**Running Intel OpenVINO and Running Inferences of Pre-Trained Models - on Windows 10 and Ubuntu**

About

**OpenVINO** (Open Visual Inferencing and Neural Network Optimization) toolkit quickly deploys applications and solutions that emulate human vision. Based on Convolutional Neural Networks (CNNs), the toolkit extends computer vision (CV) workloads across Intel® hardware, maximizing performance. The OpenVINO toolkit includes the Deep Learning Deployment Toolkit (DLDT).

The Intel Distribution of OpenVINO toolkit for Windows 10 OS:

* Enables CNN-based deep learning inference on the edge.
* Supports heterogeneous execution across Intel CPU, Intel Processor Graphics (GPU), Intel Neural Compute Stick 2, and Intel Vision Accelerator Design with Intel Movidius VPUs.
* Speeds time-to-market through an easy-to-use library of computer vision functions and pre-optimized kernels.
* Includes optimized calls for computer vision standards including OpenCV and OpenCL.

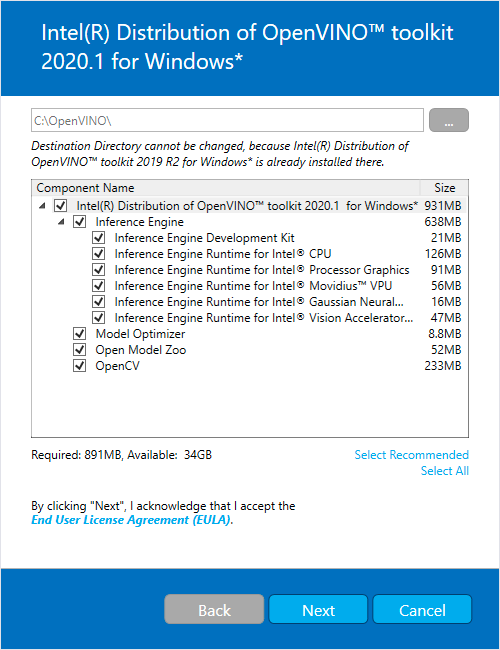
**Windows 10**

Main requirements

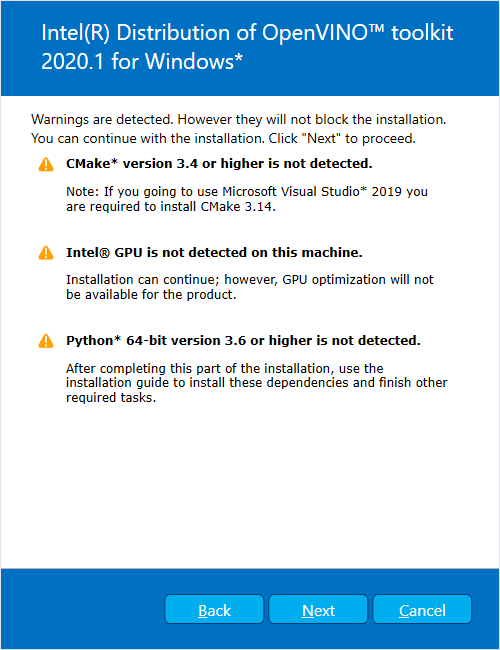
* Microsoft Windows 10 64-bit.
* Microsoft Visual Studio with C++ 2019 or 2017 with MSBuild.
* CMake 2.8.12 or higher 64-bit.
* Python 3.5 - 3.7 64-bit

Installation steps

1. Install the Intel Distribution of OpenVINO toolkit core components
   1. If you’ve not downloaded, then download the latest version from the following link: <http://software.intel.com/en-us/openvino-toolkit/choose-download/free-download-windows>
   2. Go to Downloads folder and 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.



* 1. Click Next. Then you are asked to provide consent to gather information. Choose the option of your choice. Click Next.
  2. 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.



