

OpenVINO Installation Guide

Download the Intel Distribution of OpenVINO Toolkit from Intel website:

<https://software.intel.com/content/www/us/en/develop/tools/opencvino-toolkit/download.html>

Please select the Linux, 2019 R1.144 version

The following guidance is performed on Ubuntu 18.04, 64 bit OS

•Unpack installation package

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_opencvino_toolkit_p_<version>.tgz`.

3. Unpack the .tgz file:

```
tar -xvzf l_opencvino_toolkit_p_.tgz
```

•Start installing OpenVINO

- 1.Open the terminal, enter the folder generated after decompression, and run the command to install

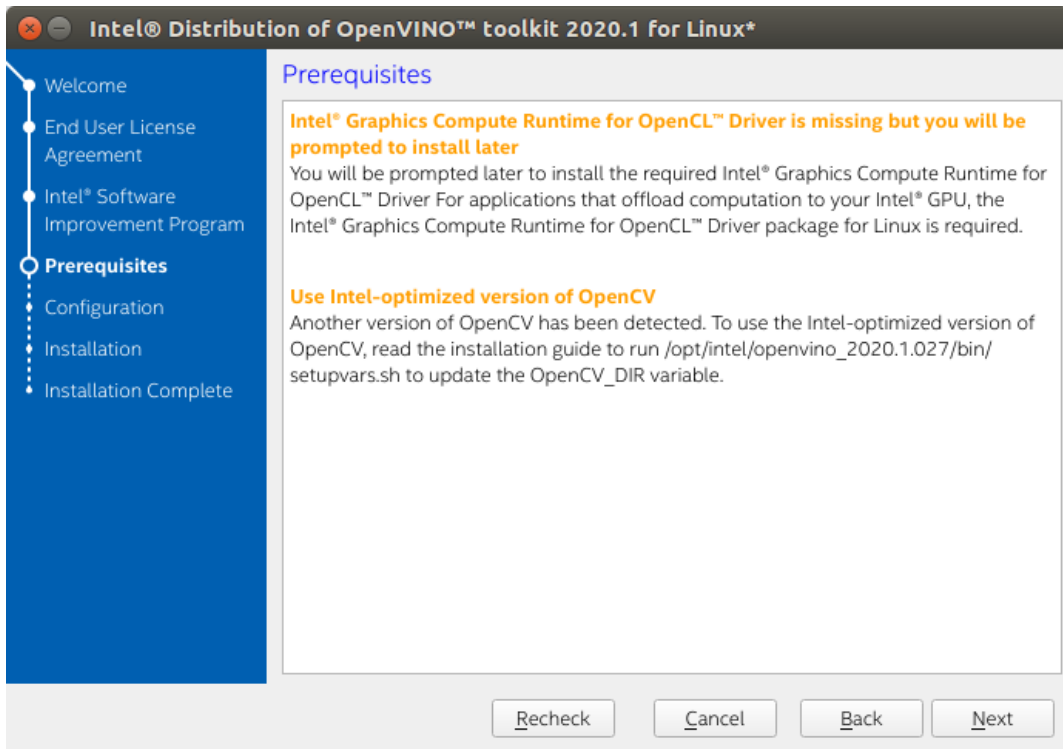
Option 1:GUI Installation Wizard:

