Unit- 4

Objective

1. Introduction to Intel OpenVino
2. OpenVino Toolkit and components
3. Model Optimizer
4. Installing and setting up Intel OpenVIno
5. Hands on application of OpenVino

Intel OpenVino toolKit

OpenVINO stands for Open Visual Inferencing and Neural Network Optimization.

- Designed to speed up networks used in visual inferencing tasks like image classification and object detection.
- DNNs used for solving visual tasks these days are Convolutional Neural Networks (CNN).
- OpenVINO speeds up computation by first optimizing the neural network model in a hardware agnostic way using a model optimizer followed by hardware-specific acceleration accomplished using the OpenVINO Inference Engine for the particular hardware.

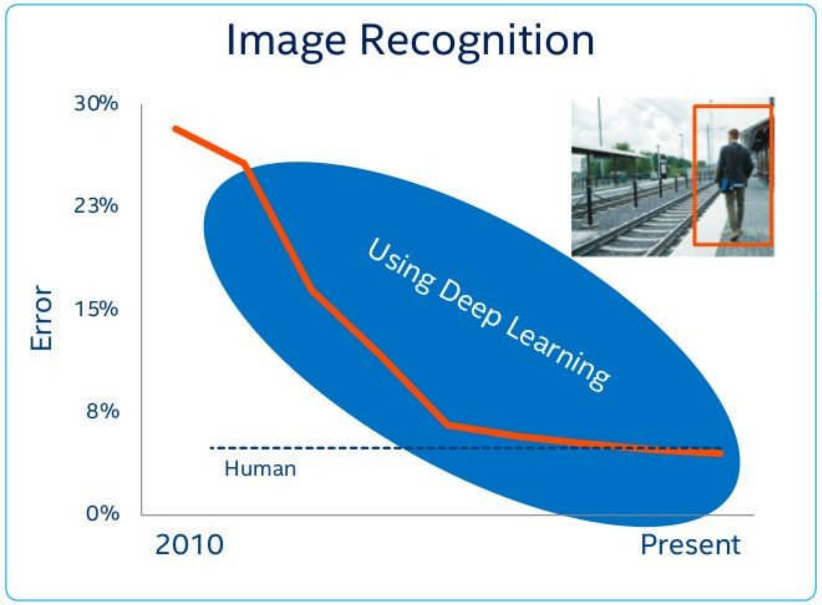
Why OpenVino

Video: The “Eye of IOT”

Use of video, computer vision, and deep learning is growing rapidly

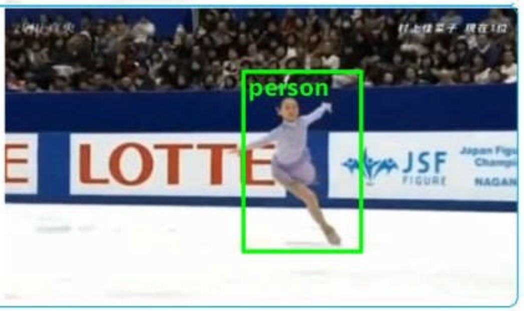
Deep Learning Usage Is Increasing

Deep learning revenue is estimated to grow from $655M in 2016 to **$35B** by 2025[1].



Traditional Computer Vision Object Detection

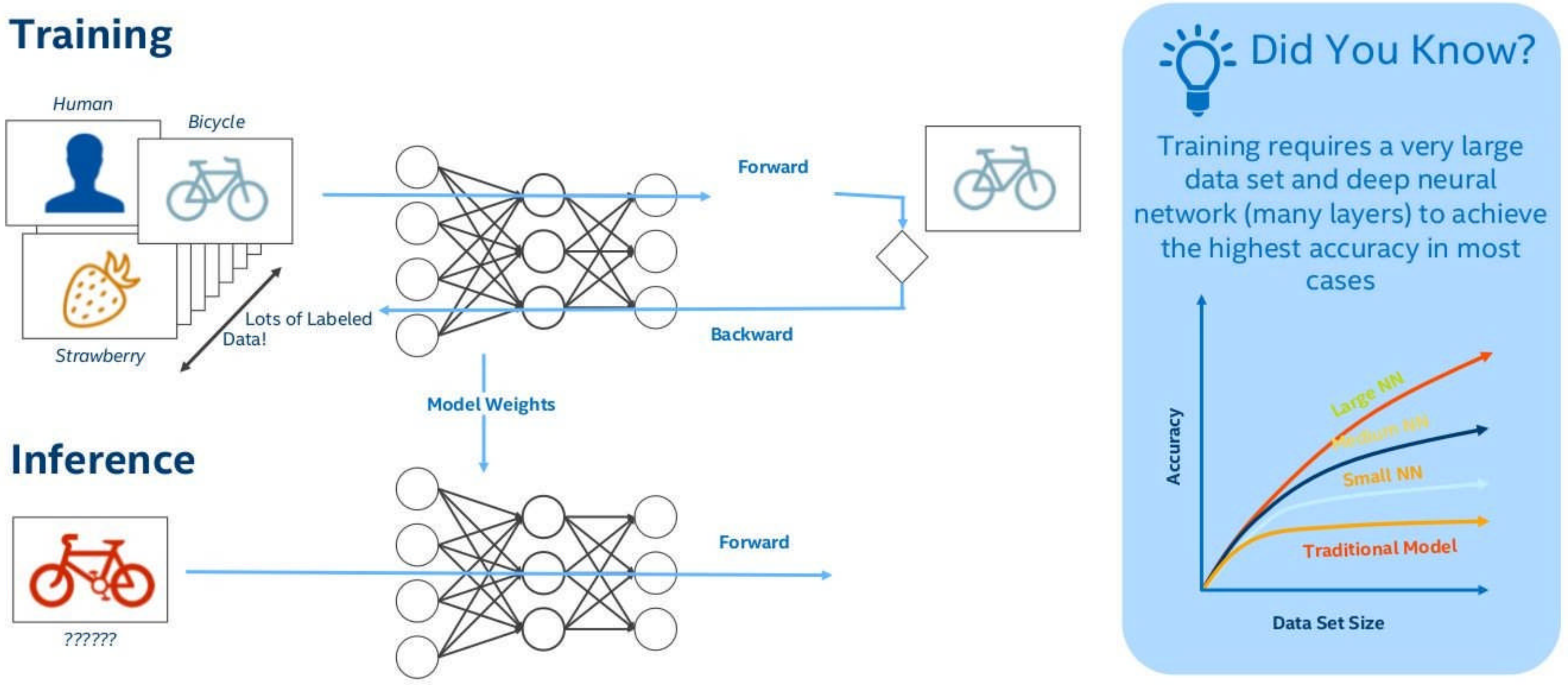
Deep Learning Computer Vision Person Recognition



Market Opportunities + Advanced Technologies Have Accelerated Deep Learning Adoption

