Optimizer is a Python*-based command line tool for importing trained models from popular deep learning frameworks such as Caffe*, TensorFlow*, Apache MXNet*, ONNX* and Kaldi*.

The Model Optimizer is a key component of the Intel Distribution of OpenVINO toolkit. You cannot perform inference on your trained model without running the model through the Model Optimizer. When you run a pre-trained model through the Model Optimizer, your output is an Intermediate Representation (IR) of the network. The Intermediate Representation is a pair of files that describe the whole model:

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

For more information about the Model Optimizer, refer to the Model Optimizer Developer Guide.

### 4.6.1 Model Optimizer Configuration Steps

You can choose to either configure all supported frameworks at once **OR** configure one framework at a time. Choose the option that best suits your needs. If you see error messages, make sure you installed all dependencies.

**NOTE**: Since the TensorFlow framework is not officially supported on CentOS*, the Model Optimizer for TensorFlow can't be configured and ran on those systems.

**IMPORTANT**: The Internet access is required to execute the following steps successfully. If you have access to the Internet through the proxy server only, please make sure that it is configured in your OS environment.

**Option 1: Configure all supported frameworks at the same time**

1. Go to the Model Optimizer prerequisites directory:

```
cd /opt/intel/openvino/deployment_tools/model_optimizer/install_prerequisites
```

2. Run the script to configure the Model Optimizer for Caffe, TensorFlow, MXNet, Kaldi*, and ONNX:

```
sudo ./install_prerequisites.sh
```

**Option 2: Configure each framework separately**

Configure individual frameworks separately **ONLY** if you did not select **Option 1** above.

1. Go to the Model Optimizer prerequisites directory:

```
cd /opt/intel/openvino/deployment_tools/model_optimizer/install_prerequisites
```

2. Run the script for your model framework. You can run more than one script:

   - For **Caffe**:

     ```
     sudo ./install_prerequisites_caffe.sh
     ```

   - For **TensorFlow**:

     ```
     sudo ./install_prerequisites_tf.sh
     ```

   - For **MXNet**:

     ```
     sudo ./install_prerequisites_mxnet.sh
     ```

   - For **ONNX**:

     ```
     sudo ./install_prerequisites_onnx.sh
     ```

   - For **Kaldi**:

     ```
     sudo ./install_prerequisites_kaldi.sh
     ```

   The Model Optimizer is configured for one or more frameworks.

You are ready to compile the samples by running the verification scripts.

## 4.7 Run the Verification Scripts to Verify Installation

**IMPORTANT**: This section is required. In addition to confirming your installation was successful, demo scripts perform other steps, such as setting up your computer to use the Inference Engine samples.

To verify the installation and compile two samples, run the verification applications provided with the product on the CPU:

1. Go to the **Inference Engine demo** directory:

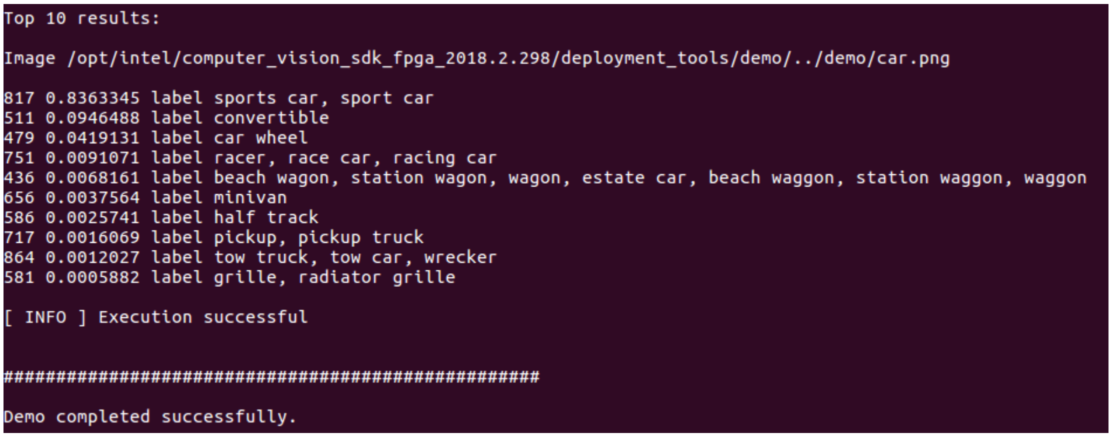
```
cd /opt/intel/openvino/deployment_tools/demo
```

2. Run the **Image Classification verification script**:

```
./demo_squeezenet_download_convert_run.sh
```

This verification script 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.

This verification script builds the Image Classification Sample Async application and run it with the `car.png` image located in the demo directory. When the verification script completes, you will have the label and confidence for the top-10 categories:



```
Top 10 results:

Image /opt/intel/computer_vision_sdk_fpga_2018.2.298/deployment_tools/demo/../demo/car.png

817 0.8363345 label sports car, sport car
511 0.0946488 label convertible
479 0.0419131 label car wheel
751 0.0091071 label racer, race car, racing car
436 0.0068161 label beach wagon, station wagon, wagon, estate car, beach waggon, station waggon, waggon
656 0.0037564 label minivan
586 0.0025741 label half track
717 0.0016069 label pickup, pickup truck
864 0.0012027 label tow truck, tow car, wrecker
581 0.0005882 label grille, radiator grille

[ INFO ] Execution successful


###################################################

Demo completed successfully.
```

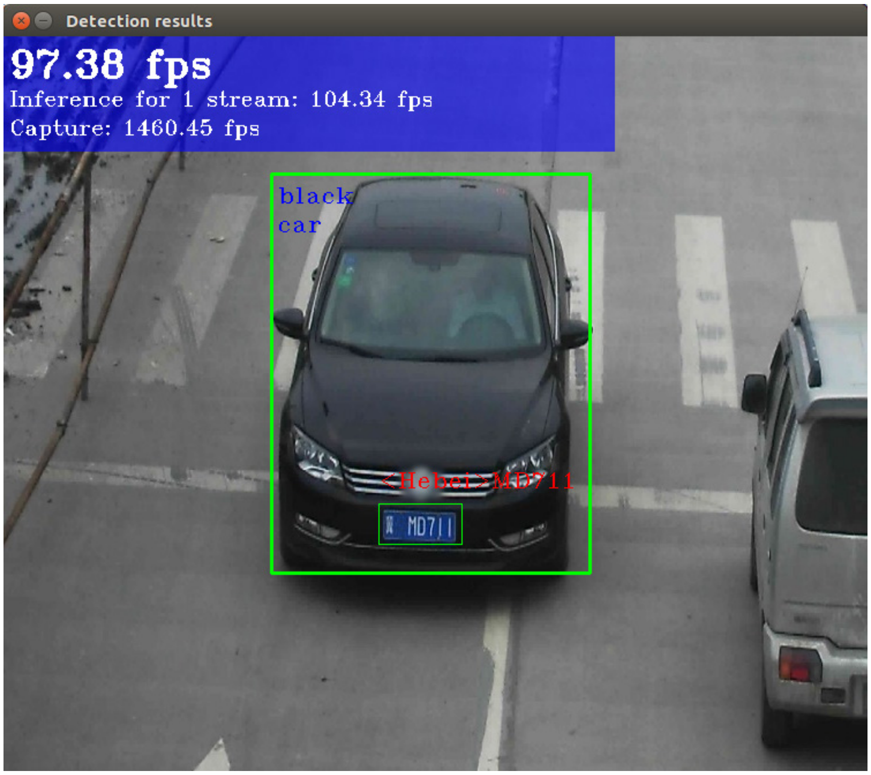
3. Run the **Inference Pipeline verification script**:

```
./demo_security_barrier_camera.sh
```

This script downloads three pre-trained model IRs, builds the Security Barrier Camera Demo application, and runs it with the downloaded models and the `car_1.bmp` image from the `demo` directory to show an inference pipeline. The verification script uses vehicle recognition in which vehicle attributes build on each other to narrow in on a specific attribute.

First, an object is identified as a vehicle. This identification is used as input to the next model, which identifies specific vehicle attributes, including the license plate. Finally, the attributes identified as the license plate are used as input to the third model, which recognizes specific characters in the license plate.

