

FLIK

Getting Started Guide



FPGA

Contents

Chapter 1	<i>Overview</i>	3
1.1	General Description	3
Chapter 2	<i>Install Thunderbolt3 driver</i>	4
2.1	Hardware Requirement	4
2.2	Install Thunderbolt 3 Driver	5
Chapter 3	<i>Install Intel Acceleration Stack</i>	9
Chapter 4	<i>Install Intel Distribution of OpenVINO toolkit</i>	12
Chapter 5	<i>Install and verify the Terasic Demonstration of the FLIK</i> ..	15
5.1	Setup.....	15
5.2	Other Demonstrations	20
Chapter 6	<i>Factory Recovery</i>	30
Chapter 7	<i>Additional Information</i>	33



Chapter 1

Overview

This chapter provides an overview of this guide.

1.1 General Description

This document will describe how to quickly run a face recognition demo using FLIK in a Linux environment. The content will include the driver, software that needs to be installed. It allows users to quickly experience the acceleration ability of deep learning inference brought by FPGA hardware platform:

1. Install Thunderbolt3 driver
2. Install Intel® PAC and the Intel® Programmable Acceleration Card Stack
3. Install Intel® Distribution of OpenVINO™ toolkit for Linux* with FPGA Support
4. Install and verify the Terasic FLIK demo package

Before you start using the FLIK, users need to prepare some Development and Target Platform. To use FLIK with Linux, the required PC hardware and Linux Operating System can be obtained from the link below.

https://docs.openvinotoolkit.org/latest/docs_install_guides_installing_openvino_linux.html

It is important to note that because the transmission and communication interface between FLIK and host PC is **Thunderbolt 3**. User's PC must have Thunderbolt 3 interface to connect with FLIK.

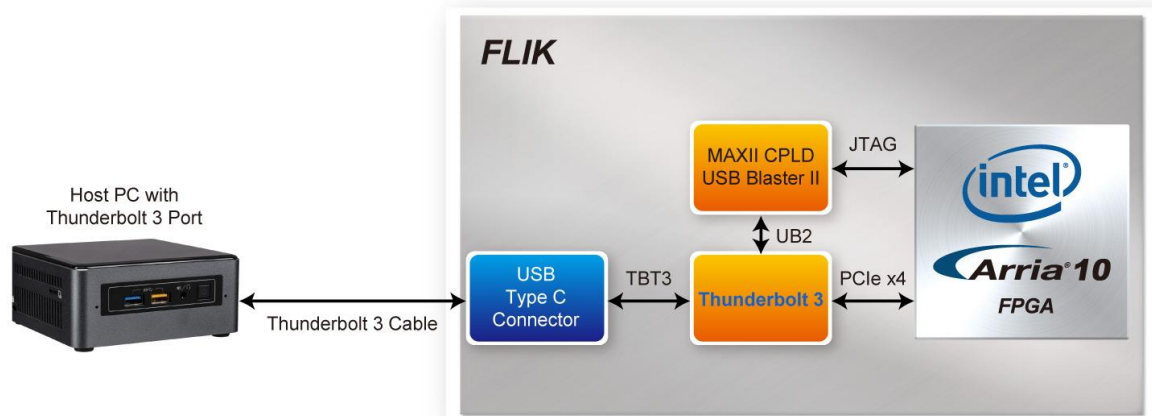
In addition, in this document, the Linux distribution we use is Ubuntu 16.04.6 LTS (64-bit). Note, for many Linux distributions of Ubuntu 16.04, Ubuntu 16.04.6 can install Thunderbolt3 driver normally, which is the ultimate system requirement for using FLIK.

Chapter 2

Install Thunderbolt3 driver

Thunderbolt 3 is unique communication path between the Host PC and FLIK. As shown in below, the FLIK use thunderbolt3 port to connect the PCIe and USB Blaster II interface to the FPGA. For the OpenCL and OpenVINO applications with the Intel FPGA, the PCIe bus is the main interface for communicating and transferring data with the host PC. The thunderbolt3 port on the FLIK allows the user to build such connection for the FPGA and the host PC via a thunderbolt3 cable. In addition, the thunderbolt3 on the FLIK also provides an USB interface to connect to the USB Blaster II circuit on the board, providing a JTAG interface for the user to program the FPGA or configuration device through the host PC.

Therefore, a host PC equipped with thunderbolt 3 port is required to work with the FLIK. This chapter will show user how to setup thunderbolt 3 connection between the Host PC and the FLIK for the first time.



2.1 Hardware Requirement

- A Host PC with Thunderbolt 3 Port is required to perform FLIK function tests. The PC should contain the features in below:

- Built-in Thunderbolt 3 Port or with Thunderbolt3 Card Installed.
- Linux Installed(In this document, we recommend you to install Ubuntu 16.04.6 LTS (64-bit))
- A FLIK kit
- A Thunderbolt 3 Cable as shown in below.



2.2 Install Thunderbolt 3 Driver

1. Download the Thunderbolt user-space components in the link below and save it to your computer:

<https://github.com/intel/thunderbolt-software-user-space>

2. You also need a C++ compiler with C++14 support. You can install the dependencies with the following command:

```
sudo apt-get install cmake libboost-filesystem-dev txt2tags pkg-config
```

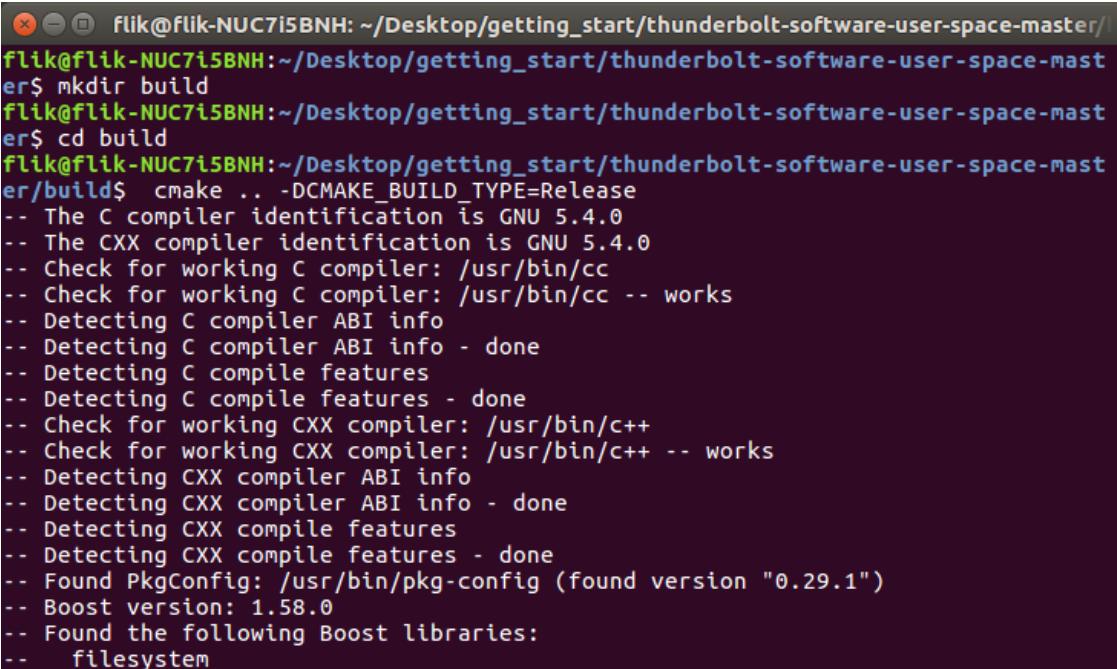
```
flik@flik-NUC7i5BNH: ~/Desktop/getting_start/thunderbolt-software-user-space-master
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

flik@flik-NUC7i5BNH:~/Desktop/getting_start/thunderbolt-software-user-space-master$ sudo apt-get install cmake libboost-filesystem-dev txt2tags pkg-config
```

3. Unzip the downloaded file **thunderbolt-software-user-space-master.zip**

4. Open terminal and locate the unzipped directory.
5. Execute the following command to install Thunderbolt 3 driver:

- `mkdir build`
- `cd build`
- `cmake .. -DCMAKE_BUILD_TYPE=Release`
- `cmake --build .`
- `sudo cmake --build . --target install`



```
flik@flik-NUC7i5BNH: ~/Desktop/getting_start/thunderbolt-software-user-space-master/
flik@flik-NUC7i5BNH:~/Desktop/getting_start/thunderbolt-software-user-space-master$ mkdir build
flik@flik-NUC7i5BNH:~/Desktop/getting_start/thunderbolt-software-user-space-master$ cd build
flik@flik-NUC7i5BNH:~/Desktop/getting_start/thunderbolt-software-user-space-master/build$ cmake .. -DCMAKE_BUILD_TYPE=Release
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found PkgConfig: /usr/bin/pkg-config (found version "0.29.1")
-- Boost version: 1.58.0
-- Found the following Boost libraries:
--   filesystem
```

6. Next, verify whether the Host PC has detected the PCIe device after connecting to the FLIK via the Thunderbolt 3 cable.
7. Plug the DC 12V power adapter to the FLIK to power on the FLIK.