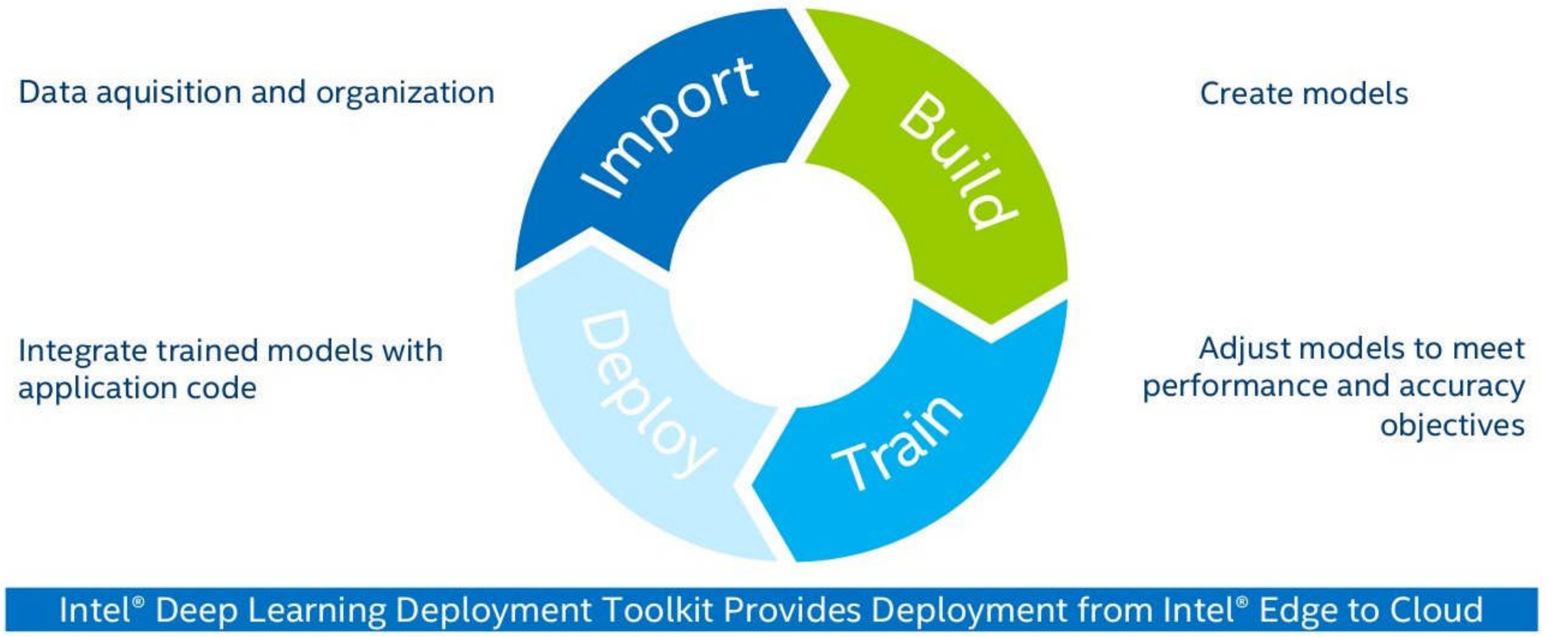
[1]Tractica* 2Q 2017

Deep Learning: Training vs. Inference



Training requires a very large data set and deep neural network (many layers) to achieve the highest accuracy in most cases

Artificial Intelligence Development Cycle


Data aquisition and organization
Import
Build
Create models
Integrate trained models with application code
Deploy
Train
Adjust models to meet performance and accuracy objectives
Intel® Deep Learning Deployment Toolkit Provides Deployment from Intel® Edge to Cloud

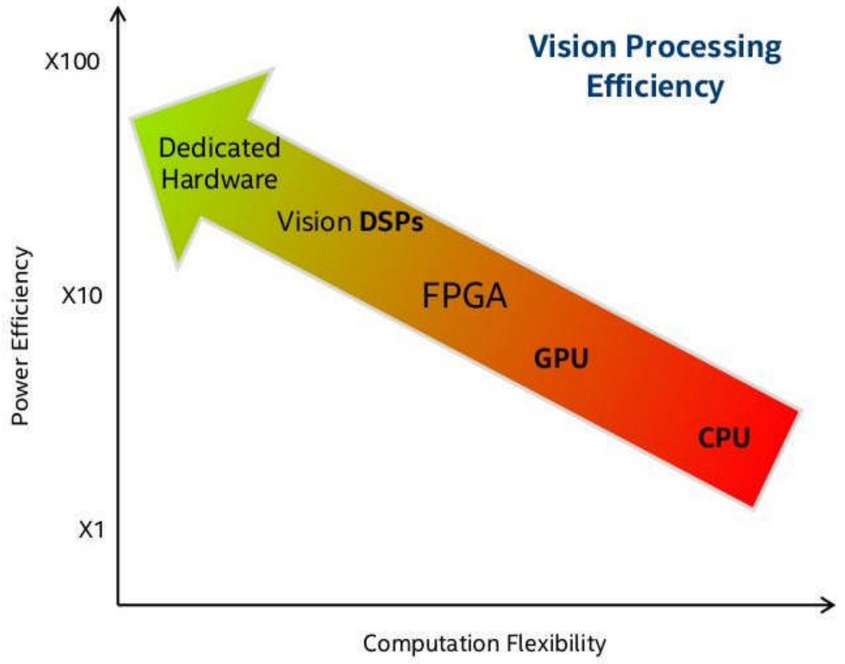
Choosing the “Right” Hardware

Power/Performance Efficiency Varies

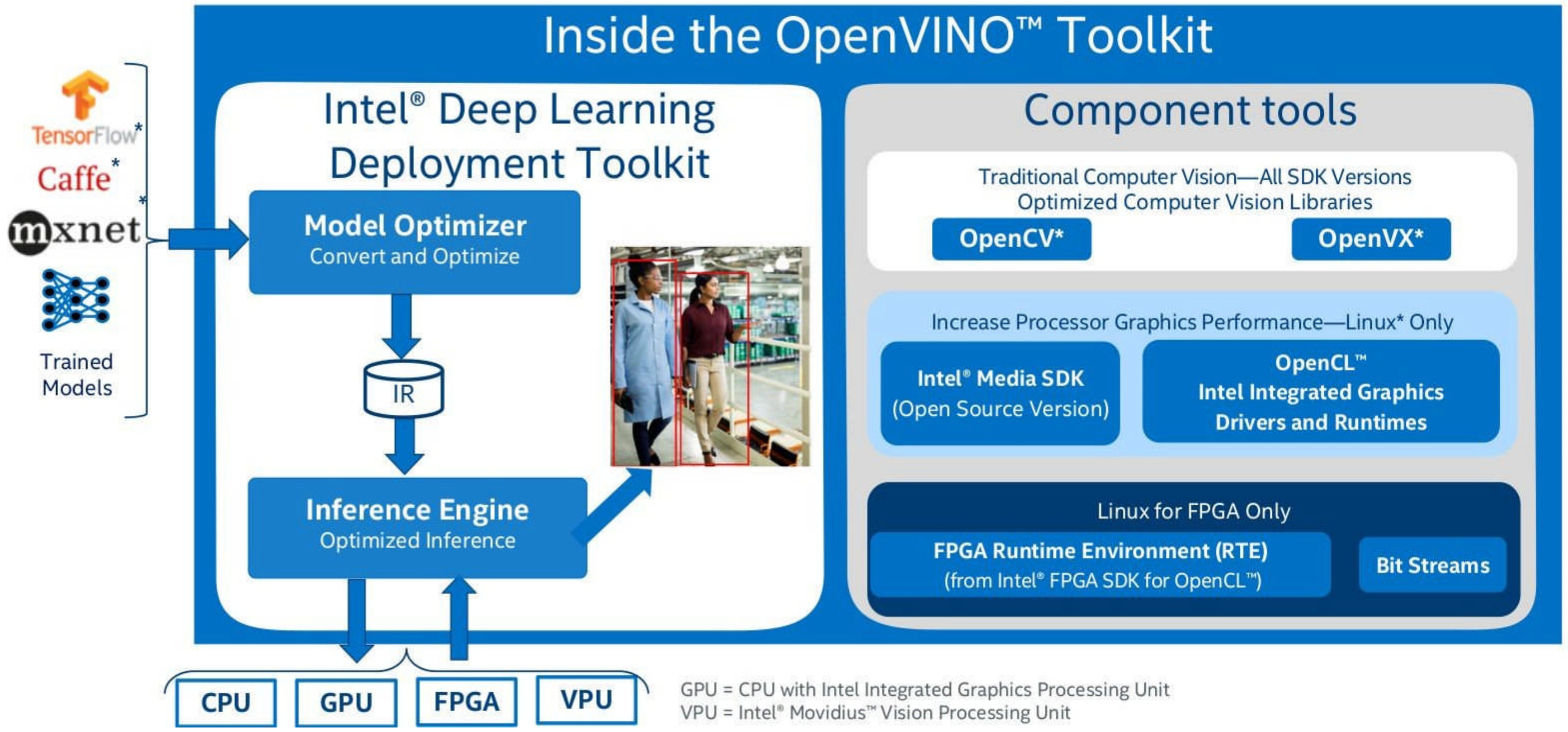
- Running the right workload on the right piece of hardware → higher efficiency
- Hardware acceleration is a must
- Heterogeneous computing?