When the verification script completes, you will see an image that displays the resulting frame with detections rendered as bounding boxes, and text:



```
97.38 fps
Inference for 1 stream: 104.34 fps
Capture: 1460.45 fps
```

4. Close the image viewer window to complete the verification script.

## 4.8 Install Mustang-M2AE-MX1 <intel HDDL> Driver

For Intel® Vision Accelerator Design with Intel® Movidius™ VPUs, the following additional installation steps are required.

1. Set the environment variables:

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

   **NOTE**: The `HDDL_INSTALL_DIR` variable is set to `<openvino_install_dir>/deployment_tools/inference_engine/external/hddl`. If you installed the Intel® Distribution of OpenVINO™ to the default install directory, the `HDDL_INSTALL_DIR` was set to `/opt/intel/openvino//deployment_tools/inference_engine/external/hddl`.

2. Install dependencies:

```
${HDDL_INSTALL_DIR}/install_IVAD_VPU_dependencies.sh
```

   Note, if the Linux kernel is updated after the installation, it is required to install drivers again:

```
cd ${HDDL_INSTALL_DIR}/drivers

reboot

sudo ./setup.sh install
```

Now the dependencies are installed and you are ready to use the Intel® Vision Accelerator Design with Intel® Movidius™ with the Intel® Distribution of OpenVINO™ toolkit.

## 4.9 Steps for verify Intel® Vision Accelerator Design with Mustang-M2AE-MX1 Intel® Movidius™ VPU

After configuration is done, you are ready to run the verification scripts with the HDDL Plugin for your Intel® Vision Accelerator Design with Intel® Movidius™ VPUs:

1. Go to the **Inference Engine demo** directory:

```
cd /opt/intel/openvino/deployment_tools/demo
```

2. Run the **Image Classification verification script**. If you have access to the Internet through the proxy server only, please make sure that it is configured in your OS environment.

```
./demo_squeezenet_download_convert_run.sh -d HDDL
```

3. Run the **Inference Pipeline verification script**:

```
./demo_security_barrier_camera.sh -d HDDL
```

Chapter

5

# OpenVINO™ Toolkit Installation (Windows 10)

## 5.1 System Requirement

Operating System:

- Microsoft Windows* 10 64-bit

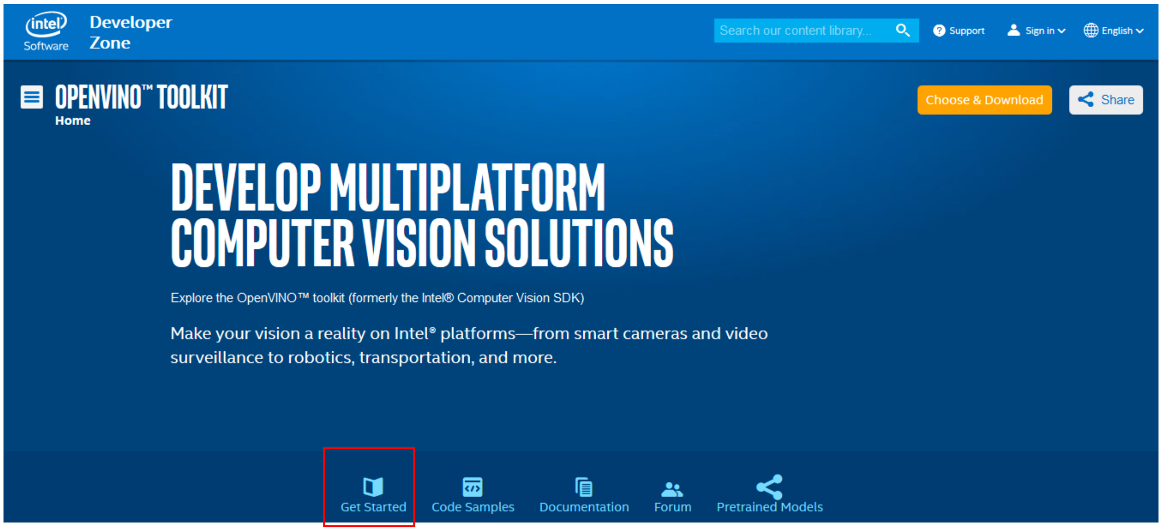
Software:

- Microsoft Visual Studio* with C++ **2019, 2017, or 2015** with MSBuild
- CMake **3.4 or higher** 64-bit

  **NOTE**: If you want to use Microsoft Visual Studio 2019, you are required to install CMake 3.14.
- Python **3.6.5** 64-bit

## 5.2 Online Installation Guide

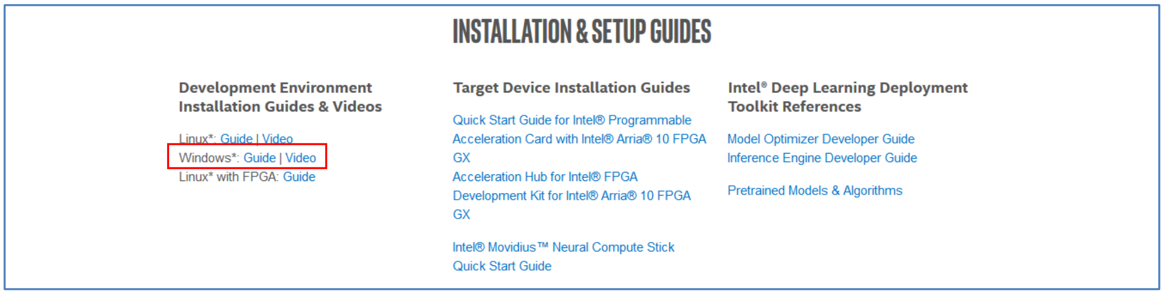
Go to https://software.intel.com/en-us/openvino-toolkit. Click "Get Started" then choose your configuration from "Development Environment Installation Guides & Videos". Follow the instruction to complete the installation procedure.



DEVELOP MULTIPLATFORM COMPUTER VISION SOLUTIONS

Explore the OpenVINO™ toolkit (formerly the Intel® Computer Vision SDK)

Make your vision a reality on Intel® platforms—from smart cameras and video surveillance to robotics, transportation, and more.



INSTALLATION & SETUP GUIDES

**Development Environment Installation Guides & Videos**

Linux*: Guide | Video
Windows*: Guide | Video
Linux* with FPGA: Guide

**Target Device Installation Guides**

Quick Start Guide for Intel® Programmable Acceleration Card with Intel® Arria® 10 FPGA GX
Acceleration Hub for Intel® FPGA
Development Kit for Intel® Arria® 10 FPGA GX