* 1. Click Next. Final screen informs that the core components have been installed and additional steps still required.
  2. Simply click Finish to close the installation wizard. A new window opens to the next section of the installation guide to set environment variables.
  3. If the installation indicated you must install dependencies, install them first. If there are no missing dependencies, you can go ahead andset the environment variables. <https://docs.openvinotoolkit.org/latest/openvino_docs_install_guides_installing_openvino_windows.html#set-the-environment-variables>

1. Set the Environment Variables
   1. 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:
      1. cd C:\Program Files (x86)\IntelSWTools\openvino\bin\
      2. setupvars.bat
2. Configure the Model Optimizer

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.

* [Option 1] Configure the Model Optimizer for all supported frameworks at the same time:
* Open a command prompt.
* Go to the Model Optimizer prerequisites directory.

*cd C:\Program Files (x86) \ IntelSWTools \ openvino \ deployment\_tools \ model\_optimizer \ install\_prerequisites*

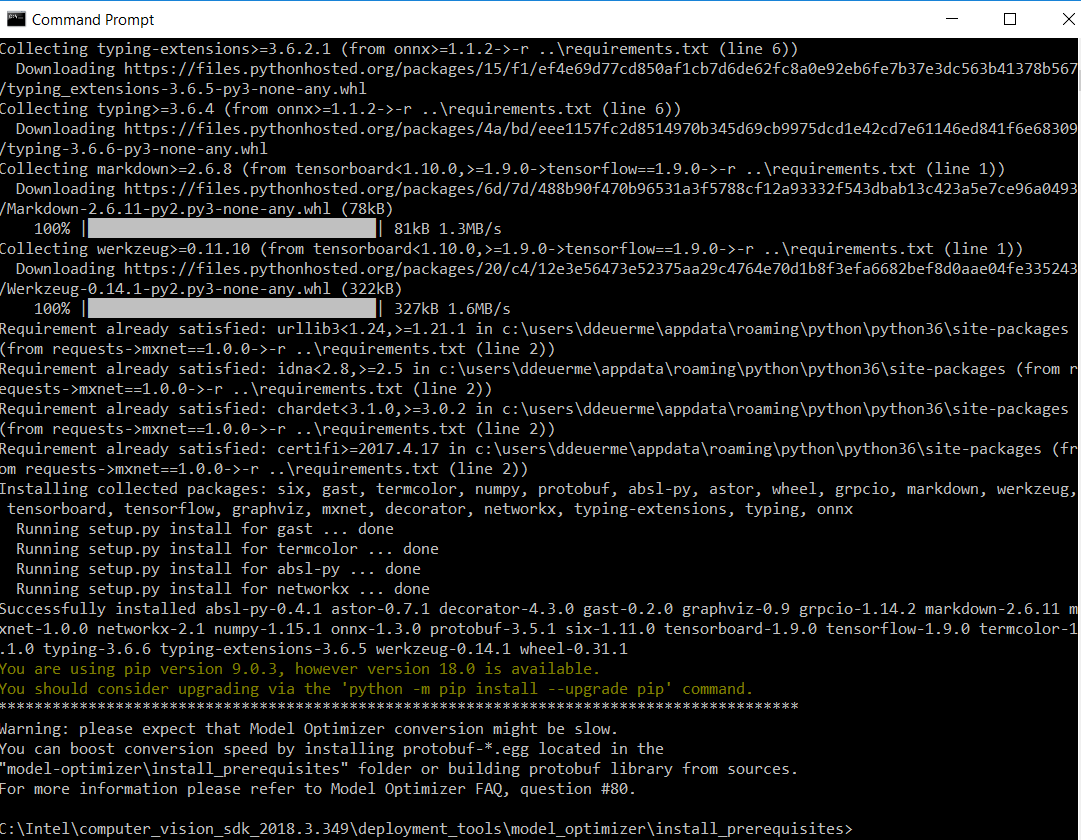
* Run the following batch file to configure the Model Optimizer for Caffe\*, TensorFlow\*, MXNet\*, Kaldi\*, and ONNX\*.