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

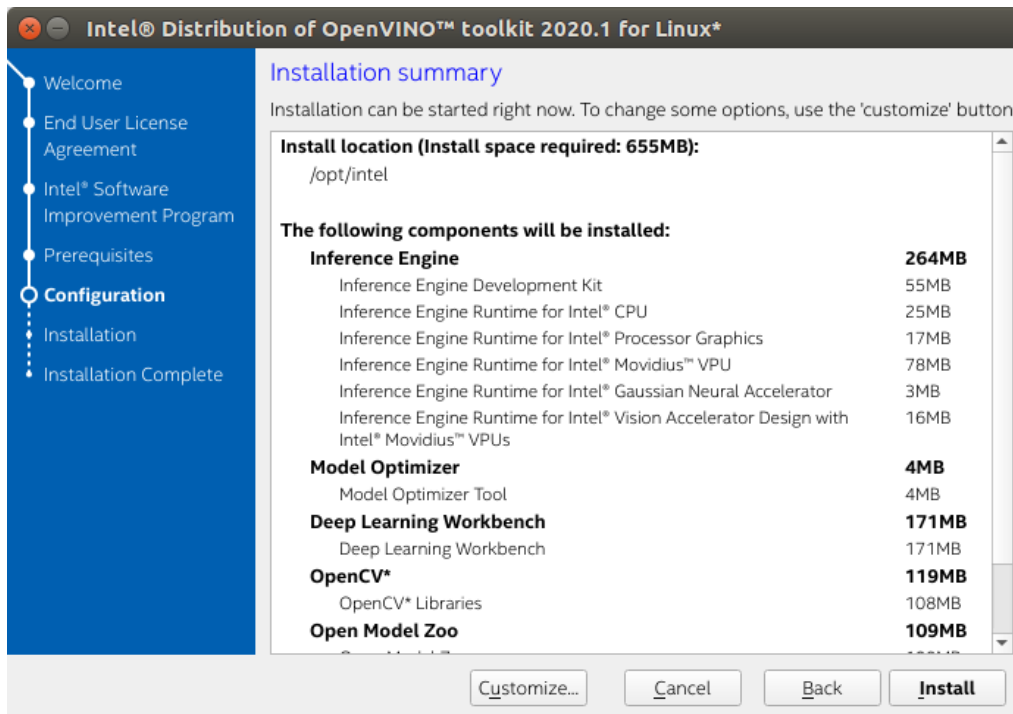
Option 2:Command-Line Instructions :

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

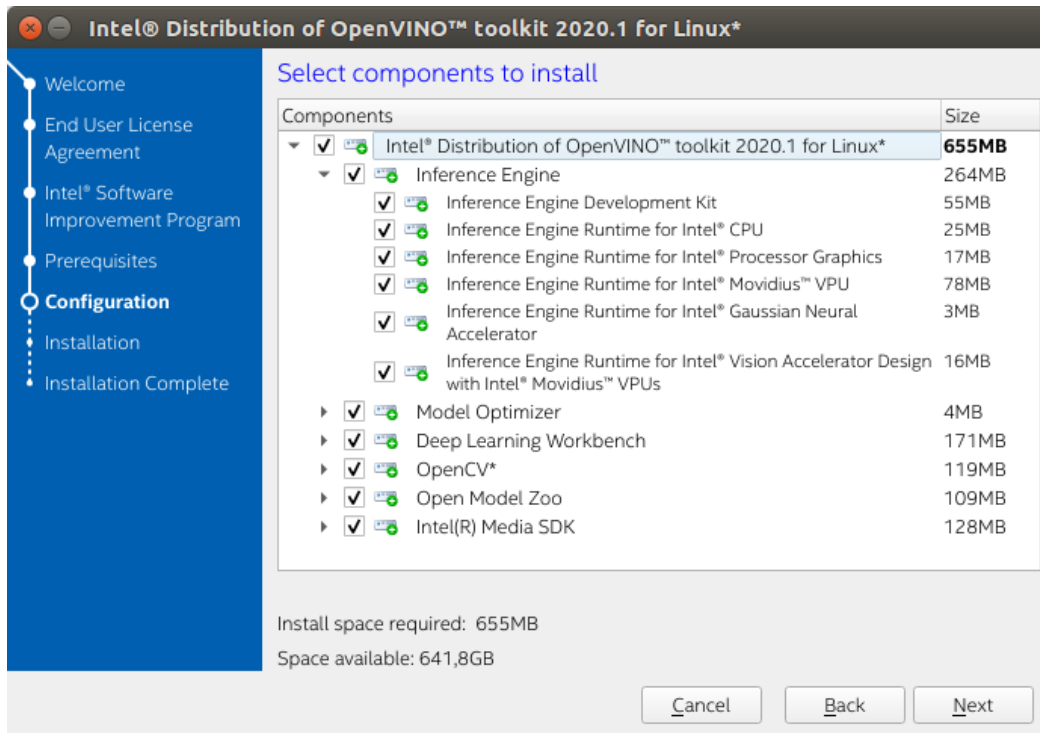
2. The following uses GUI installation as an example, which is also applicable to command line installation. Enter the installation interface:



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

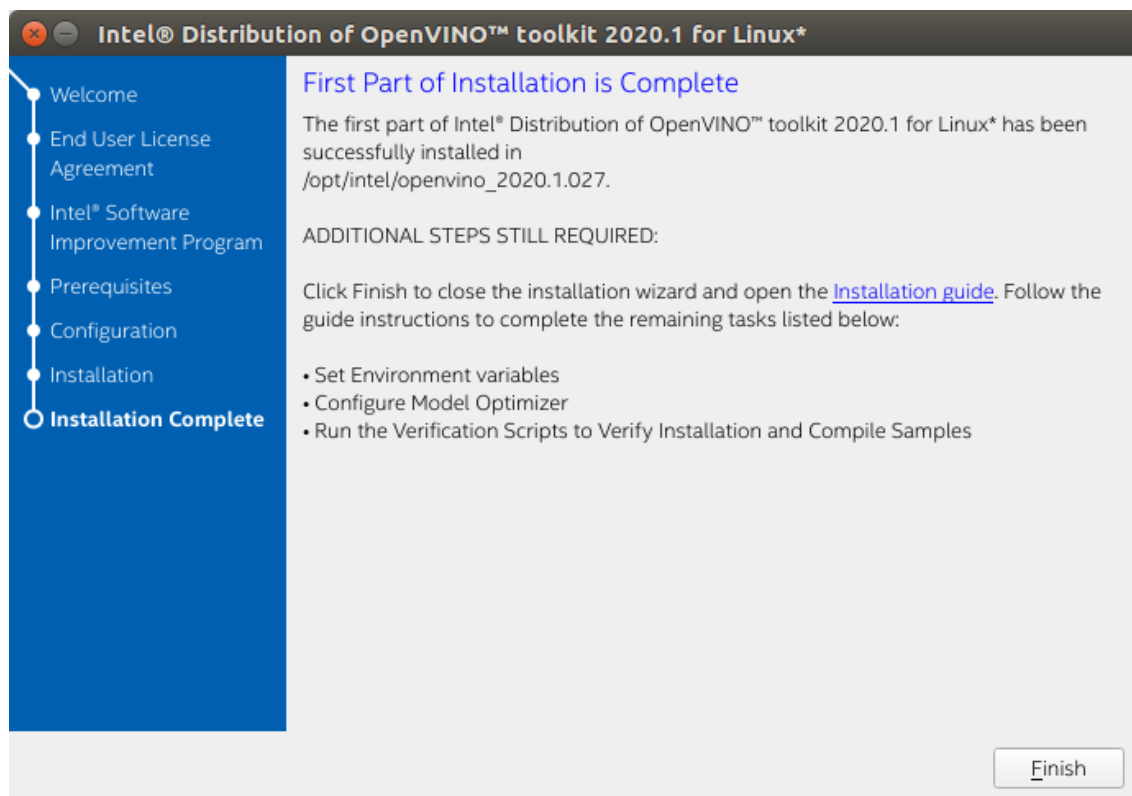


4.You can choose **Customize** to change the installation directory or the components you want to install:



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/`

5. Complete the interface, indicating that OpenVINO has been installed:



•Install External Software Dependencies

Note: If you install OpenVINO to a non-default installation directory, replace /opt/intel with the directory where the software is installed.