Intel® Movidius™ Neural Compute Stick Quick Start Guide

**Intel® Deep Learning Deployment Toolkit References**

Model Optimizer Developer Guide
Inference Engine Developer Guide

Pretrained Models & Algorithms

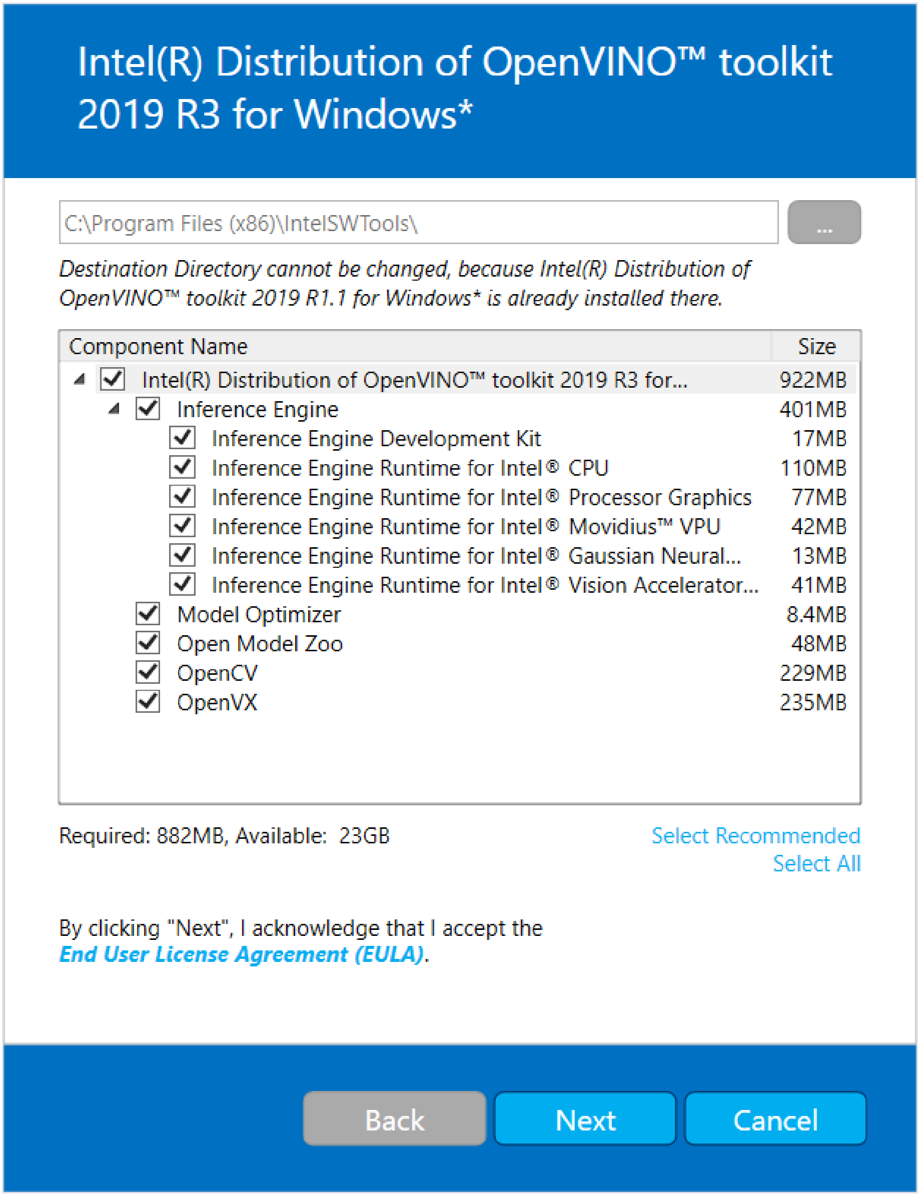
## 5.3 Installation Steps

### 5.3.1 Install the Intel® Distribution of OpenVINO™ toolkit Core Components

1. If you have not downloaded the Intel® Distribution of OpenVINO™ toolkit, download the latest version. By default, the file is saved to the `Downloads` directory as `w_openvino_toolkit_p_<version>.exe`.

2. Go to the `Downloads` folder.

3. Double-click `w_openvino_toolkit_p_<version>.exe`. A window opens to let you choose your installation directory and components. The default installation directory is `C:\Program Files (x86)\IntelSWTools\openvino_<version>`, for simplicity, a shortcut to the latest installation is also created: `C:\Program Files (x86)\IntelSWTools\openvino`. If you choose a different installation directory, the installer will create the directory for you:



Intel(R) Distribution of OpenVINO™ toolkit 2019 R3 for Windows*

C:\Program Files (x86)\IntelSWTools\

*Destination Directory cannot be changed, because Intel(R) Distribution of OpenVINO™ toolkit 2019 R1.1 for Windows* is already installed there.*

| Component Name | Size |
|---|---|
| ◢ ☑ Intel(R) Distribution of OpenVINO™ toolkit 2019 R3 for... | 922MB |
| ◢ ☑ Inference Engine | 401MB |
| ☑ Inference Engine Development Kit | 17MB |
| ☑ Inference Engine Runtime for Intel® CPU | 110MB |
| ☑ Inference Engine Runtime for Intel® Processor Graphics | 77MB |
| ☑ Inference Engine Runtime for Intel® Movidius™ VPU | 42MB |
| ☑ Inference Engine Runtime for Intel® Gaussian Neural... | 13MB |
| ☑ Inference Engine Runtime for Intel® Vision Accelerator... | 41MB |
| ☑ Model Optimizer | 8.4MB |
| ☑ Open Model Zoo | 48MB |
| ☑ OpenCV | 229MB |
| ☑ OpenVX | 235MB |

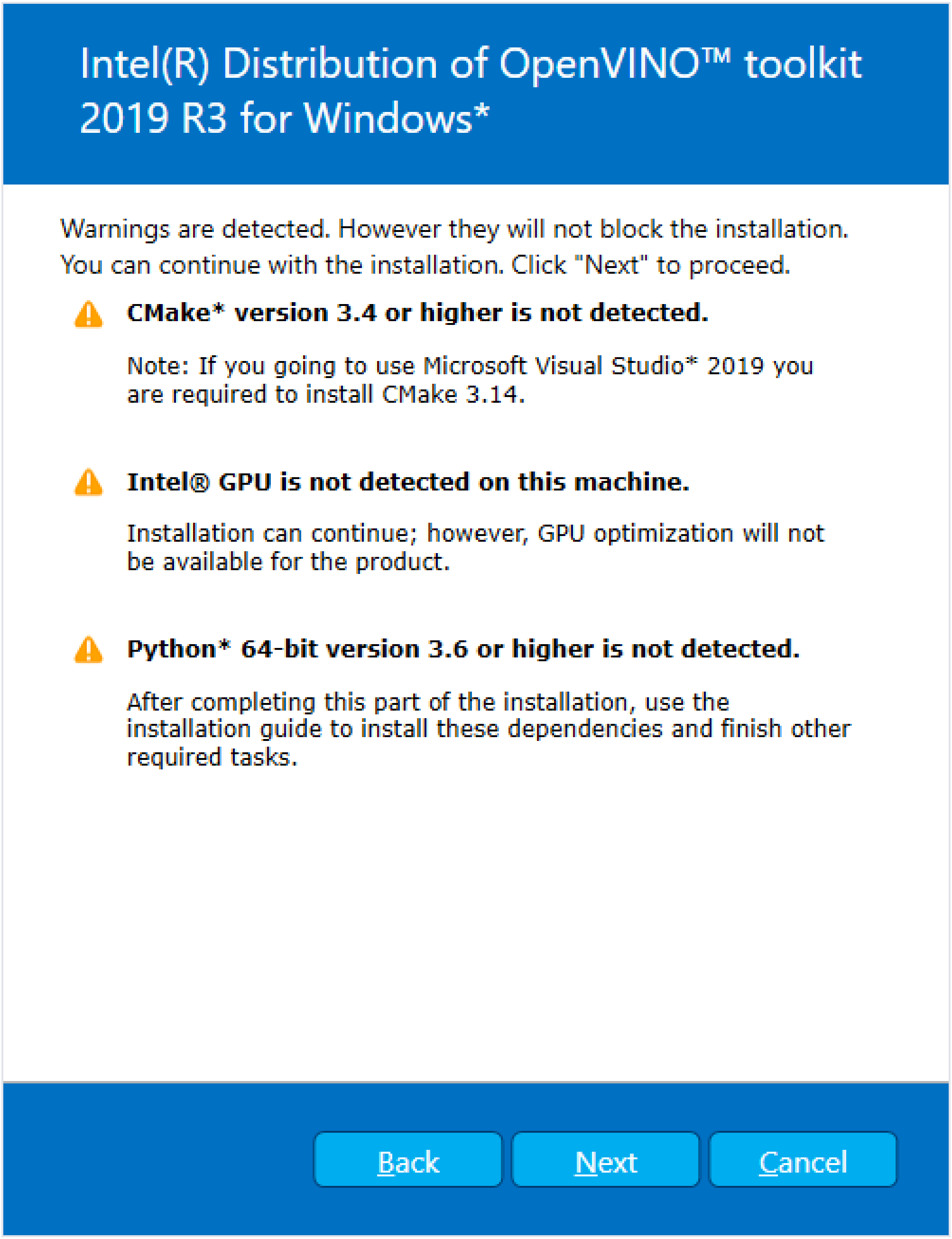
Required: 882MB, Available: 23GB

By clicking "Next", I acknowledge that I accept the ***End User License Agreement (EULA)***.

4. Click **Next**.

5. You are asked if you want to provide consent to gather information. Choose the option of your choice. Click **Next**.

6. If you are missing external dependencies, you will see a warning screen. Write down the dependencies you are missing. **You need to take no other action at this time**. After installing the Intel® Distribution of OpenVINO™ toolkit core components, install the missing dependencies. The screen example below indicates you are missing two dependencies:



Intel(R) Distribution of OpenVINO™ toolkit 2019 R3 for Windows*

Warnings are detected. However they will not block the installation. You can continue with the installation. Click "Next" to proceed.

⚠ **CMake* version 3.4 or higher is not detected.**

Note: If you going to use Microsoft Visual Studio* 2019 you are required to install CMake 3.14.