*install\_prerequisites.bat*

* [Option 2] Configure the Model Optimizer for each framework separately:
* Go to the Model Optimizer prerequisites directory.

*cd C:\Program Files (x86) \ IntelSWTools \ openvino \ deployment\_tools \ model\_optimizer \ install\_prerequisites*

* 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
* Successful screen will be like this:



**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: <https://docs.openvinotoolkit.org/latest/openvino_docs_install_guides_installing_openvino_windows.html#Update-Path>

4. Verifying the installation

1. Open command prompt.
2. Go to the inference Engine demo directory:

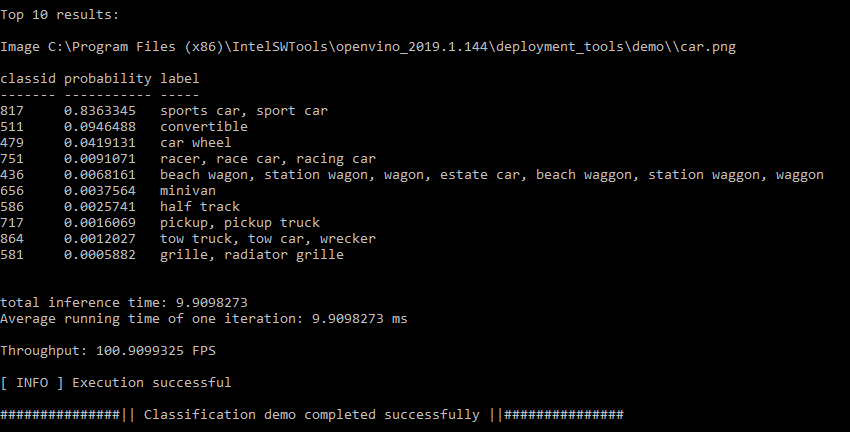
*cd C:\Program Files (x86)\IntelSWTools\openvino\deployment\_tools\demo\*

1. Run the verification scripts by following the instructions:

* Run the Image Classification Verification script
* *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.

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



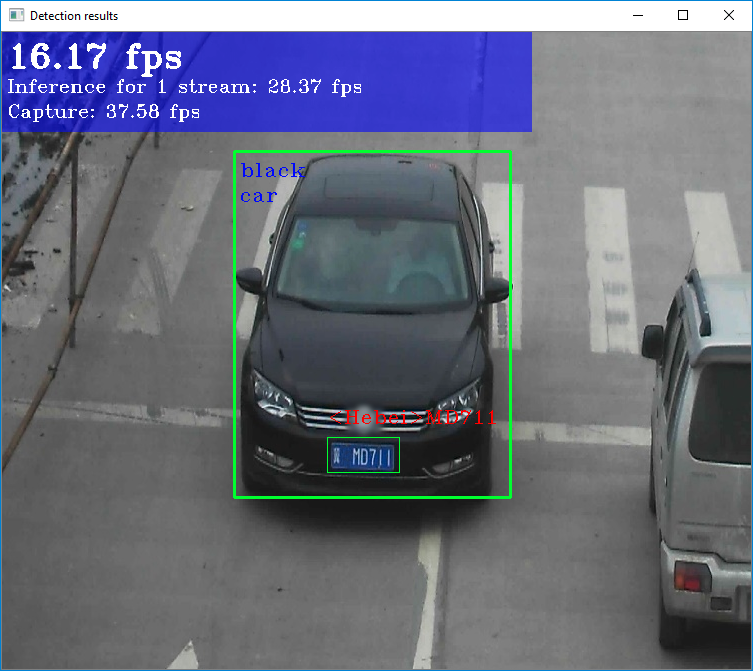
* Demo is complete. Leave the console open and follow the next set of instructions to Run Inferences.

5. Running inferences of Pre-trained Models

* Run the *demo\_security\_barrier\_camera.bat*
* This script downloads 3 pre-trained models IRs, builds the Security Barrier Camera Demo application, and runs it with the downloaded models and *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.