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
```

•Set the Environment Variables

Before compiling and running the OpenVINO application, you must update multiple environment variables.

Set the environment variables, the steps are as follows:

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

```
vi /.bashrc
```

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

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

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

4.To test your change, open a new terminal. You will see :

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

The configuration is successful.

•Configure the Model Optimizer

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
```

You can also execute the corresponding version as needed:

caffe:

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

TensorFlow:

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

MXNet:

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

ONNX:

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

Kaldi:

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

•Run the Verification Scripts to Verify Installation

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

```
cd /opt/intel/opencvino/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 0.0016969 label pickup, pickup truck
864 0.0012827 label tow truck, tow car, wrecker
581 0.0005882 label grille, radiator grille

[ INFO ] Execution successful

#####
Demo completed successfully.
```

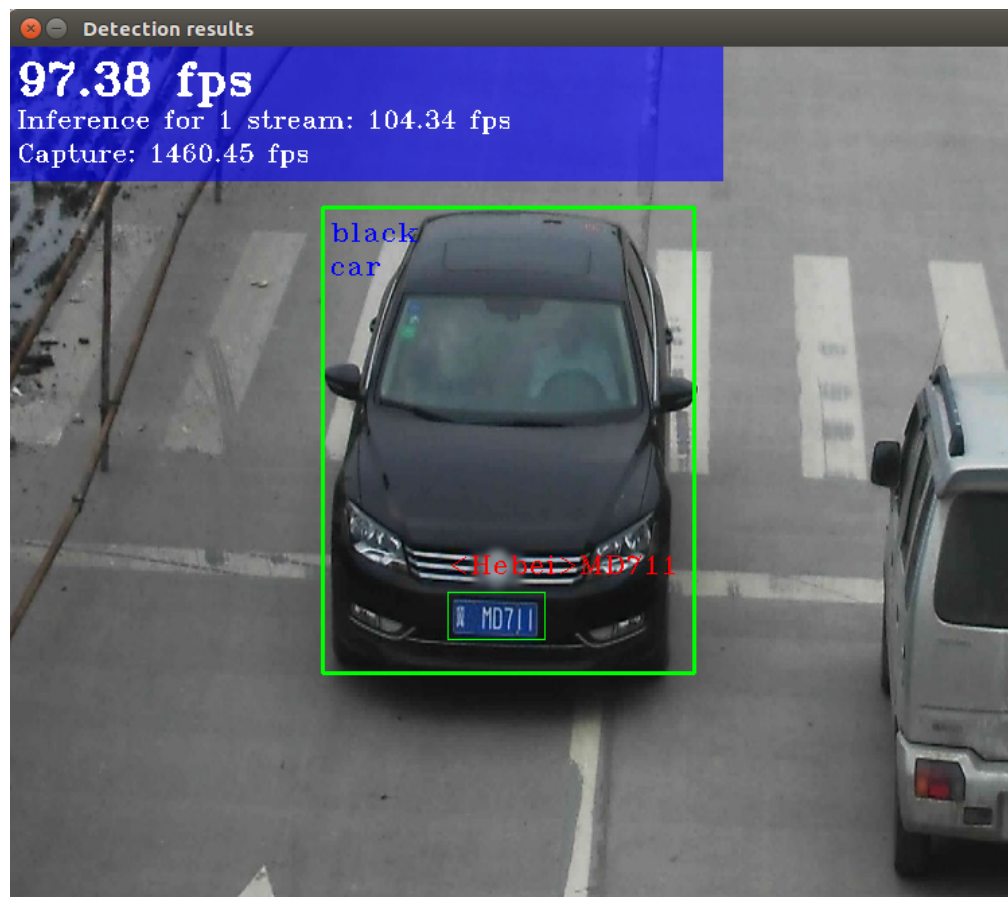
3.Run the **Inference Pipeline verification script**:

```
./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:



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

To learn about the verification scripts, see the `README.txt` file in `/opt/intel/openvino/deployment_tools/demo`.

Now, OpenVINO installation is complete!

Reference: [Intel official installation documentation](#)