⚠ **Intel® GPU is not detected on this machine.**

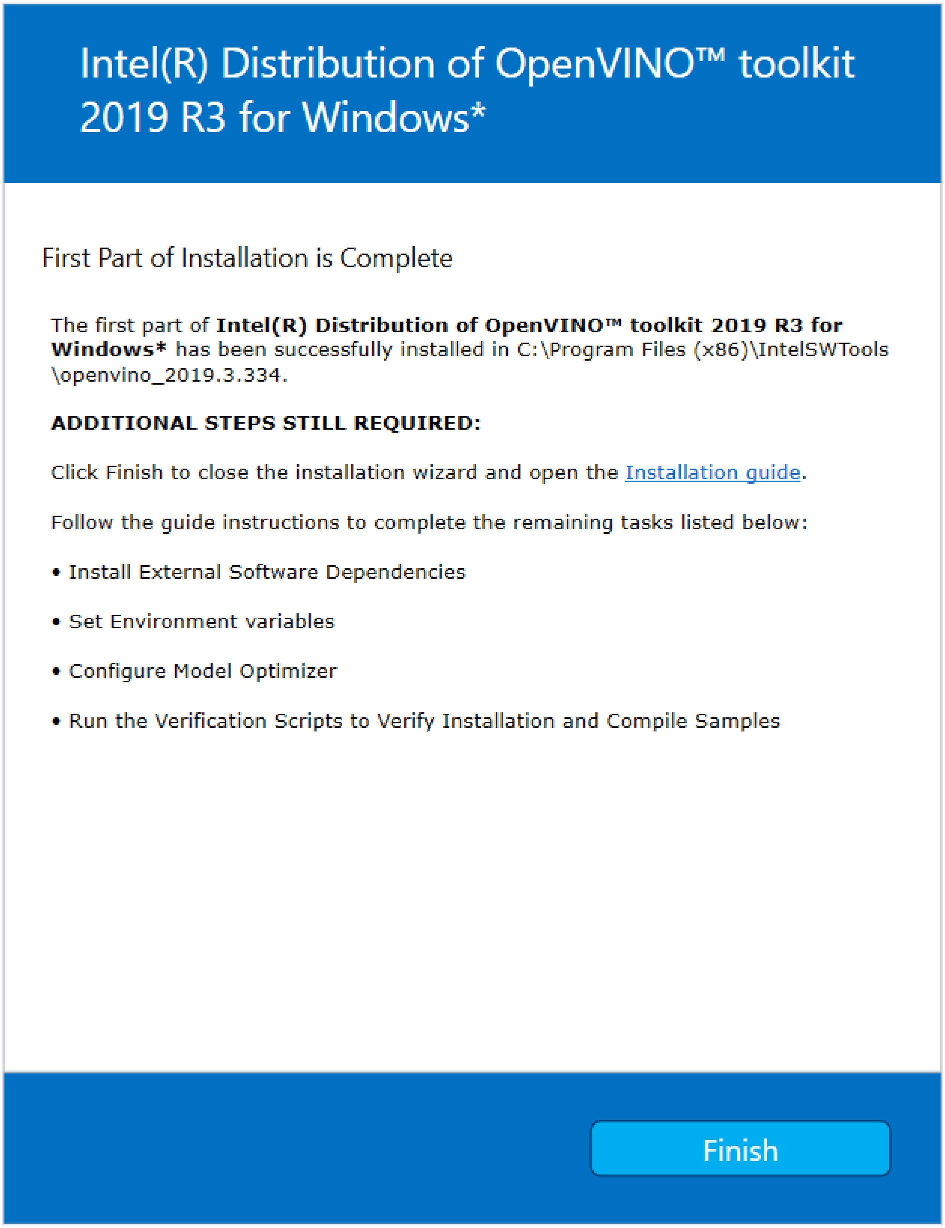
Installation can continue; however, GPU optimization will not be available for the product.

⚠ **Python* 64-bit version 3.6 or higher is not detected.**

After completing this part of the installation, use the installation guide to install these dependencies and finish other required tasks.

7. Click **Next**.

8. When the first part of installation is complete, the final screen informs you that the core components have been installed and additional steps still required:



Intel(R) Distribution of OpenVINO™ toolkit 2019 R3 for Windows*

First Part of Installation is Complete

The first part of **Intel(R) Distribution of OpenVINO™ toolkit 2019 R3 for Windows*** has been successfully installed in C:\Program Files (x86)\IntelSWTools \openvino_2019.3.334.

**ADDITIONAL STEPS STILL REQUIRED:**

Click Finish to close the installation wizard and open the Installation guide.

Follow the guide instructions to complete the remaining tasks listed below:

- Install External Software Dependencies
- Set Environment variables
- Configure Model Optimizer
- Run the Verification Scripts to Verify Installation and Compile Samples

9. Click **Finish** to close the installation wizard. A new browser window opens to the next section of the installation guide to set the environment variables. You are in the same document. The new window opens in case you ran the installation without first opening this installation guide.

10. If the installation indicated you must install dependencies, install them first. If there are no missing dependencies, you can go ahead and set the environment variables.

### 5.3.2 Set the Environment Variables

**NOTE**: If you installed the Intel® Distribution of OpenVINO™ to the non-default install directory, replace `C:\Program Files (x86)\IntelSWTools` with the directory in which you installed the software.

You must update several environment variables before you can compile and run OpenVINO™ applications. Open the Command Prompt, and run the `setupvars.bat` batch file to temporarily set your environment variables:

```
cd C:\Program Files (x86)\IntelSWTools\openvino\bin\

setupvars.bat
```

**(Optional)**: OpenVINO toolkit environment variables are removed when you close the Command Prompt window. As an option, you can permanently set the environment variables manually.

The environment variables are set. Continue to the next section to configure the Model Optimizer.

## 5.4 Configure the Model Optimizer

**IMPORTANT**: These steps are required. You must configure the Model Optimizer for at least one framework. The Model Optimizer will fail if you do not complete the steps in this section.

**NOTE**: If you see an error indicating Python is not installed when you know you installed it, your computer might not be able to find the program. For the instructions to add Python to your system environment variables, see Update Your Windows Environment Variables.

The Model Optimizer is a key component of the Intel® Distribution of OpenVINO™ toolkit. You cannot do inference on your trained model without running the model through the Model Optimizer. When you run a pre-trained model through the Model Optimizer, your output is an Intermediate Representation (IR) of the network. The IR is a pair of files that describe the whole model:

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

The Inference Engine reads, loads, and infers the IR files, using a common API across the CPU, GPU, or VPU hardware.

The Model Optimizer is a Python*-based command line tool (mo.py), which is located in C:\Program Files (x86)\IntelSWTools\openvino\deployment_tools\model_optimizer. Use this tool on models trained with popular deep learning frameworks such as Caffe*, TensorFlow*, MXNet*, and ONNX* to convert them to an optimized IR format that the Inference Engine can use.

This section explains how to use scripts to configure the Model Optimizer either for all of the supported frameworks at the same time or for individual frameworks. If you want to manually configure the Model Optimizer instead of using scripts, see the **Using Manual Configuration Process** section on the Configuring the Model Optimizer page.

For more information about the Model Optimizer, see the Model Optimizer Developer Guide.

### 5.4.1 Model Optimizer Configuration Steps

You can configure the Model Optimizer either for all supported frameworks at once or for one framework at a time. Choose the option that best suits your needs. If you see error messages, make sure you installed all dependencies.

**IMPORTANT**: The Internet access is required to execute the following steps successfully. If you have access to the Internet through the proxy server only, please make sure that it is configured in your environment.