Tradeoffs

- Power/performance
- Price
- Software flexibility, portability

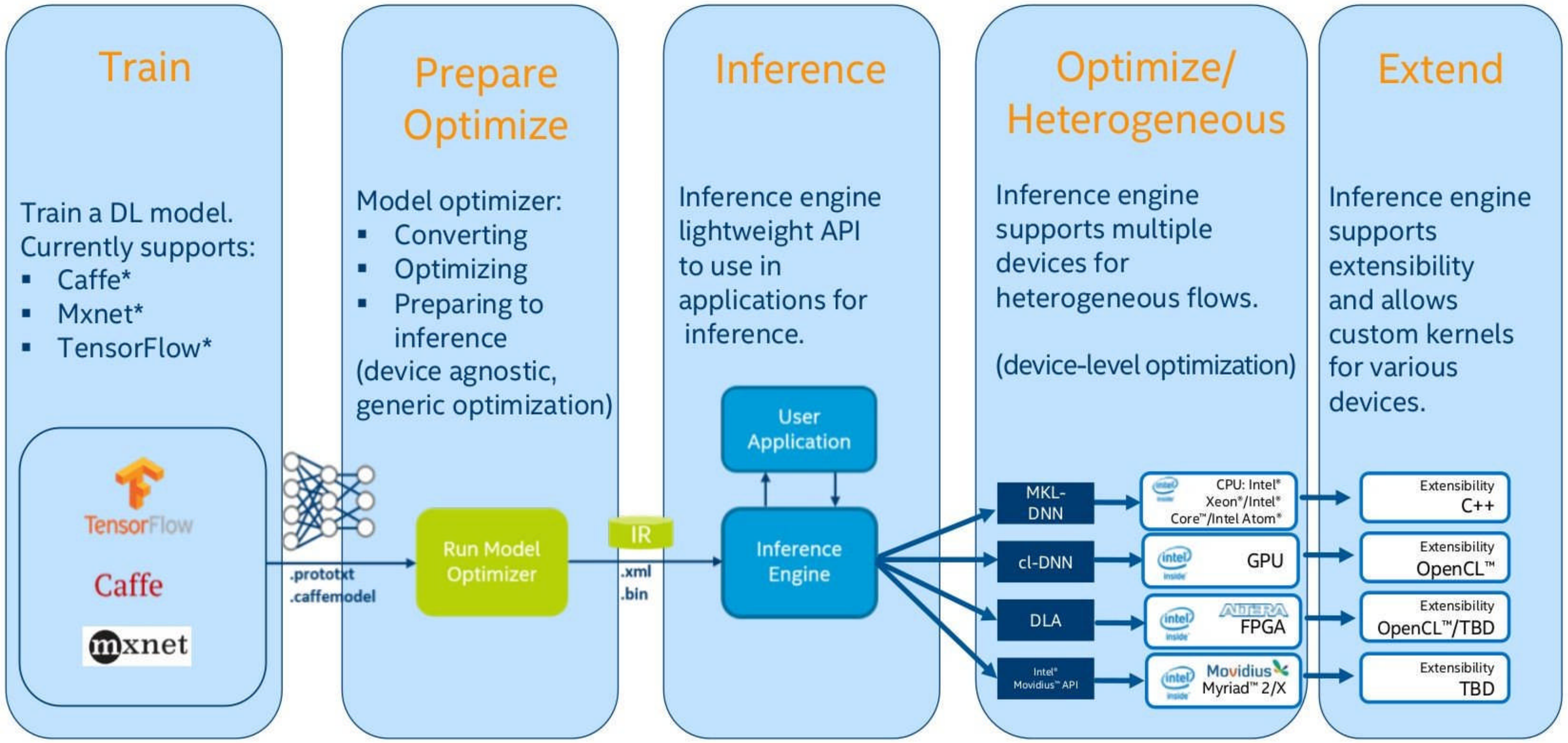


OpenVINO Toolkit and Components



Inside the OpenVINO™ Toolkit
TensorFlow*
Caffe*
mxnet*
Trained Models
Intel® Deep Learning Deployment Toolkit
Model Optimizer
Convert and Optimize
IR
Inference Engine
Optimized Inference
Component tools
Traditional Computer Vision—All SDK Versions
Optimized Computer Vision Libraries
OpenCV*
OpenVX*
Increase Processor Graphics Performance—Linux* Only
Intel® Media SDK (Open Source Version)
OpenCL™ Intel Integrated Graphics Drivers and Runtimes
Linux for FPGA Only
FPGA Runtime Environment (RTE) (from Intel® FPGA SDK for OpenCL™)
Bit Streams
CPU
GPU
FPGA
VPU
GPU = CPU with Intel Integrated Graphics Processing Unit
VPU = Intel® Movidius™ Vision Processing Unit

Computer Vision Application development - OpenVINO Toolkit



Train

Train a DL model.
Currently supports:
- Caffe*
- Mxnet*
- TensorFlow*

Prepare Optimize

Model optimizer:
- Converting
- Optimizing
- Preparing to inference

(device agnostic, generic optimization)

Inference

Inference engine lightweight API to use in applications for inference.