8. Connect the Host PC to the FLIK with a Thunderbolt3 cable.



9. Approve all currently connected Thunderbolt devices that aren't authorized yet and (if --once wasn't specified) add them to PC:

```
sudo tbtadm approve-all
```

```

flik@flik-NUC7i5BNH: ~/Desktop/getting_start/thunderbolt-software-user-space-master
flik@flik-NUC7i5BNH:~/Desktop/getting_start/thunderbolt-software-user-space-master$ sudo tbtadm approve-all
[sudo] password for flik:
Found domain "/sys/bus/thunderbolt/devices/domain0"
Found child "/sys/bus/thunderbolt/devices/domain0/0-0/0-1"
Authorizing "/sys/bus/thunderbolt/devices/domain0/0-0/0-1"
Already authorized
flik@flik-NUC7i5BNH:~/Desktop/getting_start/thunderbolt-software-user-space-master$

```

10. Execute the following command to verify whether the host PC has detected the PCIe device :

```
lspci | grep 09c4
```

The result should be returned:

```
06:00.0 Processing accelerators: Intel Corporation Device 09c4
```

```

flik@flik-NUC7i5BNH:~/Desktop/getting_start/thunderbolt-software-user-space-master$ lspci | grep 09c4
06:00.0 Processing accelerators: Intel Corporation Device 09c4

```

That shows the driver of the Thunderbolt3 has been installed on your host PC correctly and the PCIe device in the FLIK is detected by the host PC.

Chapter 3

Install Intel Acceleration Stack

This chapter describes how to install the Intel® Acceleration Stack.

1. Go to the link below download Intel® Acceleration Stack Version 1.2, please download “**Acceleration Stack for Development**”. Users do not need to modify the factory code in the serial flash if the FLIK, or do not use the Quartus tool. You can choose “**Acceleration Stack for Runtime**” version to install. Note that in the following steps, you need to pay attention to the difference between the file name and the path name.

https://www.intel.com/content/www/us/en/programmable/products/boards_and_kits/dev-kits/altera/acceleration-card-arria-10-gx/getting-started.html

Acceleration Stack for Development

Accelerator function development using the Intel Quartus Prime Pro Edition Software, Intel FPGA Software Development Kit (SDK) for OpenCL™ and Acceleration Stack

Intel Acceleration Stack Version 1.2

Requires Intel Quartus Prime Pro Edition software version 17.1.1. Software and related interfaces (SR-IOV, Low Latency 10 Gbps and 40 Gbps Ethernet MAC/PHY) are provided in the release

Intel FPGA SDK for OpenCL

[View server models](#)

RHEL 7.4, CentOS 7.4, Ubuntu 16.04

[Intel® Acceleration Stack for Intel® Xeon® CPU with FPGAs Version 1.2 Release Notes](#)

[Intel Acceleration Stack Quick Start Guide for Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA](#)

~18 GB

[Download Now](#)

2. Go to the download file location and open terminal. Unzip the downloaded file:

a10_gx_pac_ias_1_2_pv_dev_installer.tar.gz

```
tar xvf a10_gx_pac_ias_1_2_pv_dev_installer.tar.gz
```

3. Change to the unzipped folder:

```
cd a10_gx_pac_ias_1_2_pv_dev_installer
```

4. Run setup.sh

```
./setup.sh
```

5. You are prompted with the following question: **Do you wish to install OPAE?**
Answer **Y**.

```
Flik@Flik-NUC7i5BNH:~/Desktop/getting_start/a10_gx_pac_ias_1_2_pv_dev_installer$ ./setup.sh
-----
- Beginning installing Intel Acceleration Stack Development Package
-----
Do you wish to install OPAE? Note: Installing will require administrative access (sudo) and network access. (Y/n):
```

6. Accept the license.

```
Flik@Flik-NUC7i5BNH: ~/Desktop/getting_start/a10_gx_pac_ias_1_2_pv_dev_installer
subsidiaries worldwide.

"Authorized Contractors" means a person, company, or other entity
that: (i) provides design, testing, or integration services for You,
provided such integration services shall be performed solely for
implementation within Devices; and (ii) is subject to a written
confidentiality agreement protecting Altera's Confidential Information
with restrictions no less restrictive than those contained in this
Agreement, and You will enforce such agreement in the same fashion as
You would enforce Your own confidentiality agreements of similar
nature and importance, but in no event will You use less than a
reasonable standard. Any access to or use of the Licensed Software or
the Licensed Products by the Authorized Contractor is subject to the
following: (a) such access and/or use shall be for Your sole benefit;
(b) a breach of this Agreement or the terms of any other Altera
agreement by the Authorized Contractor shall be deemed to be a breach
of such agreement(s) by You, and You will be liable for any acts or
omissions of the Authorized Contractor; and (c) You shall ensure that
in no event will any such Authorized Contractor be a competitor of
Altera.

"Authorized Distributor" means a reseller, OEM, ODM, cloud platform
provider, or any distributor or reseller that is authorized by Altera
--More--(3%)
```

```
flik@flik-NUC7I5BNH: ~/Desktop/getting_start/a10_gx_pac_ias_1_2_pv_dev_installer
54. libstdc v3 (GPL v. 3 License)
55. looks 2.0.1 (BSD 2 Clause License)
56. make 3.81 (GPL v. 2 License)
57. miglayout15 3.0.3 (BSD 2 Clause License)
58. miglayout 4.0 (BSD)
59. mpc 1.0.3 (LGPL v. 3 License)
60. mpfr 3.1.0 (LGPL v. 3 License)
61. mpfr 3.1.4 (LGPL v. 3 License)
62. mpir 2.2.1 (LGPL v. 3 License)
63. mydoggy 1.4.2 (LGPL v. 3 License)
64. netbeans-swing-outline 6.9 (LGPL v. 2.1, GPL v. 2.0, and CDDL v. 1 Licenses
plus Classpath Exception)
65. newlib 2.4.0 (Red Hat and BSD 3 Clause Licenses)
66. OpenCL 1.1 (MIT License)
67. powermock 1.5 (Apache v. 2.0 license)
68. quickserver 1.4.7 (LGPL v.2.1 License)
69. stlport 7.1 (Stlport License)
70. swingworker 3 (MPL v. 1.1 and LGPL v. 2.1 Licenses)
71. symphony 5.6.1 (Eclipse Public License v. 1.0)
72. systemc 2.2.0 (SystemC Open Source License v. 3.3)
73. wraplf 0.2 (Apache v. 2.0 License)
74. xerces 2.3.0 (Apache v. 1.1 License)
75. xmlbeans 2.2.0 (Apache v. 2.0 License)
Do you accept this license? (Y/n):
```