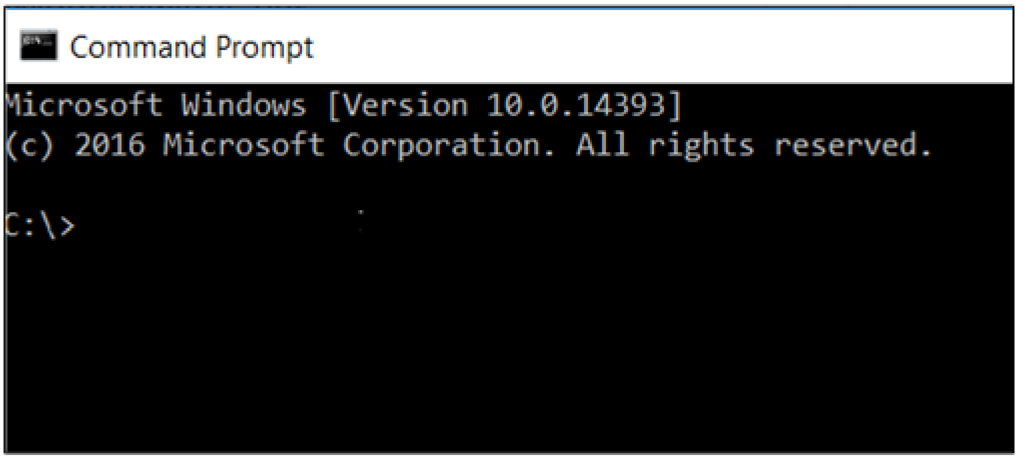
**NOTE**: In the steps below:

- If you you want to use the Model Optimizer from another installed versions of Intel® Distribution of OpenVINO™ toolkit installed, replace openvino with openvino_<version>.
- If you installed the Intel® Distribution of OpenVINO™ toolkit to the non-default installation directory, replace C:\Program Files (x86)\IntelSWTools with the directory where you installed the software.

These steps use a command prompt to make sure you see error messages.

#### 5.4.1.1 Option 1: Configure the Model Optimizer for all supported frameworks at the same time:

1. Open a command prompt. To do so, type `cmd` in your **Search Windows** box and then press **Enter**. Type commands in the opened window:



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\>
```

2. Go to the Model Optimizer prerequisites directory.

```
cd C:\Program Files
(x86)\IntelSWTools\openvino\deployment_tools\model_optimizer\
install_prerequisites
```

3. Run the following batch file to configure the Model Optimizer for Caffe*, TensorFlow*, MXNet*, Kaldi*, and ONNX*:

```
install_prerequisites.bat
```

#### 5.4.1.2 Option 2: Configure the Model Optimizer for each framework separately:

1. Go to the Model Optimizer prerequisites directory:

```
cd C:\Program Files
(x86)\IntelSWTools\openvino\deployment_tools\model_optimizer\
install_prerequisites
```

2. Run the batch file for the framework you will use with the Model Optimizer. You can use more than one:

    - For **Caffe**:

      ```
      install_prerequisites_caffe.bat
      ```

    - For **TensorFlow**:

      ```
      install_prerequisites_tf.bat
      ```

    - For **MXNet**:

      ```
      install_prerequisites_mxnet.bat
      ```

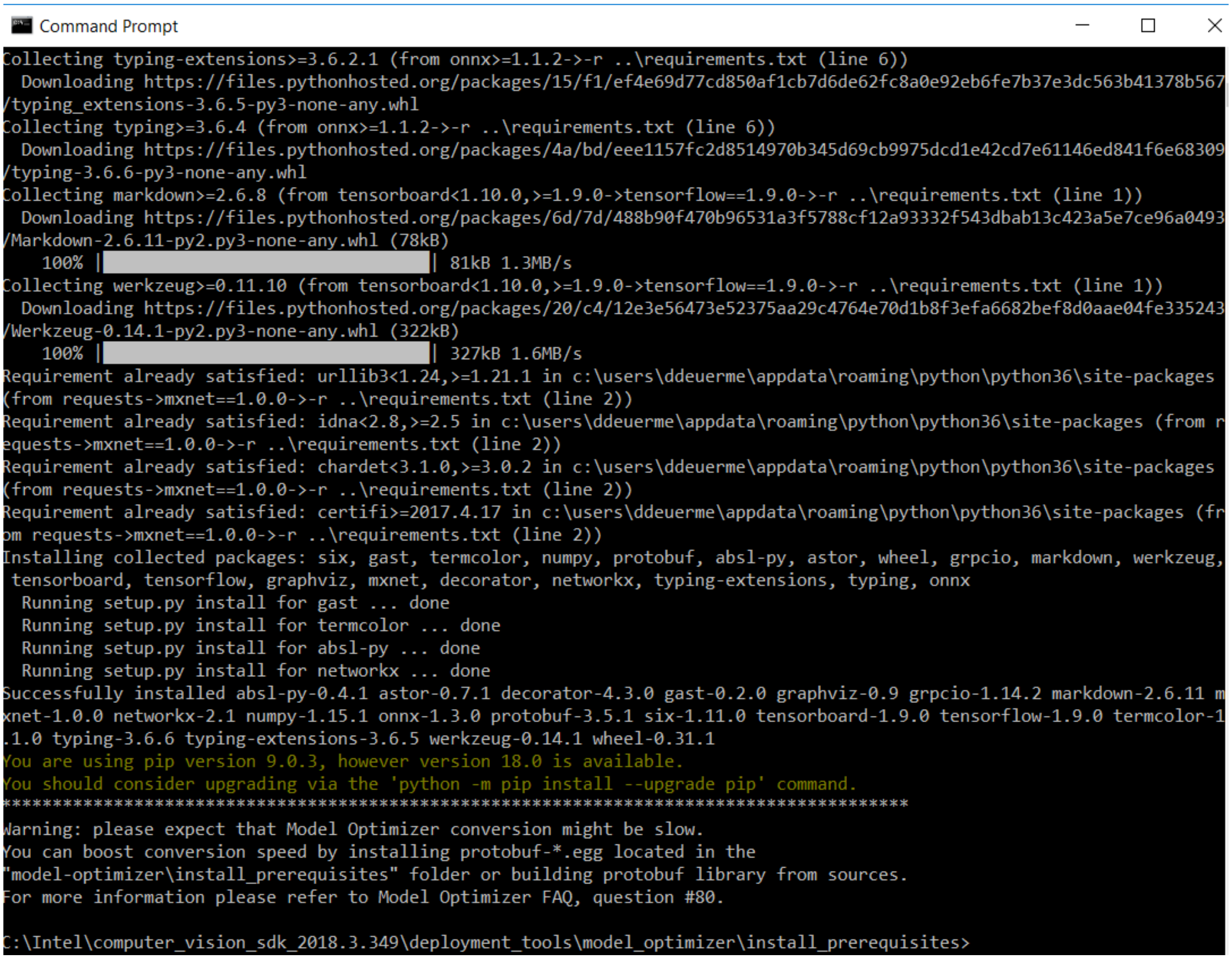
    - For **ONNX**:

      ```
      install_prerequisites_onnx.bat
      ```

    - For **Kaldi**:

      ```
      install_prerequisites_kaldi.bat
      ```

The Model Optimizer is configured for one or more frameworks. Success is indicated by a screen similar to this:



```
Collecting typing-extensions>=3.6.2.1 (from onnx>=1.1.2->-r ..\requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/15/f1/ef4e69d77cd850af1cb7d6de62fc8a0e92eb6fe7b37e3dc563b41378b567
/typing_extensions-3.6.5-py3-none-any.whl
Collecting typing>=3.6.4 (from onnx>=1.1.2->-r ..\requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/4a/bd/eee1157fc2d8514970b345d69cb9975dcd1e42cd7e61146ed841f6e68309
/typing-3.6.6-py3-none-any.whl
Collecting markdown>=2.6.8 (from tensorboard<1.10.0,>=1.9.0->tensorflow==1.9.0->-r ..\requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/6d/7d/488b90f470b96531a3f5788cf12a93332f543dbab13c423a5e7ce96a0493
/Markdown-2.6.11-py2.py3-none-any.whl (78kB)
    100% |████████████████████████████████| 81kB 1.3MB/s
Collecting werkzeug>=0.11.10 (from tensorboard<1.10.0,>=1.9.0->tensorflow==1.9.0->-r ..\requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/20/c4/12e3e56473e52375aa29c4764e70d1b8f3efa6682bef8d0aae04fe335243
/Werkzeug-0.14.1-py2.py3-none-any.whl (322kB)
    100% |████████████████████████████████| 327kB 1.6MB/s
Requirement already satisfied: urllib3<1.24,>=1.21.1 in c:\users\ddeuerme\appdata\roaming\python\python36\site-packages
(from requests->mxnet==1.0.0->-r ..\requirements.txt (line 2))
Requirement already satisfied: idna<2.8,>=2.5 in c:\users\ddeuerme\appdata\roaming\python\python36\site-packages (from r
equests->mxnet==1.0.0->-r ..\requirements.txt (line 2))
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\ddeuerme\appdata\roaming\python\python36\site-packages
(from requests->mxnet==1.0.0->-r ..\requirements.txt (line 2))
Requirement already satisfied: certifi>=2017.4.17 in c:\users\ddeuerme\appdata\roaming\python\python36\site-packages (fr
om requests->mxnet==1.0.0->-r ..\requirements.txt (line 2))
Installing collected packages: six, gast, termcolor, numpy, protobuf, absl-py, astor, wheel, grpcio, markdown, werkzeug,
 tensorboard, tensorflow, graphviz, mxnet, decorator, networkx, typing-extensions, typing, onnx
  Running setup.py install for gast ... done
  Running setup.py install for termcolor ... done
  Running setup.py install for absl-py ... done
  Running setup.py install for networkx ... done
Successfully installed absl-py-0.4.1 astor-0.7.1 decorator-4.3.0 gast-0.2.0 graphviz-0.9 grpcio-1.14.2 markdown-2.6.11 m
xnet-1.0.0 networkx-2.1 numpy-1.15.1 onnx-1.3.0 protobuf-3.5.1 six-1.11.0 tensorboard-1.9.0 tensorflow-1.9.0 termcolor-1
.1.0 typing-3.6.6 typing-extensions-3.6.5 werkzeug-0.14.1 wheel-0.31.1
You are using pip version 9.0.3, however version 18.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
*****************************************************************************************
Warning: please expect that Model Optimizer conversion might be slow.
You can boost conversion speed by installing protobuf-*.egg located in the
"model-optimizer\install_prerequisites" folder or building protobuf library from sources.
For more information please refer to Model Optimizer FAQ, question #80.