Optimize/ Heterogeneous

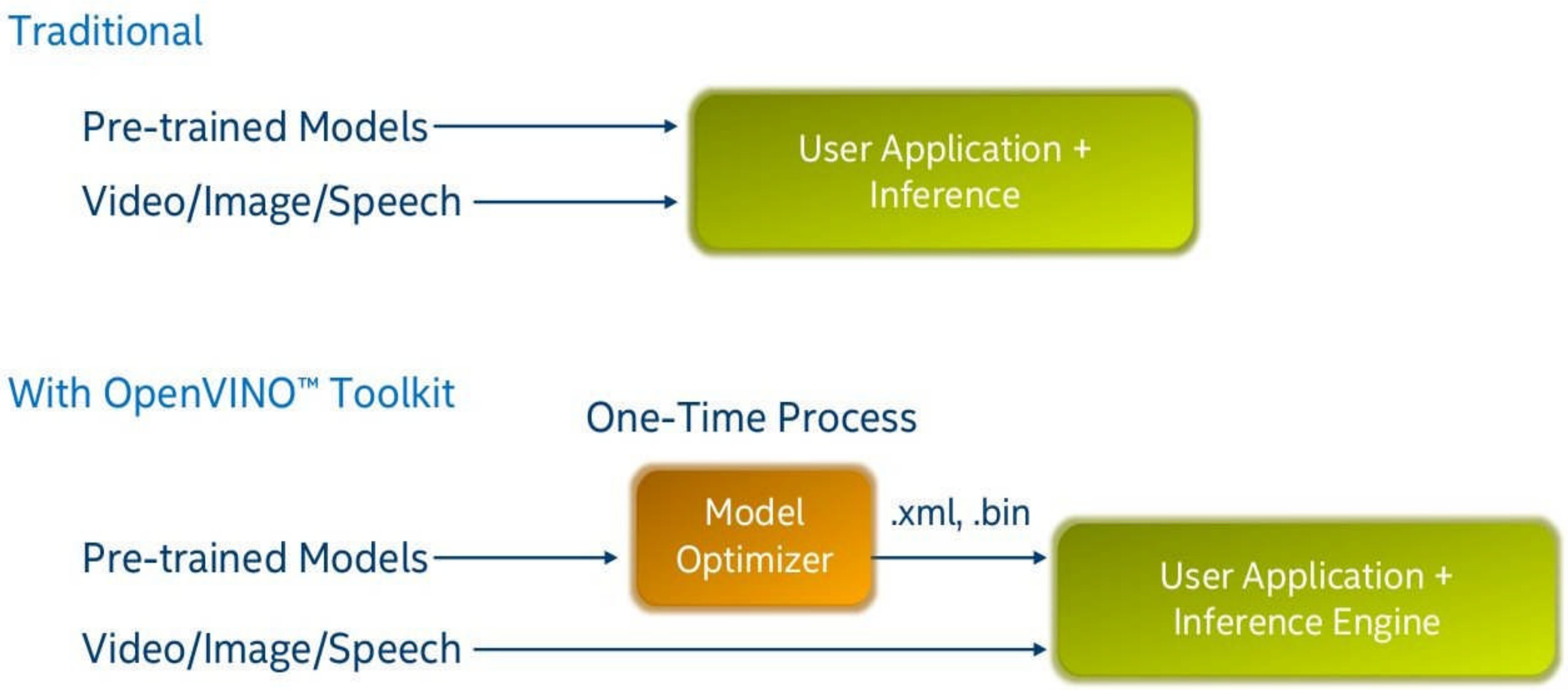
Inference engine supports multiple devices for heterogeneous flows.

(device-level optimization)

Extend

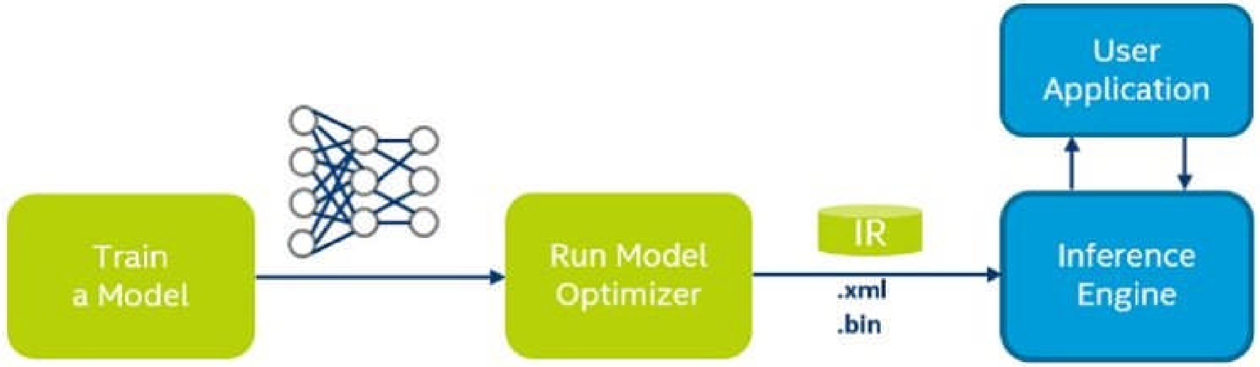
Inference engine supports extensibility and allows custom kernels for various devices.

Deep Learning Application Deployment



Model Optimizer

- The Model Optimizer is a Python*-based cross-platform command line tool for importing trained models from popular deep learning frameworks such as Caffe*, TensorFlow*, Apache MXNet*, ONNX* and Kaldi*.
- It facilitates the transition between the training and deployment environment, performs static model analysis, and adjusts deep learning models for optimal execution on end-point target devices.
- The Inference Engine API offers a unified API across a number of supported Intel® platforms.



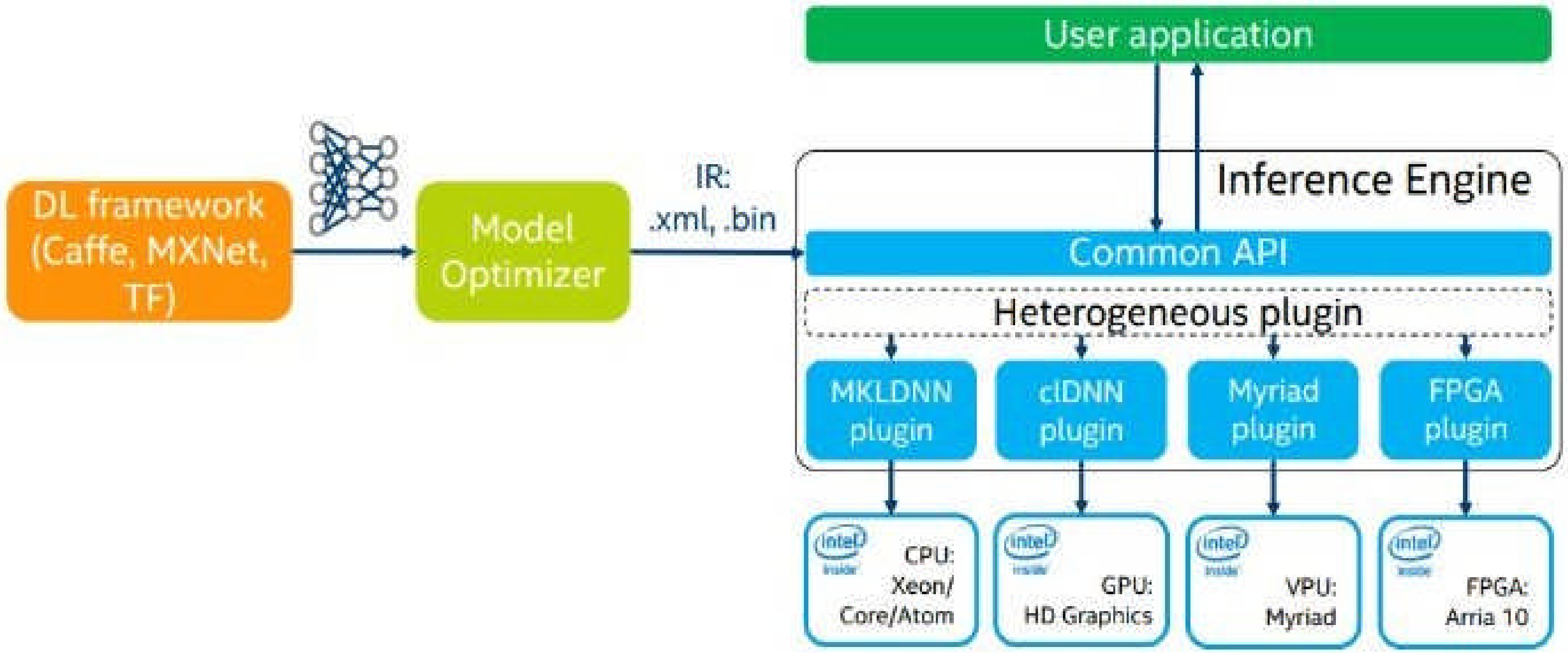
Model Optimizer

- Model Optimizer process assumes you have a network model trained using a supported deep learning framework.
- 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