7. When you receive an installation directory prompt, you can specify an install directory. Otherwise, the installer uses the default directory at /home/<username>/inteldevstack to install Intel® Quartus® Prime Pro Edition and OpenCL* SDK.

```
flik@flik-NUC7I5BNH: ~/Desktop/getting_start/a10_gx_pac_ias_1_2_pv_dev_installer
57. miglayout15 3.0.3 (BSD 2 Clause License)
58. miglayout 4.0 (BSD)
59. mpc 1.0.3 (LGPL v. 3 License)
60. mpfr 3.1.0 (LGPL v. 3 License)
61. mpfr 3.1.4 (LGPL v. 3 License)
62. mpir 2.2.1 (LGPL v. 3 License)
63. mydoggy 1.4.2 (LGPL v. 3 License)
64. netbeans-swing-outline 6.9 (LGPL v. 2.1, GPL v. 2.0, and CDDL v. 1 Licenses
plus Classpath Exception)
65. newlib 2.4.0 (Red Hat and BSD 3 Clause Licenses)
66. OpenCL 1.1 (MIT License)
67. powermock 1.5 (Apache v. 2.0 license)
68. quickserver 1.4.7 (LGPL v.2.1 License)
69. stlport 7.1 (Stlport License)
70. swingworker 3 (MPL v. 1.1 and LGPL v. 2.1 Licenses)
71. symphony 5.6.1 (Eclipse Public License v. 1.0)
72. systemc 2.2.0 (SystemC Open Source License v. 3.3)
73. wraplf 0.2 (Apache v. 2.0 License)
74. xerces 2.3.0 (Apache v. 1.1 License)
75. xmlbeans 2.2.0 (Apache v. 2.0 License)
Do you accept this license? (Y/n): y

Enter the path you want to extract the Intel PAC with Intel Arria10 GX FPGA rele
ase package [default: /home/flik/inteldevstack]:
```

Chapter 4

Install Intel Distribution of OpenVINO toolkit

This chapter describes how to install the Intel Distribution of OpenVINO toolkit **for Linux* with FPGA support**. The following link is Intel's official installation guide, users can refer to the contents of this article for more information.

https://docs.openvino toolkit.org/latest/docs_install_guides_installing_openvino_linux_fpga.html

1. Go to the link below to download the Intel® Distribution of OpenVINO toolkit for Linux* with FPGA support:
<https://software.intel.com/en-us/openvino-toolkit/choose-download/free-download-linux-fpga>
2. Please select **2019R1** version and select **Full Package** to download.

Serial number : CXWL-X8B25TSL

- Save this serial number. You may need it to activate your product in the installer.
- For your reference, you will receive an email that includes your serial number and download instructions.

Choose a Version

2019 R1 ▼

Build date: 01 Apr 2019

[Release Notes](#) | [Installation Guide](#)

Choose a Download Option

I want to download only the components I need. Time and space are important to me. While I'm connected to the internet, I can install the components I choose. Initial download 15 MB, max download 2344 MB based on component selection.

Customizable Package

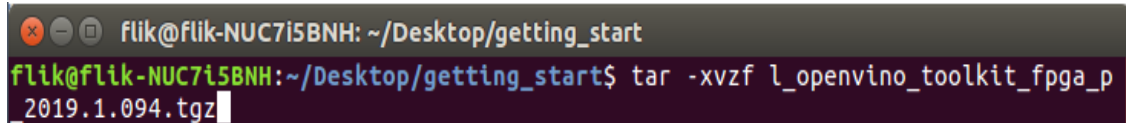
I prefer a single large install package with all components. I can install offline after downloading the entire package. Download size 2344 MB.

Full Package

[Related downloads](#) ▶

3. Open terminal and change directories to where you downloaded the Intel Distribution of OpenVINO toolkit for Linux* with FPGA support package file.
4. Unpack the .tgz file

```
tar -xvzf l_openvino_toolkit_fpga_p_2019.1.094.tgz
```

A terminal window with a dark background. The prompt is 'flik@flik-NUC7i5BNH: ~/Desktop/getting_start'. The command entered is 'tar -xvzf l_openvino_toolkit_fpga_p_2019.1.094.tgz'.

```
flik@flik-NUC7i5BNH: ~/Desktop/getting_start  
flik@flik-NUC7i5BNH:~/Desktop/getting_start$ tar -xvzf l_openvino_toolkit_fpga_p_2019.1.094.tgz
```

The files are unpacked to the l_openvino_toolkit_fpga_p_<version> directory.

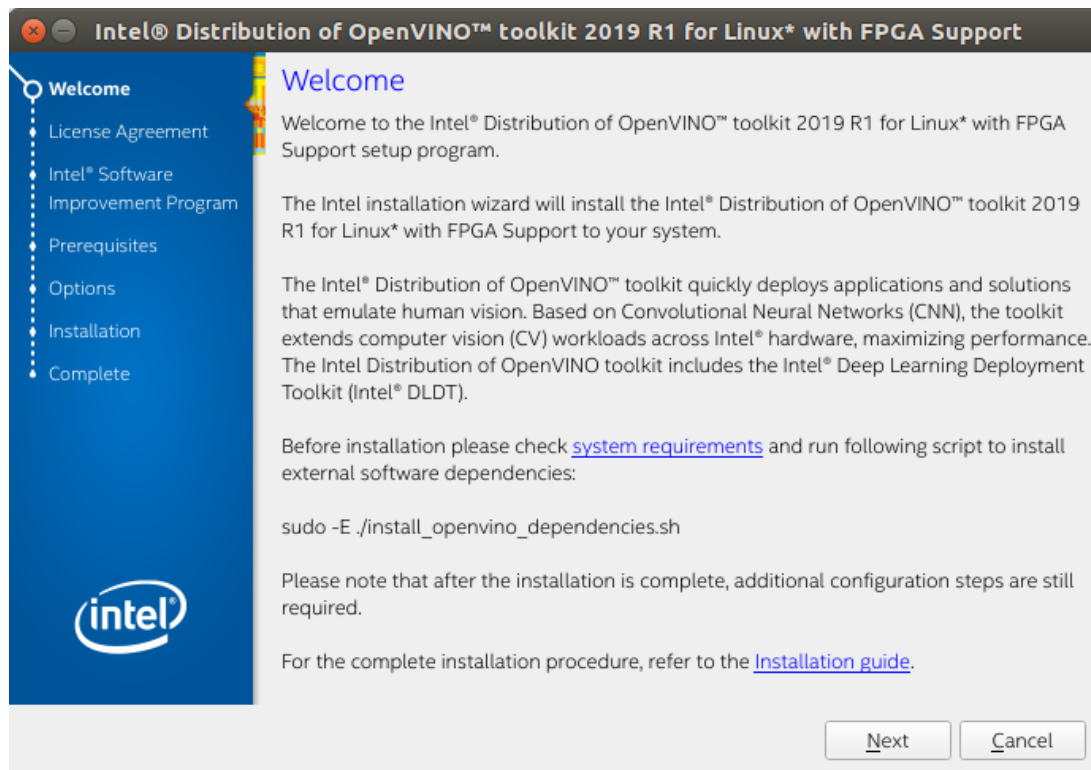
5. Go to the l_openvino_toolkit_fpga_p_<version> directory:

```
cd l_openvino_toolkit_fpga_p_2019.1.094
```

6. Install the toolkit via GUI Installation Wizard.

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

7. Follow the instructions to install the toolkit



8. When installed as root the default installation directory for the Intel Distribution of OpenVINO is /opt/intel/opencvino_fpga_2019.1.094/. For simplicity, a symbolic link to the latest installation is also created: /opt/intel/opencvino/.
9. Install external software dependencies, change to the install_dependencies directory:

```
cd /opt/intel/opencvino/install_dependencies
```

10. Turn the script into an executable file.

```
sudo chmod +x install_openvino_dependencies.sh
```

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

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

Chapter 5

Install and verify the Terasic

Demonstration of the FLIK

This chapter will show you how to quickly experience the Inference Engine demo on the FLIK.





5.1 Setup

This section describes how to quickly setup the environment for running an Inference Engine demonstration.

1. Users first need to download the OpenVINO BSP for Linux " **terasic_demo_flik_linux_2019R1.tar.gz** " in the link below. This file contains the board support package, environment variable settings and demonstration file.