* After completion of the demo, two windows open:
* Console window displaying information about the tasks performed by the demo.
* Image viewer window that displays a resulting frame with detections rendered as bounding boxes.



* For more Pre-trained models: <https://github.com/openvinotoolkit/open_model_zoo>

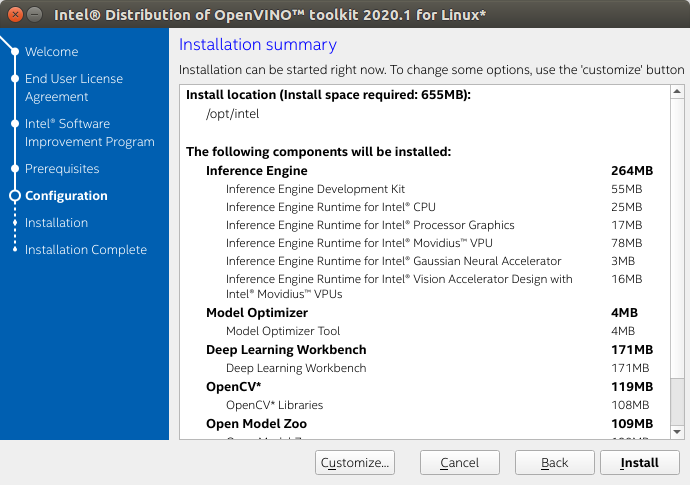
**Ubuntu**

Requirements

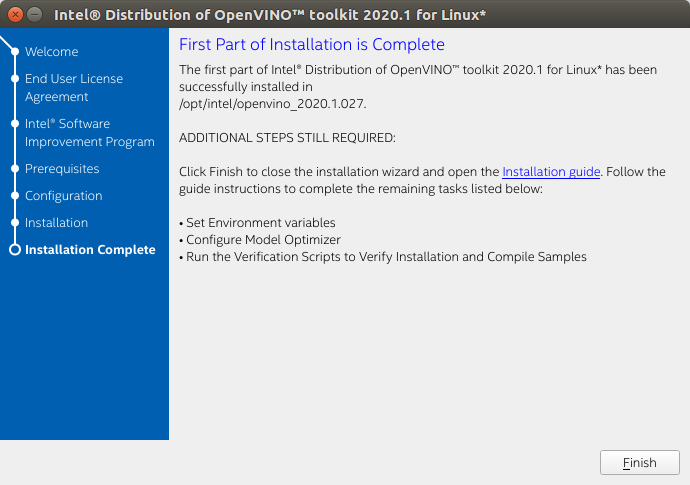
* Ubuntu 18.04.x long-term support (LTS), 64-bit

Installation Steps

* Install the Intel Distribution of OpenVINO Toolkit Core Components
* Download the toolkit: <https://software.intel.com/en-us/openvino-toolkit/choose-download>
* Open a terminal window.
* Change directory to Downloads: *cd ~/Downloads/*
* Unpack the .tgz file: *tar -xvzf l\_openvino\_toolkit\_p\_<version>.tgz*
* Go to the *l\_openvino\_toolkit\_p\_<version>* directory: *cd l\_openvino\_toolkit\_p\_<version>*
* ~/inference\_engine\_samples\_build
* ~/openvino\_models
* Choose the installation option (GUI or CLI) and run the script as root.
* GUI Installation: *sudo ./install\_GUI.sh*
* Command Line: *sudo ./install.sh*
* For GUI, follow the instructions on your screen.
* Installation summary (GUI) would look like this:



* 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/.*
* A Complete screen indicates that the core components have been installed:



## Install External Software Dependencies

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

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

* sudo -E ./install\_openvino\_dependencies.sh
* Set the Environment Variables

**Temporarily set environment variables**

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

**Permanently set environment variables**

1. Open .bashrc file in <user> directory: *vi <user\_directory>/.bashrc*
2. Add the following line to the end of the file: *source /opt/intel/openvino/bin/setupvars.sh*
3. Save and close the file.
4. Test your change by opening a new terminal. You will see “[setupvars.sh] OpenVINO environment initialized”.