Operations of Model Optimizer

- Reshaping
- Batching
- Modifying the network structure
- Standardizing and Scaling
- Quantization

OpenVINO Inference Engine : Hardware Specific Optimizations



Intel® Distribution of OpenVINO™ Toolkit

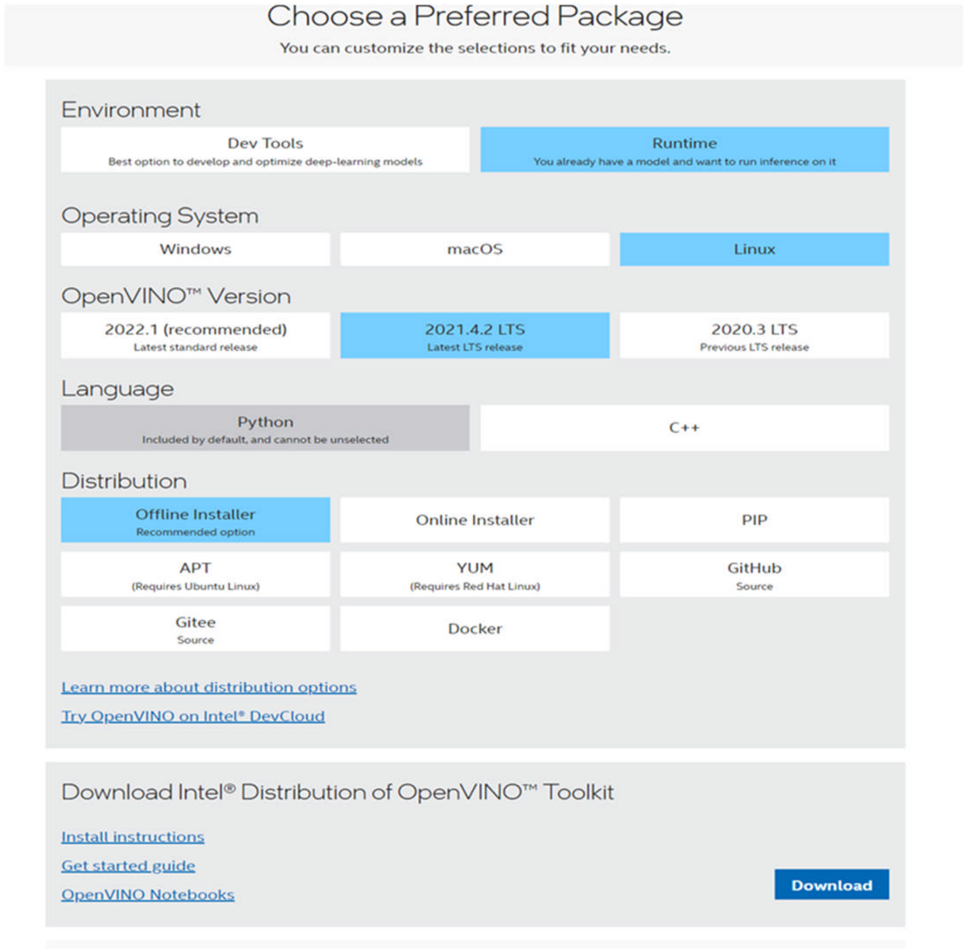
NOTE - Only proceed with the installation when you have all the pre required softwares installed on your machine.

- A Linux build environment needs these components:
  - ✓ OpenCV 3.4 or higher
  - ✓ GNU Compiler Collection (GCC)* 3.4 or higher
  - ✓ CMake* 2.8 or higher
  - ✓ Python* 3.5 or higher

Installing Intel Openvino

- Step 1: Download the Correct Version of OpenVINO Toolkit.
- Click the download button. It will download a file named

l_openvino_toolkit_p_<version>.tgz, where <version> is the version number you have chosen to download.



Choose a Preferred Package

You can customize the selections to fit your needs.

Environment

- Dev Tools — Best option to develop and optimize deep-learning models
- [✓ Runtime — You already have a model and want to run inference on it]

Operating System

- Windows
- macOS
- [✓ Linux]

OpenVINO™ Version

- 2022.1 (recommended) — Latest standard release
- [✓ 2021.4.2 LTS — Latest LTS release]
- 2020.3 LTS — Previous LTS release

Language

- [✓ Python — Included by default, and cannot be unselected]
- C++

Distribution

- [✓ Offline Installer — Recommended option]
- Online Installer
- PIP
- APT — (Requires Ubuntu Linux)
- YUM — (Requires Red Hat Linux)
- GitHub — Source
- Gitee — Source
- Docker

Learn more about distribution options

Try OpenVINO on Intel® DevCloud

Download Intel® Distribution of OpenVINO™ Toolkit

Install instructions

Get started guide

OpenVINO Notebooks

Installing Intel Openvino

Step 2. Unpack the Installer File

Open your terminal and cd into the directory where you kept the OpenVINO installer file.

Then unpack the file, using the following command:

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

After unpacking, you will see a l_openvino_toolkit_p_<version> folder in your current working directory.

This contains the OpenVINO installer and other required files

Installing Intel Openvino

Step 3. Install OpenVINO Toolkit

- Now, cd into the l_openvino_toolkit_p_<version>. You get three options to install OpenVINO on your system.

- The graphical user interface (GUI) installation wizard
- The command line installer
- The command line installer, with silent instructions.

We will use the GUI installation wizard, that is the install_GUI.sh file, as it is the easiest and most intuitive to follow. To start the installation, type the following command in your terminal.

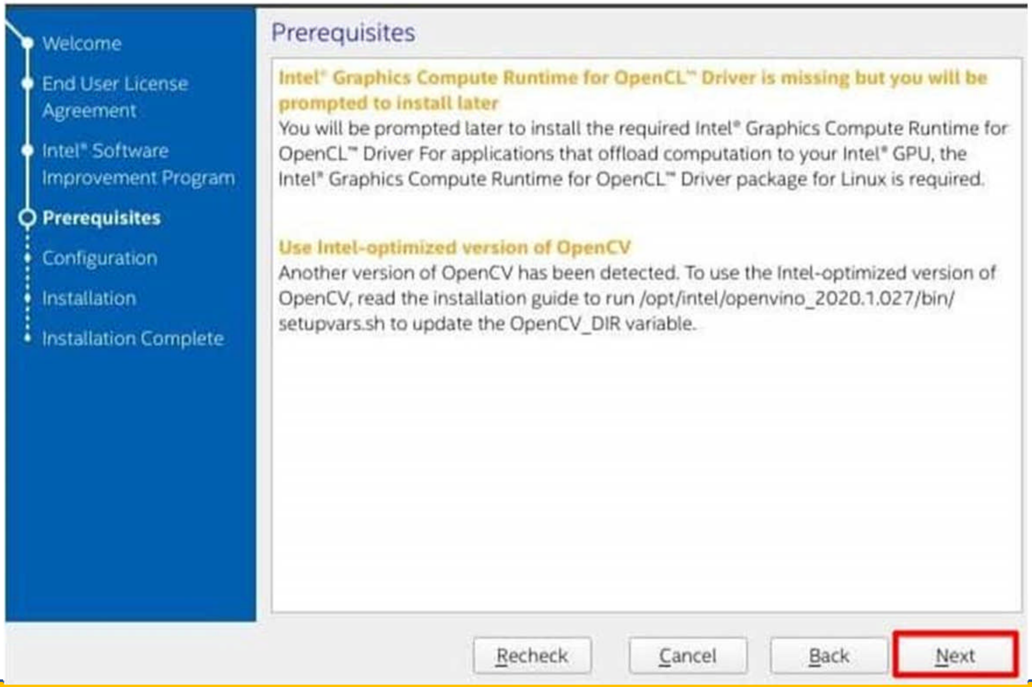
```
sudo ./install_GUI.sh
```

Note: You can also continue with the installation processes, without the sudo access.