<http://flik.terasic.com/cd>

OpenVINO 2019 R1

Title	Version	Size(KB)	Date Added	Download
OpenVINO BSP for Windows	1.0 		2019-11-27	
OpenVINO BSP for Linux	1.0 		2019-11-27	

2. Open terminal and change directory to where you downloaded the board package file.
3. Extract the downloaded file "terasic_demo_flik_linux_2019R1.tar.gz".

```
tar -xvzf terasic_demo_flik_linux_2019R1.tar.gz
```

4. Copy the "terasic_demo_flik_linux_2019R1" folder to the path:
/opt/intel/opencvino/deployment_tools.

```
sudo cp -r terasic_demo_flik_linux_2019R1 /opt/intel/opencvino/deployment_tools
```

5. Go to the folder location you just copied

```
cd /opt/intel/opencvino/deployment_tools/terasic_demo_flik_linux_2019R1/
```

6. Switch to superuser

```
sudo su
```

```
flik@flik-NUC7i5BNH:/opt/intel/opencvino/deployment_tools/terasic_demo_flik$ sudo  
su  
root@flik-NUC7i5BNH:/opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo  
flik#
```

7. Execute the setting environment variable script.

```
source setup_board_flik.sh
```



```

root@flik-NUC7i5BNH: /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_
root@flik-NUC7i5BNH:/opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo
_flik_linux_2019R1# source setup_board_flik.sh
[setupvars.sh] OpenVINO environment initialized
export AOCL_BOARD_PACKAGE_ROOT=/opt/intel/opencvino_2019.1.094/deployment_tools/t
erasic_demo_flik/opencvcl_bsp
This script handles device permissions and huge page table setup.

Please refer to DCP Quick Start User Guide for OPAE driver and SW installation
instructions.

/etc/security/limits.d/99-opae_memlock.conf is already setup.
setup huge pages. must be done after every reboot
sudo bash -c "echo 20 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages"

setup permissions for device. must be done after every reboot.
sudo chmod 666 /dev/intel-fpga-port.*
sudo chmod 666 /sys/class/fpga/intel-fpga-dev.*/intel-fpga-port.*/userclk_freqcm
d
sudo chmod 666 /sys/class/fpga/intel-fpga-dev.*/intel-fpga-port.*/userclk_freqcn
trcmd
sudo chmod 666 /sys/class/fpga/intel-fpga-dev.*/intel-fpga-port.*/errors/clear
sudo chmod 666 /dev/intel-fpga-fme.*
root@flik-NUC7i5BNH:/opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo
_flik_linux_2019R1#

```

8. Make sure the FLIK is powered on and connected to the host PC via Thunderbolt 3 cable.
9. Execute the following command to verify whether the host PC has detected PCIe device in the FLIK.

```
lspci | grep 09c4
```

The result should be returned:

```
06:00.0 Processing accelerators: Intel Corporation Device 09c4
```

10. Run the simple diagnostic utility:

```
aocl diagnose
```

The **aocl diagnose** command is useful to query the available PCIe device in a system. It will report device information and identifies issues.

There is a diagnose report on the command shell as shown in below. If the command returns "**DIAGNOSTIC_PASSED**", it shows that the hardware configuration is successful.

```
root@flik-NUC7i5BNH: /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_
root@flik-NUC7i5BNH:/opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo
_flik_linux_2019R1# aocl diagnose
-----
Device Name:
acl0

BSP Install Location:
/opt/intel/opencvino/deployment_tools/terasic_demo_flik/opencl_bsp

Vendor: Intel Corp

Physical Dev Name      Status      Information
pac_a10_ec000000      Passed      PAC Arria 10 Platform (pac_a10_ec000000)
                                           PCIe 06:00.0
                                           FPGA temperature = 52 degrees C.

DIAGNOSTIC_PASSED
-----

Call "aocl diagnose <device-names>" to run diagnose for specified devices
Call "aocl diagnose all" to run diagnose for all devices
root@flik-NUC7i5BNH:/opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo
_flik_linux_2019R1#
```

11. The user can also do the test for the individual device connected to the host PC. The command is: `aocl diagnose <device_name>`. If the host PC is connected to only one device, `<device_name>` is usually named "acl0". So after entering the **aocl diagnose acl0** command, the memory transfer test will start and the test result will be returned.

```
aocl diagnose acl0
```

```
root@flik-NUC7i5BNH: /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_
134217728 1170.13 1171.68 1168.58 1170.07
268435456 1176.11 1176.11 1176.11 1176.11

Reading 262144 KBs with block size (in bytes) below:

Block_Size Avg      Max      Min      End-End (MB/s)
524288 1062.86 1283.94 598.86 936.17
1048576 1070.56 1104.71 1032.14 1014.43
2097152 958.09 1046.55 931.38 929.26
4194304 1034.83 1121.87 1008.47 1021.38
8388608 1526.44 1602.76 1428.03 1517.06
16777216 1804.56 1839.18 1696.20 1800.18
33554432 1961.94 1982.98 1913.88 1959.66
67108864 2030.44 2032.39 2027.73 2029.76
134217728 2067.22 2068.04 2066.40 2066.95
268435456 2083.41 2083.41 2083.41 2083.41

Write top speed = 1219.64 MB/s
Read top speed = 2083.41 MB/s
Throughput = 1651.52 MB/s

DIAGNOSTIC_PASSED
```

Next, we will introduce how to use the squeezeNet model with the FLIK to identify objects in the image.

In this demo, a SqueezeNet bitstream will be programmed into the FPGA on the FLIK and deploy the classification sample with a SqueezeNet model.

12. After installing the OpenVINO toolkit, there are several pre-compile bitstreams that can be directly used by Arria 10 FPGA on the FLIK. The location of these files is `/opt/intel/opencvino/bitstreams/a10_dcp_bitstreams` directory. Please execute the following command to program the bitstream file into the FPGA of the FLIK.

In this section, we will use `2019R1_RC_FP11_ResNet_SqueezeNet_VGG.aocx` for programming bit stream.

```
aocl program aocl0 /opt/intel/opencvino/bitstreams/a10_dcp_bitstreams/2019R1_RC_FP11_ResNet_SqueezeNet_VGG.aocx
```

```
root@flik-NUC7i5BNH:/opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_flik_linux_2019R1# cd /opt/intel/opencvino/bitstreams/a10_dcp_bitstreams
root@flik-NUC7i5BNH:/opt/intel/opencvino/bitstreams/a10_dcp_bitstreams# aocl program aocl0 /opt/intel/opencvino/bitstreams/a10_dcp_bitstreams/2019R1_RC_FP11_ResNet_SqueezeNet_VGG.aocx
aocl program: Running program from /opt/intel/opencvino/deployment_tools/terasic_demo_flik/opencv_bsp/linux64/libexec
Program succeed.
```

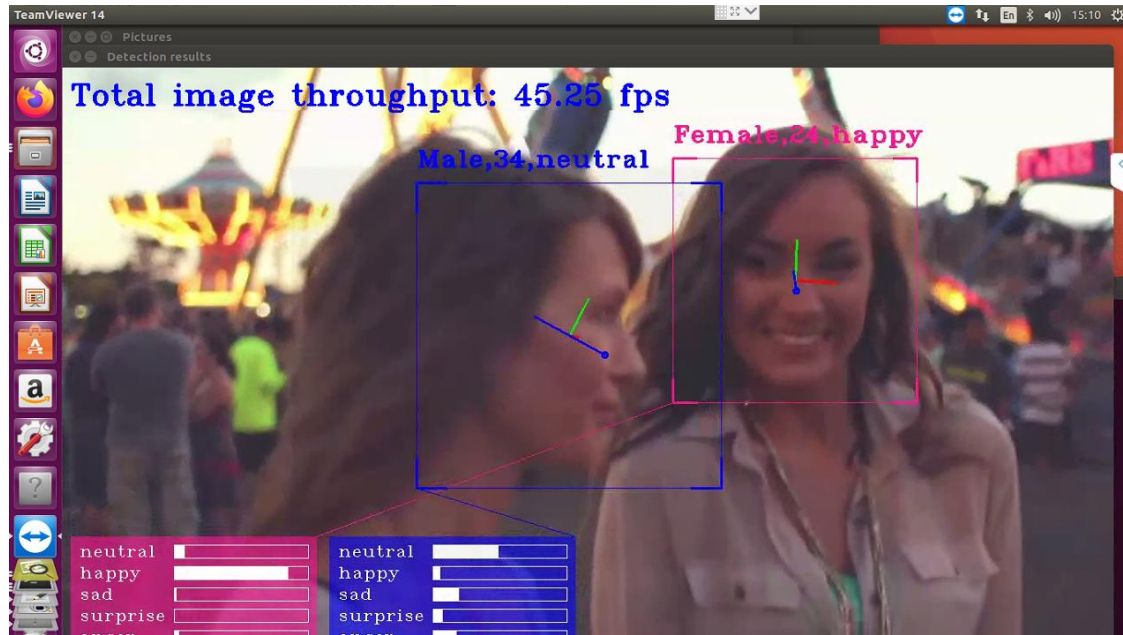
13. Go to the demo file location:

```
cd /opt/intel/opencvino/deployment_tools/terasic_demo_flik_linux_2019R1/demo
```

14. execute the demo batch file

```
./04_face_detection.sh video fpga
```

You will see the face detection as shown in below.