C:\Intel\computer_vision_sdk_2018.3.349\deployment_tools\model_optimizer\install_prerequisites>
```

You are ready to use two short demos to see the results of running the Intel Distribution of OpenVINO toolkit and to verify your installation was successful. The demo scripts are required since they perform additional configuration steps. Continue to the next section.

If you want to use a GPU or VPU, or update your Windows* environment variables, read through the Optional Steps section.

## 5.5 Use Verification Scripts to Verify Your Installation

**IMPORTANT**: This section is required. In addition to confirming your installation was successful, demo scripts perform other steps, such as setting up your computer to use the Inference Engine samples.

**NOTE**: The paths in this section assume you used the default installation directory. If you used a directory other than `C:\Program Files (x86)\IntelSWTools`, update the directory with the location where you installed the software.

To verify the installation and compile two samples, run the verification applications provided with the product on the CPU:

1. Open a command prompt window.

2. Go to the Inference Engine demo directory:

```
cd C:\Program Files
(x86)\IntelSWTools\openvino\deployment_tools\demo\
```

3. Run the verification scripts by following the instructions in the next section.

### 5.5.1 Run the Image Classification Verification Script

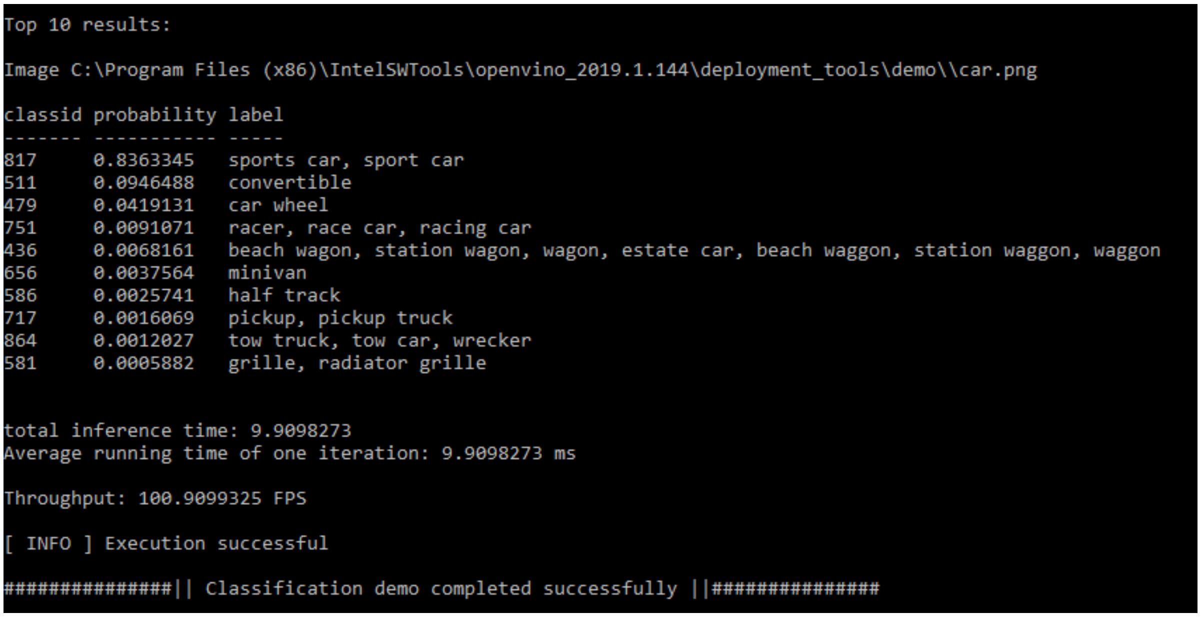
To run the script, start the `demo_squeezenet_download_convert_run.bat` file:

```
demo_squeezenet_download_convert_run.bat
```

This script 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.

This verification script builds the Image Classification Sample Async application and run it with the `car.png` image in the demo directory. For a brief description of the Intermediate Representation, see Configuring the Model Optimizer.

When the verification script completes, you will have the label and confidence for the top-10 categories:



```
Top 10 results:

Image C:\Program Files (x86)\IntelSWTools\openvino_2019.1.144\deployment_tools\demo\\car.png

classid probability label
------- ----------- -----
817     0.8363345   sports car, sport car
511     0.0946488   convertible
479     0.0419131   car wheel
751     0.0091071   racer, race car, racing car
436     0.0068161   beach wagon, station wagon, wagon, estate car, beach waggon, station waggon, waggon
656     0.0037564   minivan
586     0.0025741   half track
717     0.0016069   pickup, pickup truck
864     0.0012027   tow truck, tow car, wrecker
581     0.0005882   grille, radiator grille


total inference time: 9.9098273
Average running time of one iteration: 9.9098273 ms