* Configure the Model Optimizer
* [Option 1] Configure all supported framework at the same time
* Go to Model Optimizer prerequisites directory:

*cd /opt/intel/openvino/deployment\_tools/model\_optimizer/install\_prerequisites*

* 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
* Go to Model Optimizer prerequisites directory:

*cd /opt/intel/openvino/deployment\_tools/model\_optimizer/install\_prerequisites*

* 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
* Verify Installation

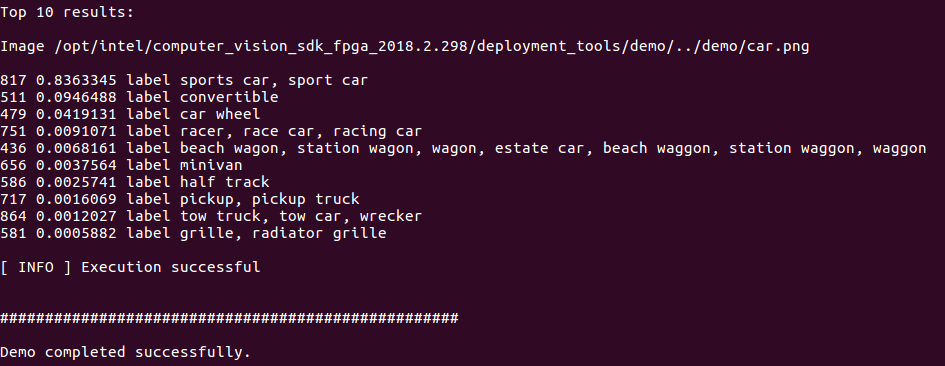
1. Go to the *Inference Engine demo* directory:

cd /opt/intel/openvino/deployment\_tools/demo

1. Run the *Image Classification verification script*:

./demo\_squeezenet\_download\_convert\_run.sh

1. 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:



* Running Inference of Pre-Trained Models

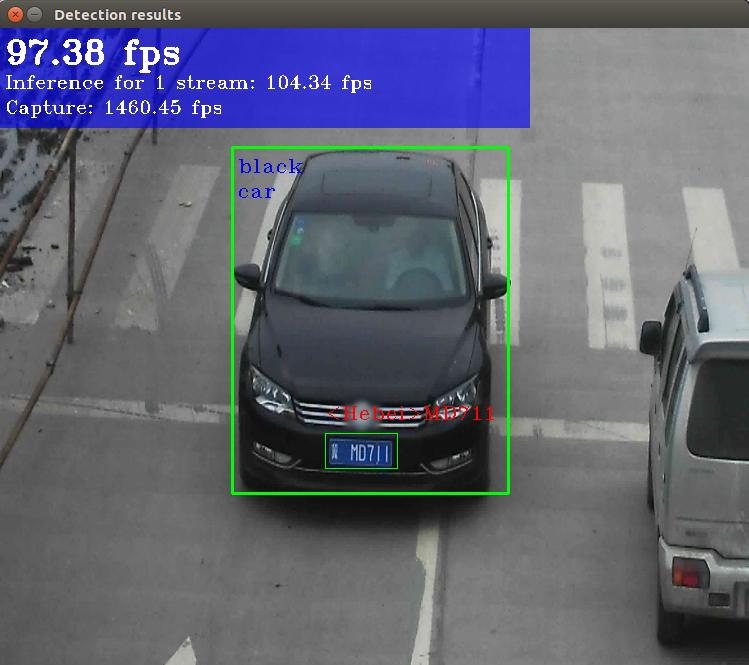
1. Run the Inference Pipeline verification script

* ./demo\_security\_barrier\_camera.sh

1. 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.

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



1. Close the image viewer window to complete the verification script.
2. For more Pre-trained models: <https://github.com/openvinotoolkit/open_model_zoo>