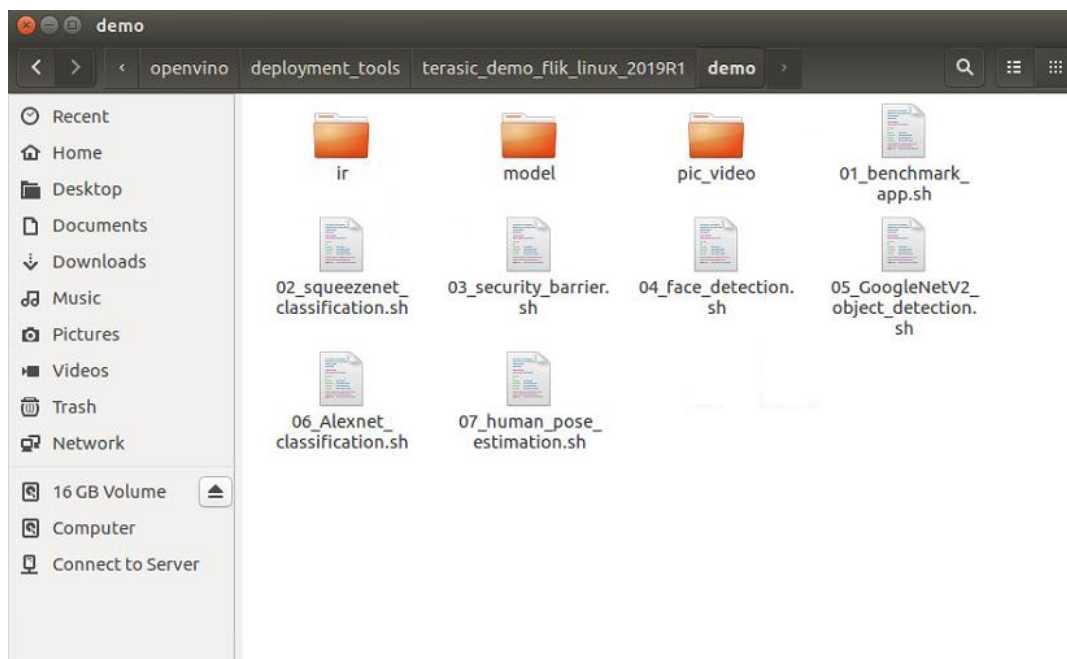


5.2 Other Demonstrations

This section describes how to run the reset of the demonstrations provided for the FLIK.

There are seven Inference Engine demos are provided in the "terasic_demo_flik_linux_2019R1" package. This section will briefly introduce the contents and results of the demo.

As shown in the figure below, there are some shell scripts in the terasic_demo_flik_linux_2019R1/demo folder. Below is the brief introduction of the demo folder.



1. How to use these Shell script files :
Users can run any one of the shell script files with default parameters, and users can also use cpu , vpu or fpga to specify the target device to run the demo. There are also other parameters for using, users can run shell script file with '-h' for more details. And the default parameter is for cpu.
2. The images and video required by the demo are in the pic_video folder.
3. The Caffe model downloaded from internet is in the model folder. The folder includes:
 - Alexnet
 - Squeezenet1.1
 - GoogleNetV2

Users can add Caffe model by referring to the writing rule of the script. Please pay attention to the path and name. Please refer to [Converting a TensorFlow* Model](#) to transfer the Tensorflow model
4. IR folder : While running the demo, the corresponding model IR file will be generated automatically if it's needed.

- The model generated under FP16 folder is used for FPGA
- The model generated under FP32 folder is used for CPU

Please note that each time you restart your terminal, please refer to steps 5~ 7 in section 5.1 to set the environment variable. Also, each time you restart your FLIK, please refer to steps 8~12 to set and verify the FPGA configuration in FLIK.

■ Benchmark Application

This topic demonstrates how to use the Benchmark Application to estimate deep learning inference performance on supported devices. The application outputs latency and throughput. The following are the steps to execute the demo.

1. Go to the demo folder.

```
cd /opt/intel/opencvino/deployment_tools/terasic_demo_flik_linux_2019R1/demo
```

2. Execute the demo batch file (run demo with FPGA).

```
./01_benchmark_app.sh fpga
```

3. The application outputs latency and throughput.

```
root@flik-NUC7i5BNH: /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_
[Step 5/8] Loading model to the plugin
Progress: [.....] 100.00% done

[Step 6/8] Create infer requests and fill input blobs with images
[ INFO ] Infer Request 0 created
[ INFO ] Network Input dimensions (NCHW): 1 3 300 300
[ INFO ] Prepare image /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_d
emo_flik_linux_2019R1/demo/pic_video/test.bmp
[ INFO ] Infer Request 1 created
[ INFO ] Network Input dimensions (NCHW): 1 3 300 300
[ INFO ] Prepare image /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_d
emo_flik_linux_2019R1/demo/pic_video/test.bmp
Progress: [.....] 100.00% done

[Step 7/8] Start inference asynchronously (2000 async inference executions, 2 in
ference requests in parallel)
Progress: [.....] 100.00% done

[Step 8/8] Dump statistics report
[ INFO ] Statistics collecting was not requested. No reports are dumped.
Progress: [.....] 100.00% done

Latency: 6.71 ms
Throughput: 274.66 FPS
root@flik-NUC7i5BNH: /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo
_flik_linux_2019R1/demo#
```