Throughput: 100.9099325 FPS

[ INFO ] Execution successful

###############|| Classification demo completed successfully ||###############
```

This demo is complete. Leave the console open and continue to the next section to run the Inference Pipeline demo.

### 5.5.2 Run the Inference Pipeline Verification Script

To run the script, start the `demo_security_barrier_camera.bat` file while still in the console:

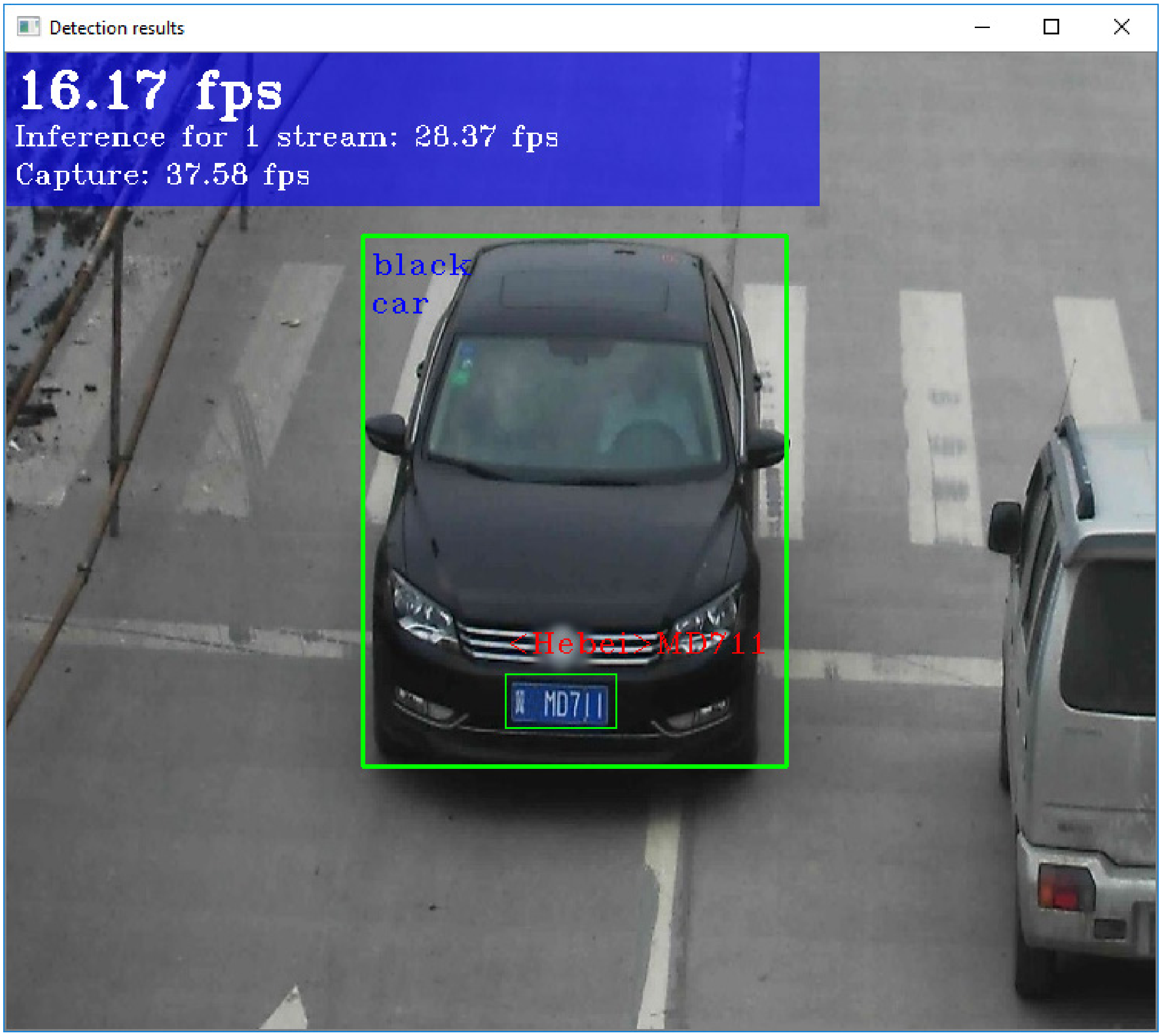
```
demo_security_barrier_camera.bat
```

This script downloads three pre-trained model IRs, builds the Security Barrier Camera Demo application, and runs it with the downloaded models and the `car_1.bmp` image from the `demo` directory to show an inference pipeline. The verification script uses vehicle recognition in which vehicle attributes build on each other to narrow in on a specific attribute.

First, an object is identified as a vehicle. This identification is used as input to the next model, which identifies specific vehicle attributes, including the license plate. Finally, the attributes identified as the license plate are used as input to the third model, which recognizes specific characters in the license plate.

When the demo completes, you have two windows open:

- A console window that displays information about the tasks performed by the demo
- An image viewer window that displays a resulting frame with detections rendered as bounding boxes, similar to the following:



Close the image viewer window to end the demo.

To learn more about the verification scripts, see `README.txt` in `C:\Program Files (x86)\IntelSWTools\openvino\deployment_tools\demo`.

### 5.5.3 Installation Steps for iEi Mustang-M2AE-MX1 the Intel® Vision Accelerator Design with Intel® Movidius™ VPU

**NOTE**: These steps are required only if you want to use Intel® Vision Accelerator Design with Intel® Movidius™ VPUs.

To perform inference on Intel® Vision Accelerator Design with Intel® Movidius™ VPUs, the following additional installation steps are required:

1. Install the Movidius™ VSC driver:

   a. Go to the `<INSTALL_DIR>\deployment_tools\inference-engine\external\MovidiusDriver` directory, where `<INSTALL_DIR>` is the directory in which the Intel Distribution of OpenVINO toolkit is installed.

   b. Right click on the `Movidius_VSC_Device.inf` file and choose **Install** from the pop up menu.

2. If your Intel® Vision Accelerator Design with Intel® Movidius™ VPUs card requires SMBUS connection to PCIe slot (Raw video data card with HW version Fab-B and before), install the SMBUS driver:

   a. Go to the `<INSTALL_DIR>\deployment_tools\inference-engine\external\hddl\SMBusDriver` directory, where `<INSTALL_DIR>` is the directory in which the Intel Distribution of OpenVINO toolkit is installed.

   b. Right click on the `hddlsmbus.inf` file and choose **Install** from the pop up menu.

3. Download and install Visual C++ Redistributable for Visual Studio 2015

You are done installing your device driver and are ready to use your Intel® Vision Accelerator Design with Intel® Movidius™ VPUs.

See also:

- For advanced configuration steps for your IEI Mustang-V100-MX8 accelerator, see Intel® Movidius™ VPUs Setup Guide for Use with Intel® Distribution of OpenVINO™ toolkit.
- After you've configurated your Intel® Vision Accelerator Design with Intel® Movidius™ VPUs, see Intel® Movidius™ VPUs Programming Guide for Use with Intel® Distribution of OpenVINO™ toolkit to learn how to distribute a model across all 8 VPUs to maximize performance.

After configuration is done, you are ready to run the verification scripts with the HDDL Plugin for your Intel® Vision Accelerator Design with Intel® Movidius™ VPUs.

1. Open a command prompt window.

2. Go to the Inference Engine demo directory:

```
cd C:\Program Files
(x86)\IntelSWTools\openvino\deployment_tools\demo\
```

3. Run the Image Classification verification script. If you have access to the Internet through the proxy server only, please make sure that it is configured in your environment.

```
demo_squeezenet_download_convert_run.bat -d HDDL
```

4. Run the Inference Pipeline verification script:

```
demo_security_barrier_camera.bat -d HDDL
```

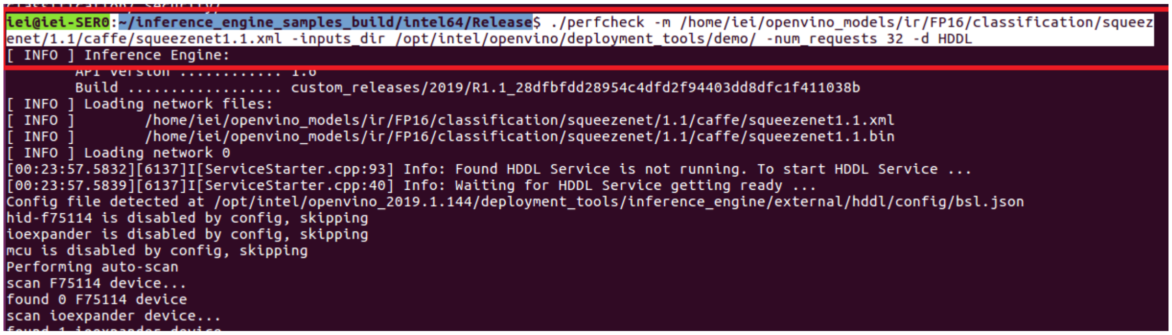
# Appendix A

# Performance Test

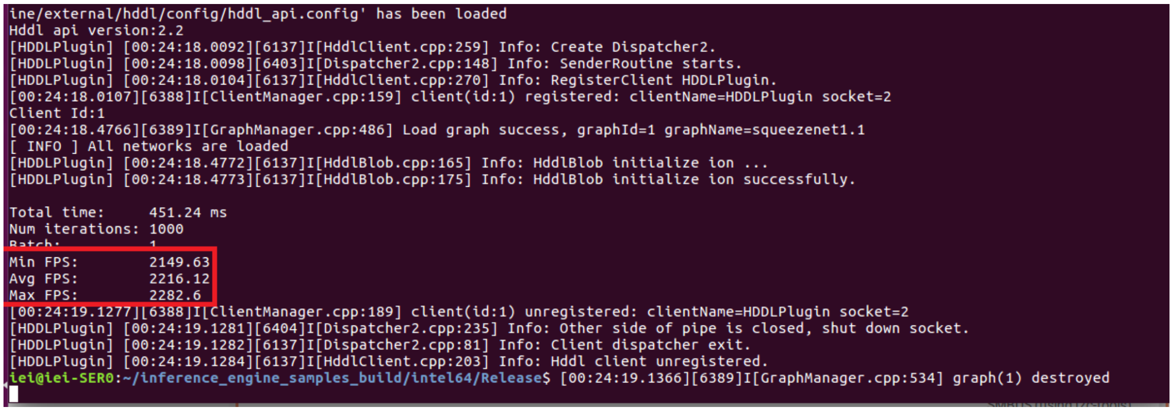
./perfcheck -m **<model_Dir>** -inputs_dir**<image_Dir>** -num_requests **8** -d **HDDL**

Note: suggest -num_requests # = 4xVPU qty. (Mustang-M2AE-MX1=> 8)

Ex. squeezenet1.1.xml