Installing Intel Openvino

- **Following the Installation Instruction on Screen**

- First, comes the welcome screen, and then the license agreement. Next, you will see the prerequisites screen, similar to the image below.



Prerequisites

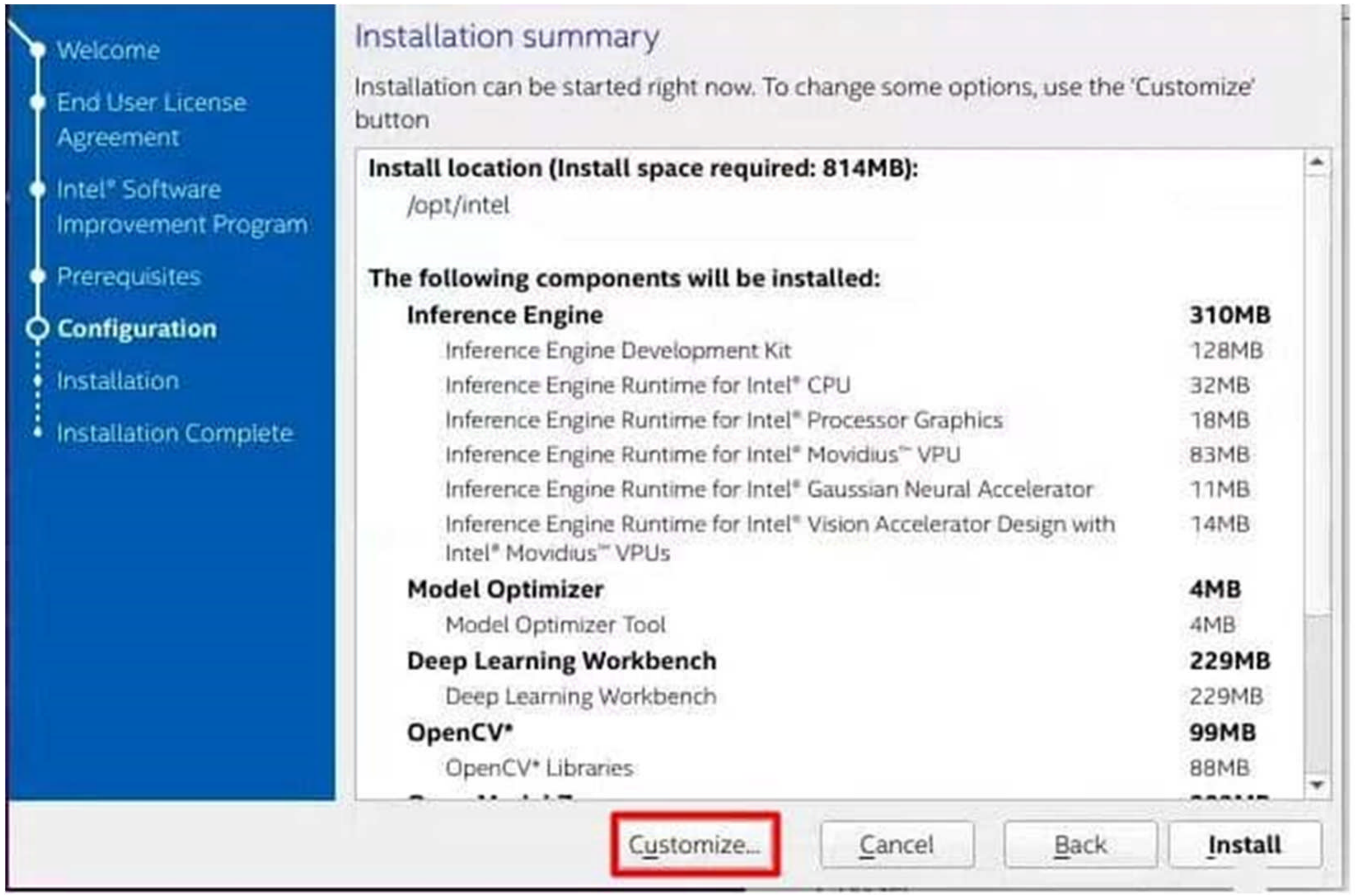
Intel® Graphics Compute Runtime for OpenCL™ Driver is missing but you will be prompted to install later

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_2020.1.027/bin/setupvars.sh to update the OpenCV_DIR variable.

Installing Intel Openvino



Installation summary

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

Install location (Install space required: 814MB):

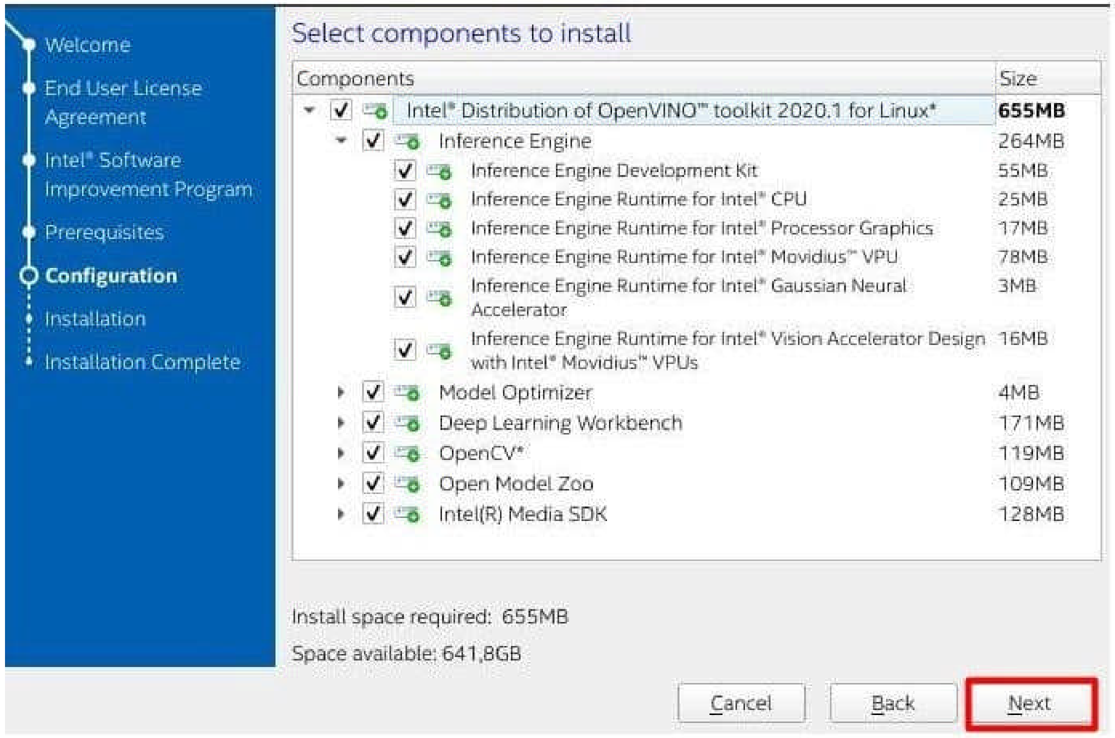
/opt/intel

The following components will be installed:

Inference Engine	**310MB**
Inference Engine Development Kit	128MB
Inference Engine Runtime for Intel® CPU	32MB
Inference Engine Runtime for Intel® Processor Graphics	18MB
Inference Engine Runtime for Intel® Movidius™ VPU	83MB
Inference Engine Runtime for Intel® Gaussian Neural Accelerator	11MB
Inference Engine Runtime for Intel® Vision Accelerator Design with Intel® Movidius™ VPUs	14MB
Model Optimizer	**4MB**
Model Optimizer Tool	4MB
Deep Learning Workbench	**229MB**
Deep Learning Workbench	229MB
OpenCV*	**99MB**
OpenCV* Libraries	88MB

Installing Intel Openvino

Configuring the Intel OpenVINO installation. If you wish, you can also customize the configurations, and choose what you want to install.



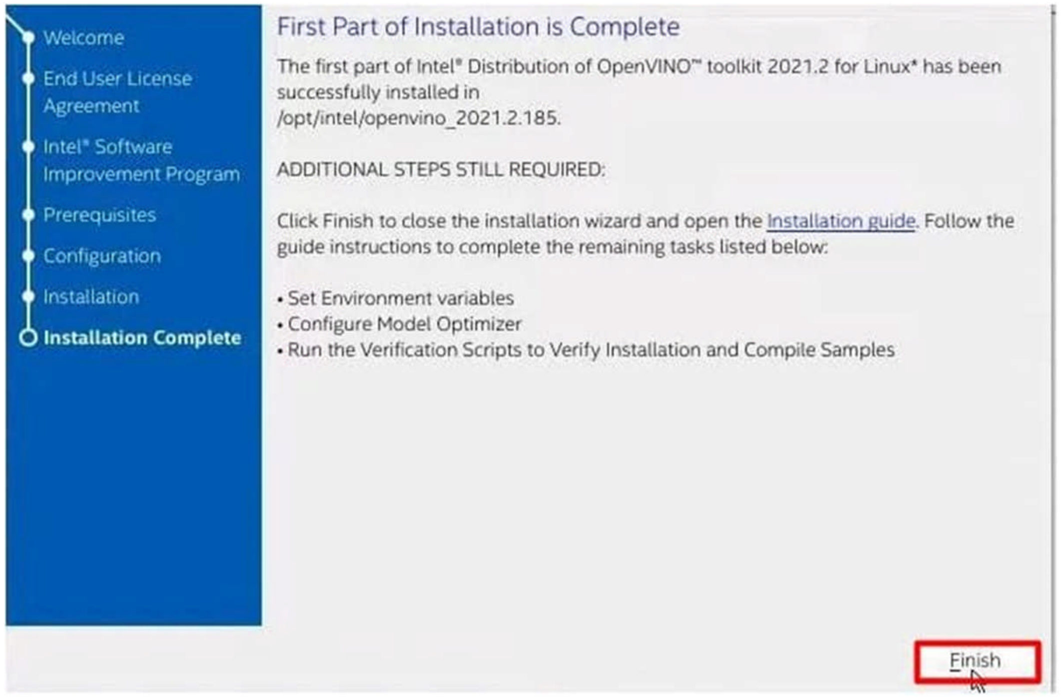
Select components to install

Components	Size
✓ Intel® Distribution of OpenVINO™ toolkit 2020.1 for Linux*	**655MB**
✓ Inference Engine	264MB
✓ Inference Engine Development Kit	55MB
✓ Inference Engine Runtime for Intel® CPU	25MB
✓ Inference Engine Runtime for Intel® Processor Graphics	17MB
✓ Inference Engine Runtime for Intel® Movidius™ VPU	78MB
✓ Inference Engine Runtime for Intel® Gaussian Neural Accelerator	3MB
✓ Inference Engine Runtime for Intel® Vision Accelerator Design with Intel® Movidius™ VPUs	16MB
✓ Model Optimizer	4MB
✓ Deep Learning Workbench	171MB
✓ OpenCV*	119MB
✓ Open Model Zoo	109MB
✓ Intel(R) Media SDK	128MB

Install space required: 655MB

Space available: 641,8GB

Installing Intel Openvino



First Part of Installation is Complete

The first part of Intel® Distribution of OpenVINO™ toolkit 2021.2 for Linux* has been successfully installed in /opt/intel/openvino_2021.2.185.

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

Installing Intel Openvino

Step 5: Installing Software Dependencies

These include software dependencies for:

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

Change your current working directory to the install_dependencies folder.

```
cd /opt/intel/openvino_2021/install_dependencies
```

Run the following script to install the dependencies.

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

Installing Intel Openvino

- **Step 6: Set Up the Environment Variables**
- To set the environment variables, open a new terminal and type:

```
vi ~/.bashrc
```

- Go to the end of the file, and add the following line:

```
source /opt/intel/openvino_2021/bin/setupvars.sh
```

Now, save and close the file. Shut down your current terminal and open a new one so that system-wide changes can take place. We're only one step away from completing installation now. Just configure the Model Optimizer and you're done.

Installing Intel Openvino

- **Step 7: Installing Model Optimizer Prerequisites**
- The Model Optimizer is a command-line tool in the OpenVINO Toolkit.
- The OpenVINO Toolkit does not support inference directly. None of the models trained with Deep-Learning frameworks like TensorFlow, Caffe, MXNet, ONNX, or even Kaldi can help you infer. First, you'll need to convert these trained models into an Intermediate Representation (IR), which consists of:

- A .xml file, which describes the network architecture
- A .bin file that holds the weights and biases of the trained model

Configure the Model Optimizer

Configure all supported frameworks at the same time

- Go to the 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

NOTE - You can also configure the frameworks individually by running

sudo ./install_prerequisites_<framework>.sh

eg: For TensorFlow: sudo ./install_prerequisites_tf.sh

Run the Verification Scripts to Verify Installation

1. Go to the Inference Engine demo directory:

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

2. Run the Image Classification verification script: **use -h to see options at the end**

   ./demo_squeezenet_download_convert_run.sh -d cpu

3. Run the Inference Pipeline verification script:

   ./demo_security_barrier_camera.sh

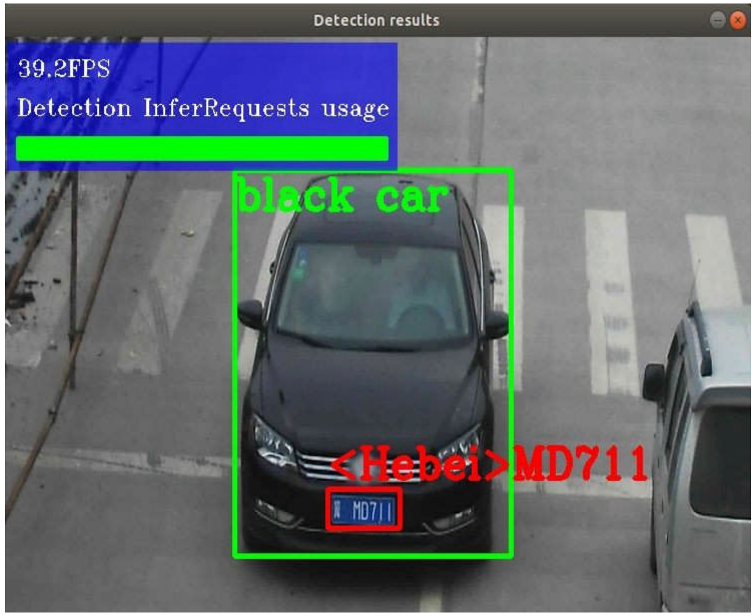
Close the image window viewer to complete the verification script.