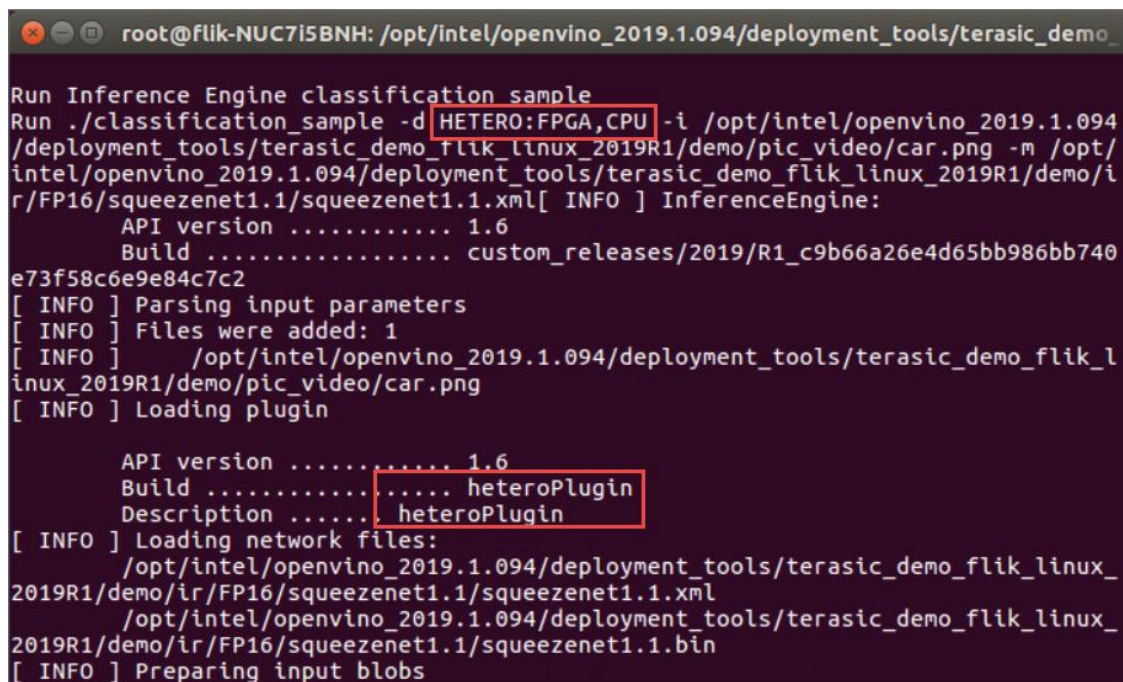

■ Squeezenet Classification

This demo can recognize the objects in the figure by using the squeezenet model

1. Execute the demo batch file (run demo with FPGA).

```
./02_squeezenet_classification.sh fpga
```

2. Users can see “HETERO:FPGA, CPU”, which prompts the DEMO is running on FPGA and CPU.



```
root@flik-NUC7i5BNH: /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_
Run Inference Engine classification sample
Run ./classification_sample -d HETERO:FPGA,CPU -i /opt/intel/opencvino_2019.1.094
/deployment_tools/terasic_demo_flik_linux_2019R1/demo/pic_video/car.png -m /opt/
intel/opencvino_2019.1.094/deployment_tools/terasic_demo_flik_linux_2019R1/demo/i
r/FP16/squeezenet1.1/squeezenet1.1.xml[ INFO ] InferenceEngine:
    API version ..... 1.6
    Build ..... custom_releases/2019/R1_c9b66a26e4d65bb986bb740
e73f58c6e9e84c7c2
[ INFO ] Parsing input parameters
[ INFO ] Files were added: 1
[ INFO ] /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_flik_l
inux_2019R1/demo/pic_video/car.png
[ INFO ] Loading plugin

    API version ..... 1.6
    Build ..... heteroPlugin
    Description ..... heteroPlugin
[ INFO ] Loading network files:
    /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_flik_linux_
2019R1/demo/ir/FP16/squeezenet1.1/squeezenet1.1.xml
    /opt/intel/opencvino_2019.1.094/deployment_tools/terasic_demo_flik_linux_
2019R1/demo/ir/FP16/squeezenet1.1/squeezenet1.1.bin
[ INFO ] Preparing input blobs
```

3. It prints out the top 10 results.

```

root@flik-NUC7I5BNH: /opt/intel/openvino_2019.1.094/deployment_tools/terasic_demo_

Top 10 results:

Image /opt/intel/openvino_2019.1.094/deployment_tools/terasic_demo_flik_linux_20
19R1/demo/pic_video/car.png

classid probability label
-----
817 0.8935490 sports car, sport car
511 0.0444872 convertible
479 0.0444872 car wheel
436 0.0060207 beach wagon, station wagon, wagon, estate car, beach waggon,
station waggon, waggon
751 0.0060207 racer, race car, racing car
656 0.0022149 minivan
864 0.0008148 tow truck, tow car, wrecker
717 0.0008148 pickup, pickup truck
586 0.0008148 half track
581 0.0002998 grille, radiator grille

total inference time: 3.6003520
Average running time of one iteration: 3.6003520 ms

```

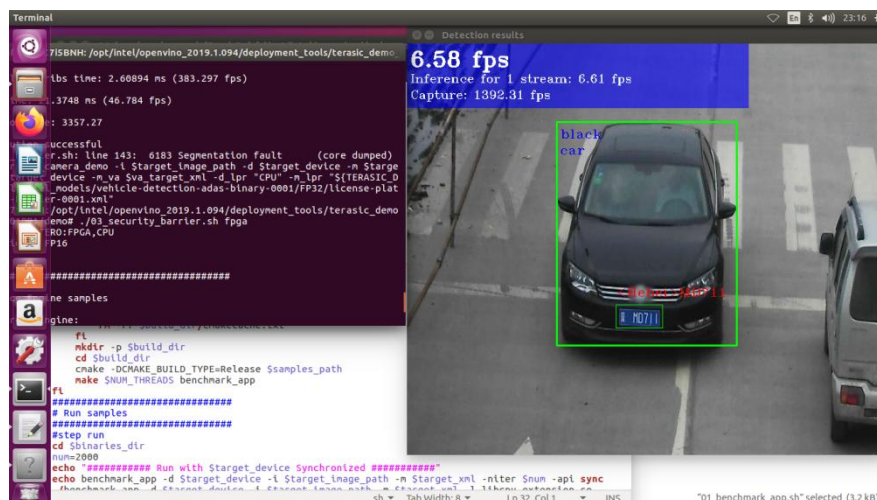
■ Security Barrier

This demo can recognize the car, car license number, and its location by using the three models.

1. Execute the demo batch file

```
./03_security_barrier.sh fpga
```

2. The result is shown in the figure below. Enter Ctrl+C to close the application.



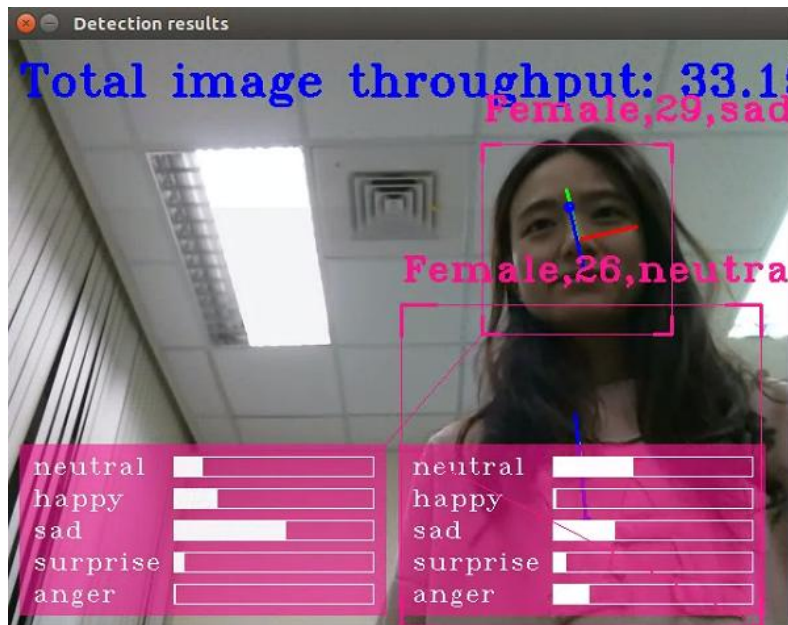
■ Face Detection

This demo uses four models and it can recognize human face position in the figure. It can also judge the human gender, age, expression, and head gesture according to the human face.

1. This video source of this demo can use USB Video Class(UVC) camera or video file in the demo folder. To use an UVC USB camera as the video source, plug a UVC USB camera to the host PC USB port.
2. Execute the demo batch file to run the demo with FPGA and USB camera.

```
./04_face_detection.sh fpga
```

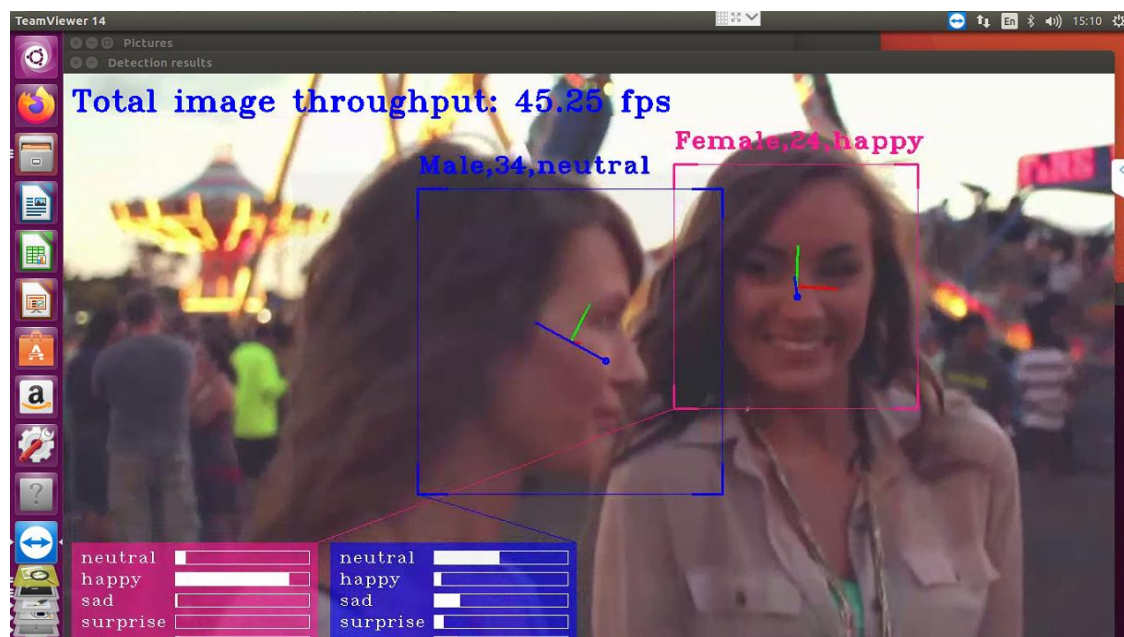
3. The result is shown in the figure below. Enter Ctrl+C to close the application.