```
iei@iei-SER0:~/inference_engine_samples_build/intel64/Release$ ./perfcheck -m /home/iei/openvino_models/ir/FP16/classification/squeez
enet/1.1/caffe/squeezenet1.1.xml -inputs_dir /opt/intel/openvino/deployment_tools/demo/ -num_requests 32 -d HDDL
[ INFO ] Inference Engine:
        API version ............ 1.6
        Build .................. custom_releases/2019/R1.1_28dfbfdd28954c4dfd2f94403dd8dfc1f411038b
[ INFO ] Loading network files:
[ INFO ]        /home/iei/openvino_models/ir/FP16/classification/squeezenet/1.1/caffe/squeezenet1.1.xml
[ INFO ]        /home/iei/openvino_models/ir/FP16/classification/squeezenet/1.1/caffe/squeezenet1.1.bin
[ INFO ] Loading network 0
[00:23:57.5832][6137]I[ServiceStarter.cpp:93] Info: Found HDDL Service is not running. To start HDDL Service ...
[00:23:57.5839][6137]I[ServiceStarter.cpp:40] Info: Waiting for HDDL Service getting ready ...
Config file detected at /opt/intel/openvino_2019.1.144/deployment_tools/inference_engine/external/hddl/config/bsl.json
hid-f75114 is disabled by config, skipping
ioexpander is disabled by config, skipping
mcu is disabled by config, skipping
Performing auto-scan
scan F75114 device...
found 0 F75114 device
scan ioexpander device...
found 1 ioexpander device
```



```
ine/external/hddl/config/hddl_api.config' has been loaded
Hddl api version:2.2
[HDDLPlugin] [00:24:18.0092][6137]I[HddlClient.cpp:259] Info: Create Dispatcher2.
[HDDLPlugin] [00:24:18.0098][6403]I[Dispatcher2.cpp:148] Info: SenderRoutine starts.
[HDDLPlugin] [00:24:18.0104][6137]I[HddlClient.cpp:270] Info: RegisterClient HDDLPlugin.
[00:24:18.0107][6388]I[ClientManager.cpp:159] client(id:1) registered: clientName=HDDLPlugin socket=2
Client Id:1
[00:24:18.4766][6389]I[GraphManager.cpp:486] Load graph success, graphId=1 graphName=squeezenet1.1
[ INFO ] All networks are loaded
[HDDLPlugin] [00:24:18.4772][6137]I[HddlBlob.cpp:165] Info: HddlBlob initialize ion ...
[HDDLPlugin] [00:24:18.4773][6137]I[HddlBlob.cpp:175] Info: HddlBlob initialize ion successfully.

Total time:     451.24 ms
Num iterations: 1000
Batch:          1
Min FPS:        2149.63
Avg FPS:        2216.12
Max FPS:        2282.6
[00:24:19.1277][6388]I[ClientManager.cpp:189] client(id:1) unregistered: clientName=HDDLPlugin socket=2
[HDDLPlugin] [00:24:19.1281][6404]I[Dispatcher2.cpp:235] Info: Other side of pipe is closed, shut down socket.
[HDDLPlugin] [00:24:19.1282][6137]I[Dispatcher2.cpp:81] Info: Client dispatcher exit.
[HDDLPlugin] [00:24:19.1284][6137]I[HddlClient.cpp:203] Info: Hddl client unregistered.
iei@iei-SER0:~/inference_engine_samples_build/intel64/Release$ [00:24:19.1366][6389]I[GraphManager.cpp:534] graph(1) destroyed
```

Appendix

B

# Regulatory Compliance

## DECLARATION OF CONFORMITY

This equipment has been tested and found to comply with specifications for CE marking. If the user modifies and/or installs other devices in the equipment, the CE conformity declaration may no longer apply.

## FCC WARNING

This equipment complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- This device may not cause harmful interference, and
- This device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Appendix

C

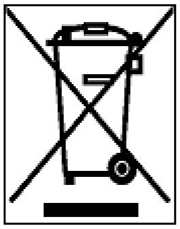
# Product Disposal

**CAUTION:**

Risk of explosion if battery is replaced by an incorrect type. Only certified engineers should replace the on-board battery.

Dispose of used batteries according to instructions and local regulations.

- Outside the European Union – If you wish to dispose of used electrical and electronic products outside the European Union, please contact your local authority so as to comply with the correct disposal method.
- Within the European Union – The device that produces less waste and is easier to recycle is classified as electronic device in terms of the European Directive 2012/19/EU (WEEE), and must not be disposed of as domestic garbage.



EU-wide legislation, as implemented in each Member State, requires that waste electrical and electronic products carrying the mark (left) must be disposed of separately from normal household waste. This includes monitors and electrical accessories, such as signal cables or power cords. When you need to dispose of your device, please follow the guidance of your local authority, or ask the shop where you purchased the product. The mark on electrical and electronic products only applies to the current European Union Member States.

Please follow the national guidelines for electrical and electronic product disposal.

# Appendix D

# Hazardous Materials Disclosure

The details provided in this appendix are to ensure that the product is compliant with the Peoples Republic of China (China) RoHS standards. The table below acknowledges the presences of small quantities of certain materials in the product, and is applicable to China RoHS only.

A label will be placed on each product to indicate the estimated "Environmentally Friendly Use Period" (EFUP). This is an estimate of the number of years that these substances would "not leak out or undergo abrupt change." This product may contain replaceable sub-assemblies/components which have a shorter EFUP such as batteries and lamps. These components will be separately marked.

Please refer to the following table.

| Part Name | Toxic or Hazardous Substances and Elements | | | | | |
|---|---|---|---|---|---|---|
| | Lead (Pb) | Mercury (Hg) | Cadmium (Cd) | Hexavalent Chromium (CR(VI)) | Polybrominated Biphenyls (PBB) | Polybrominated Diphenyl Ethers (PBDE) |
| Housing | O | O | O | O | O | O |
| Display | O | O | O | O | O | O |
| Printed Circuit Board | O | O | O | O | O | O |
| Metal Fasteners | O | O | O | O | O | O |
| Cable Assembly | O | O | O | O | O | O |
| Fan Assembly | O | O | O | O | O | O |
| Power Supply Assemblies | O | O | O | O | O | O |
| Battery | O | O | O | O | O | O |

O: This toxic or hazardous substance is contained in all of the homogeneous materials for the part is below the limit requirement in SJ/T11363-2006 (now replaced by GB/T 26572-2011).

X: This toxic or hazardous substance is contained in at least one of the homogeneous materials for this part is above the limit requirement in SJ/T11363-2006 (now replaced by GB/T 26572-2011).

此附件旨在确保本产品符合中国 RoHS 标准。以下表格标示此产品中某有毒物质的含量符合中国 RoHS 标准规定的限量要求。

本产品上会附有"环境友好使用期限"的标签，此期限是估算这些物质"不会有泄漏或突变"的年限。本产品可能包含有较短的环境友好使用期限的可替换元件，像是电池或灯管，这些元件将会单独标示出来。

| 部件名称 | 有毒有害物质或元素 | | | | | |
|---|---|---|---|---|---|---|
| | 铅<br>(Pb) | 汞<br>(Hg) | 镉<br>(Cd) | 六价铬<br>(CR(VI)) | 多溴联苯<br>(PBB) | 多溴二苯醚<br>(PBDE) |
| 壳体 | O | O | O | O | O | O |
| 显示 | O | O | O | O | O | O |
| 印刷电路板 | O | O | O | O | O | O |
| 金属螺帽 | O | O | O | O | O | O |
| 电缆组装 | O | O | O | O | O | O |
| 风扇组装 | O | O | O | O | O | O |
| 电力供应组装 | O | O | O | O | O | O |
| 电池 | O | O | O | O | O | O |
| O: 表示该有毒有害物质在该部件所有物质材料中的含量均在 SJ/T 11363-2006 (现由 GB/T 26572-2011 取代) 标准规定的限量要求以下。<br>X: 表示该有毒有害物质至少在该部件的某一均质材料中的含量超出 SJ/T 11363-2006 (现由 GB/T 26572-2011 取代) 标准规定的限量要求。 | | | | | | |