Image Classification verification script:

- 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:

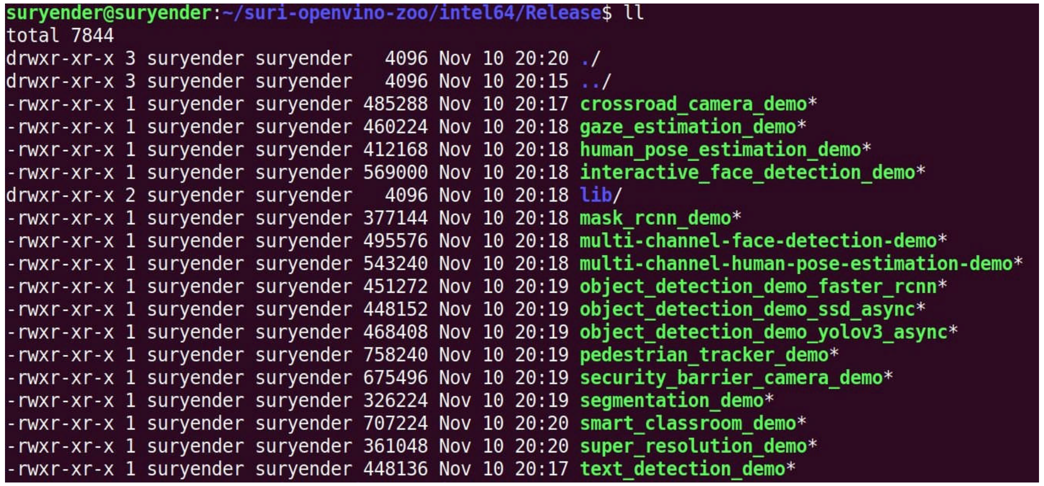
Inference Pipeline verification script:

- 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.
- 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.



Building other demos

- **Many downloaded models are available in the following directory**
  - /opt/intel/openvino/deployment_tools/open_model_zoo/demos
- **Create a directory in home folder and build the demos**
  - mkdir demo-zoo
  - cd demo-zoo
  - cmake /opt/intel/openvino/deployment_tools/open_model_zoo/demo
  - make



```
suryender@suryender:~/suri-openvino-zoo/intel64/Release$ ll
total 7844
drwxr-xr-x 3 suryender suryender   4096 Nov 10 20:20 ./
drwxr-xr-x 3 suryender suryender   4096 Nov 10 20:15 ../
-rwxr-xr-x 1 suryender suryender 485288 Nov 10 20:17 crossroad_camera_demo*
-rwxr-xr-x 1 suryender suryender 460224 Nov 10 20:18 gaze_estimation_demo*
-rwxr-xr-x 1 suryender suryender 412168 Nov 10 20:18 human_pose_estimation_demo*
-rwxr-xr-x 1 suryender suryender 569000 Nov 10 20:18 interactive_face_detection_demo*
drwxr-xr-x 2 suryender suryender   4096 Nov 10 20:18 lib/
-rwxr-xr-x 1 suryender suryender 377144 Nov 10 20:18 mask_rcnn_demo*
-rwxr-xr-x 1 suryender suryender 495576 Nov 10 20:18 multi-channel-face-detection-demo*
-rwxr-xr-x 1 suryender suryender 543240 Nov 10 20:18 multi-channel-human-pose-estimation-demo*
-rwxr-xr-x 1 suryender suryender 451272 Nov 10 20:19 object_detection_demo_faster_rcnn*
-rwxr-xr-x 1 suryender suryender 448152 Nov 10 20:19 object_detection_demo_ssd_async*
-rwxr-xr-x 1 suryender suryender 468408 Nov 10 20:19 object_detection_demo_yolov3_async*
-rwxr-xr-x 1 suryender suryender 758240 Nov 10 20:19 pedestrian_tracker_demo*
-rwxr-xr-x 1 suryender suryender 675496 Nov 10 20:19 security_barrier_camera_demo*
-rwxr-xr-x 1 suryender suryender 326224 Nov 10 20:19 segmentation_demo*
-rwxr-xr-x 1 suryender suryender 707224 Nov 10 20:20 smart_classroom_demo*
-rwxr-xr-x 1 suryender suryender 361048 Nov 10 20:20 super_resolution_demo*
-rwxr-xr-x 1 suryender suryender 448136 Nov 10 20:17 text_detection_demo*
```

Model Download Guide

- Pre-requisites
  - Install python3 (version 3.5.2 or higher)
  - Install yaml and requests modules with the command
    - **sudo -E pip3 install pyyaml requests**
- **cd /opt/intel/openvino/deployment_tools/model_downloader**
- **./downloader.py -h** - to see the help message
- **./downloader.py --print_all** - to see all the available topologies
- **./downloader.py --name <topology_name>** - to download <topology name> network

- NOTE - Expected free space to download all the topologies with the default configuration file is around 4.3 GB

Model Optimizer Guide

- Configure your model optimizer (if you have not done that already) for different frameworks by executing 'install_prerequisites.sh' present in
  - /opt/intel/openvino/deployment_tools/model_optimizer/install_prerequisites
- **cd /opt/intel/openvino/deployment_tools/model_optimizer**
- **python3 mo.py --input_model <INPUT_MODEL>** - to optimize the
  - <INPUT MODEL>
- For example, to optimize the alexnet model based on caffe framework, execute
  - **python3 mo.py --input_model alexnet.caffemodel**
- As a result of executing the above command, two files - 'alexnet.xml' and 'alexnet.bin' will be created in your working directory.
- Download the model from internet separately

Model Optimizer Guide

Converting a caffe model: A caffe model has 2 associated files,

1 **.prototxt** — The definition of CNN goes in here. This file defines the layers in the neural network, each layer's inputs, outputs and functionality.

2 **.caffemodel** — This contains the information of the trained neural network (trained model). download both the files and use model optimizer to convert:

- **https://github.com/BVLC/caffe/wiki/Model-Zoo**
- **python3 mo.py --input_model** /home/suryender/Downloads/age_net.caffemodel
- **--input_proto** /home/suryender/Downloads/deploy_age.prototxt **--output_dir**

.

Face Detection using OpenVINO on R-PI

1. Clone the github repository into local system
   - **git clone https://github.com/ameer-aiml/face-detect-ov-rpi**
   - This repository contains both xml and bin file of Face Detection model.
2. Go to Face Detection folder
   - **cd face-detect-ov-rpi /Face Detection**

- **Note: Add a single before and after Face Detection wording if not recognized**

3. Run the python file (it should be python3)
   - **python3 face_detection.py**

References

1. "Release Notes for Intel Distribution of OpenVINO toolkit 2022". March 2022.
2. "OpenVINO Toolkit: Welcome to OpenVINO".
3. "Introduction to Intel Deep Learning Deployment Toolkit – OpenVINO Toolkit".
4. Wilbur, Marcia. "Use the Model Downloader and Model Optimizer for the Intel® Distribution of OpenVINO™ Toolkit on Raspberry Pi*"