4. To use video file as the video source of this demo. Please execute the command in below.

```
./04_face_detection.sh video fpga
```

5. The result is shown in the figure below. Enter Ctrl+C to close the application.



■ GoogleNetV2 Object Detection

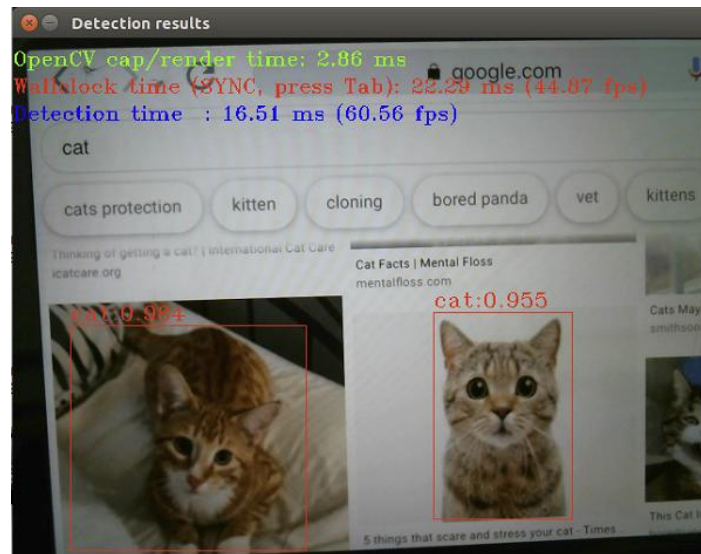
This demo can recognize the target object by using GoogleNetV2. The object tags are shown in the figure below:

1	aeroplane
2	bicycle
3	bird
4	boat
5	bottle
6	bus
7	car
8	cat
9	chair
10	cow
11	diningtable
12	dog
13	horse
14	motorbike
15	person
16	pottedplant
17	sheep
18	sofa
19	train
20	tvmonitor

1. Plug the UVC USB camera to the host PC USB port.
2. Execute the demo batch file to run the demo with FPGA.

`./05_GoogleNetV2_object_detection.sh fpga`

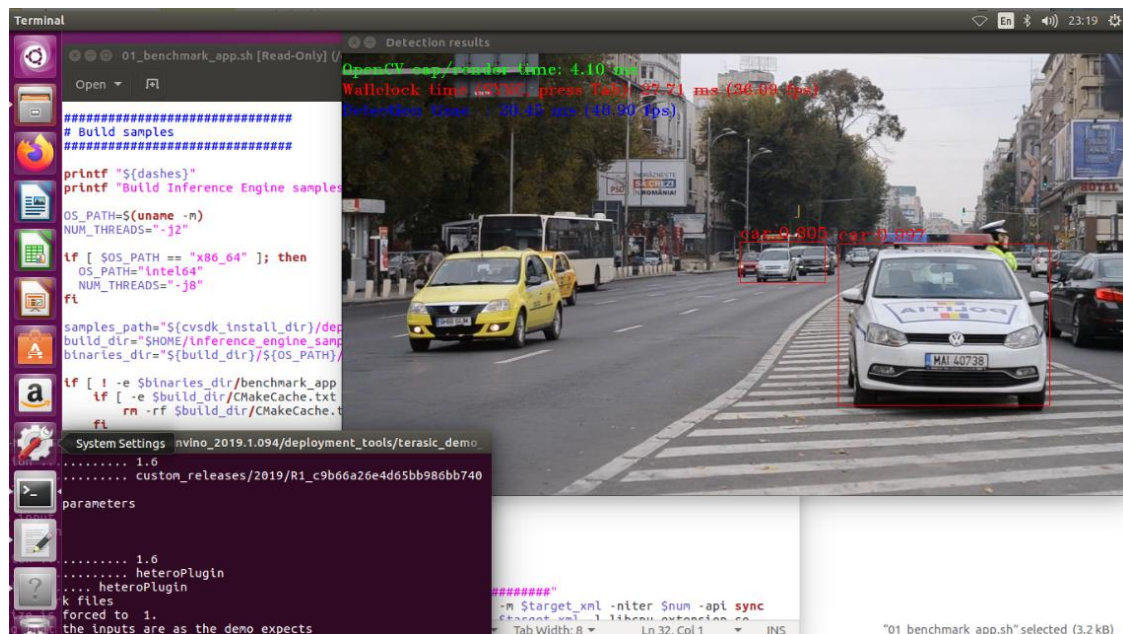
3. The results are shown in the figure below.



4. User can also use the video file as the input source of this demo.

`./05_GoogleNetV2_object_detection.sh video fpga`

5. The results are shown in the figure below.



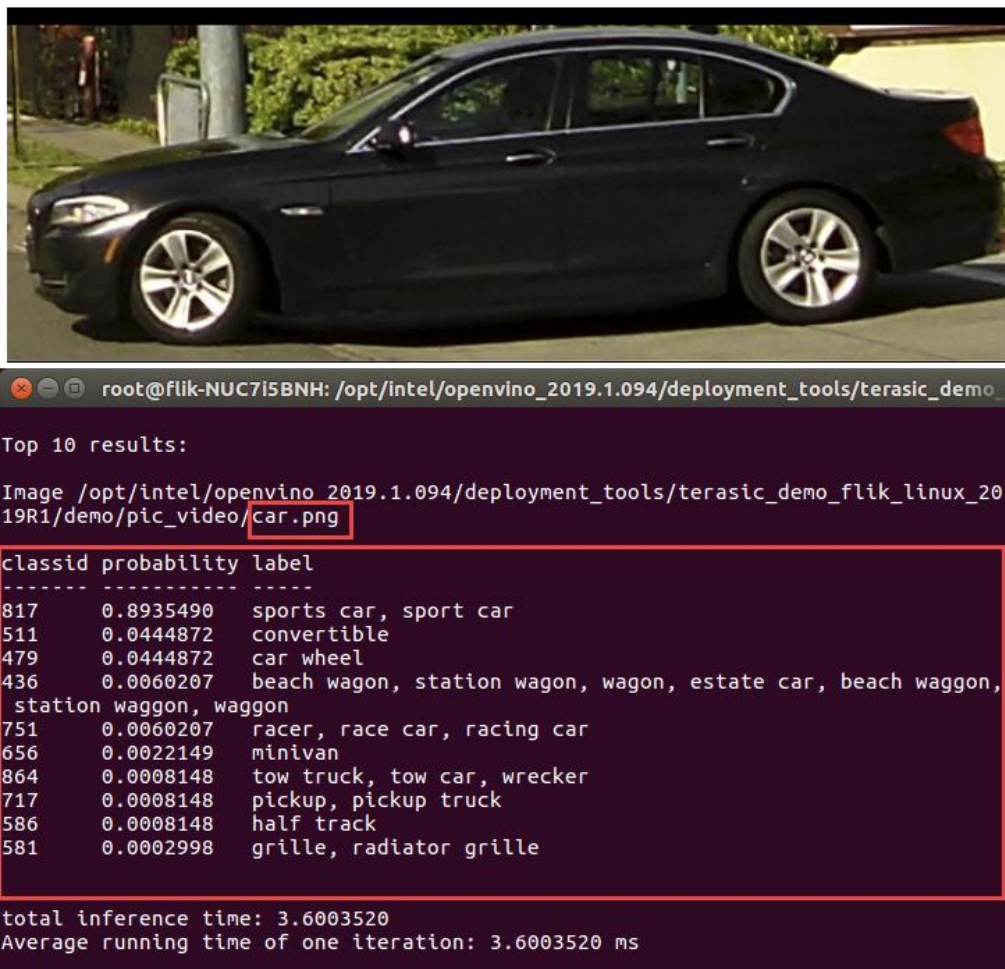
■ Alexnet Classification

This demo can recognize the target objects (a car picture) by using Alexnet model and print out the top 10 information (the recognized result in top 10 probabilities).

1. Execute the demo batch file to run the demo with FPGA.

```
./06_Alexnet_classification.sh fpga
```

2. The results are shown in the figure below.



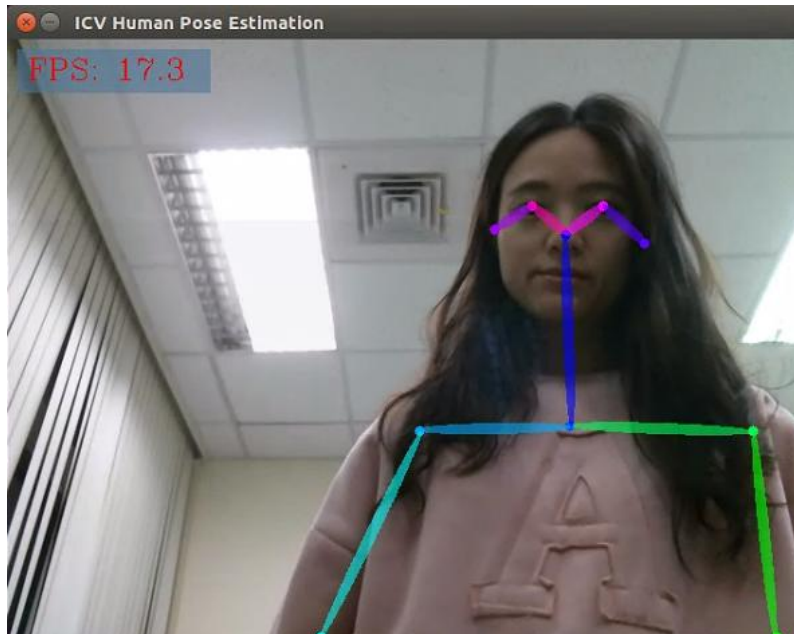
■ Human Pose Estimation

This demo can recognize human pose and display it.

1. Execute the demo batch file to run the demo with FPGA.


```
./07_human_pose_estimation.sh fpga
```

2. The result is shown in the figure below. Enter Ctrl+C to close the application.



Chapter 6

Factory Recovery

This chapter will show you how to recover the factory image file into the serial flash in the FLIK. When the FLIK is power on, the FPGA will read the factory configure file from the serial flash. The factory configuration allows the FPGA perform the acceleration task with the host PC. To program the serial flash file, user needs to connect the host PC to the FLIK via Thunderbolt 3. The configuration file is program to the serial flash from the host PC to the FLIK via the USB Blaster II interface.

Before referring to the steps below, please make sure that the user has already referenced the above chapters and has installed all the software and environment settings required.

- **Hardware Setup**

Power on the FLIK, and connect the host PC to the FLIK using the Thunderbolt 3 cable

- **Install USB Blaster II driver**

1. Please refer the link in below to install the Linux driver for USB Blaster II.
<https://rocketboards.org/foswiki/Documentation/UsingUSBBlasterUnderLinux>
2. After setup USB Blaster II driver, user can enter the command "**lsusb**" to check if the system has detected the USB Blaster correctly. Normally you should see "**Altera**" appear on the device list.

```
lsusb
```

```
flik@flik-NUC7i5BNH:~/Desktop/gs/thunderbolt-software-user-space-master$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 09fb:6810 Altera
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 8087:0a2b Intel Corp.
Bus 001 Device 002: ID 046d:c52b Logitech, Inc. Unifying Receiver
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

3. Go to a path: `/home/<username>/inteldevstack` and execute the following command to run the initialization script to set the required environment variables.

The environment variables setting will include the Quartus programmer. The Quartus programmer is a tool for downloading configuration files to serial flash and FPGA. This tool is available when you install Intel Acceleration Stack for **Development** version as described in Chapter3.

Source `init_env.sh`

```
root@flik-NUC7i5BNH: /home/flik/inteldevstack
flik@flik-NUC7i5BNH:~$ cd /home/flik/inteldevstack/
flik@flik-NUC7i5BNH:~/inteldevstack$ sudo su
[sudo] password for flik:
root@flik-NUC7i5BNH:/home/flik/inteldevstack# source in
init_env.sh intelFPGA_pro/
root@flik-NUC7i5BNH:/home/flik/inteldevstack# source init_env.sh
export QUARTUS_HOME=/home/flik/inteldevstack/intelFPGA_pro/quartus
export INTELFPGA_OCLSDKROOT=/home/flik/inteldevstack/intelFPGA_pro/hld
Adding $QUARTUS_HOME/bin to PATH
export OPAE_PLATFORM_ROOT=/home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv
export AOCL_BOARD_PACKAGE_ROOT=/home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/op
encl/opencl_bsp
source /home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/opencl/opencl_bsp/linux64/
libexec/setup_permissions.sh
Adding $OPAE_PLATFORM_ROOT/bin to PATH
sudo cp /home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/sw/fpgaflash /usr/bin/
sudo cp /home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/sw/afu_platform_info /usr
/bin/
find /home/flik/inteldevstack/a10_gx_pac_ias_1_2_pv/hw/samples/ -type d -name *S
10* -exec rm -r {} +
root@flik-NUC7i5BNH:/home/flik/inteldevstack#
```

4. You can use the Quartus programmer tool to check the JTAG chain status in the FLIK by entering the following command. The result will list the FPGA device status in the FLIK.

`jtagconfig`

```
flik@flik-NUC7i5BNH:~/inteldevstack$ jtagconfig
1) FLIK [3-2]
   02E660DD    10AX115H1(.|E2|ES)/10AX115H2/..
   0318A0DD    10M04S(A|C)
```

5. Go to the bring up file path :

```
cd /opt/intel/opencvino/deployment_tools/terasic_demo_flik_linux_2019R1/bringup/
```

Execute the script file to launch the function for programming the serial flash.

Please note the subcommand "flash".

```
./bringup_board.sh flash
```

```
flik@flik-NUC7i5BNH: /opt/intel/opencvino/deployment_tools/terasic_demo_flik_linux_20
flik@flik-NUC7i5BNH:/opt/intel/opencvino/deployment_tools/terasic_demo_flik_linux
_2019R1/bringup$ ./bringup_board.sh flash
Programing startup image into flash
16M
Info: *****
Info: Running Quartus Prime Programmer
Info: Version 17.1.1 Build 273 12/19/2017 Patches 1.01dcp,1.02dcp,1.36,1.38
SJ Pro Edition
Info: Copyright (C) 2017 Intel Corporation. All rights reserved.
Info: Your use of Intel Corporation's design tools, logic functions
Info: and other software and tools, and its AMPP partner logic
Info: functions, and any output files from any of the foregoing
Info: (including device programming or simulation files), and any
Info: associated documentation or information are expressly subject
Info: to the terms and conditions of the Intel Program License
Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
Info: the Intel FPGA IP License Agreement, or other applicable license
Info: agreement, including, without limitation, that your use is for
Info: the sole purpose of programming logic devices manufactured by
Info: Intel and sold by Intel or its authorized distributors. Please
Info: refer to the applicable agreement for further details.
Info: Processing started: Tue Nov 12 15:59:06 2019
Info: Command: quartus_pgm -m jtag -c 1 -o p;Flik_sfl.sof
```

6. This procedure will first erase flash and then program it. It may take a few minutes.
7. After the program is complete, please power cycling the FLIK (Power off the FLIK first, then reopen the FLIK). Then reboot your system. Thus, the FLIK can restore to factory status for using.

Chapter 7

Additional Information

Here are the addresses where you can get help if you encounter problems:

■ Terasic Technologies

9F., No.176, Sec.2, Gongdao 5th Rd,
East Dist, HsinChu City, Taiwan, 30070

Email: support@terasic.com

Web: www.terasic.com

FLIK Web: flik.terasic.com

■ Revision History

Date	Version	Changes
2019.09	First publication	
2020.01	V1.1	Modify chapter 6