

Enable Tensorflow Machine Learning Framework on LS1046 with Layerscape SDK

Contents

1. Introduction

This application note describes the procedure to enable machine learning frameworks, Tensorflow, on top of the existing Layerscape LSDK releases.

1.1. Scope

The Tensorflow procedure to enable machine learning frameworks has been verified on LS1046. This procedure is applicable on all ARMv8 based Layerscape devices.

1.2. Document Content

This application note details the installation of LSDK on LS1046ARDB board, Tensorflow, and Tensorflow validation using HelloWorld application, image classification and face recognitions. The required software is available from open source repositories. You can see the verified procedure, prerequisite packages, as well as sample outputs in this app note.

This app note has the sections below:

- LSDK installation
- Tensorflow installation
- Tensorflow validation on LSDK

1.	Introduction.....	1
1.1.	Scope.....	1
1.2.	Document Content	1
1.3.	Software and hardware compatibility matrix	2
2.	LSDK installation.....	2
2.1	LSDK download and compilation.....	2
2.2	Flash LSDK composite firmware.....	3
2.3	Deploy Ubuntu userland on LS1046	4
2.4	Create user ID and sudo permission	7
3.	Tensorflow installation.....	8
3.1	Prerequisite package installation	8
3.2	Bazel installation.....	8
3.3	Tensorflow installation.....	8
3.3.1	Source download and patching.....	9
3.3.2	Configuration	10
3.3.3	Compilation and installation	12
4.	Tensorflow validation on LSDK	13
4.1	Hello World!	13
4.2	Image classification using GoogLeNet	13
4.3	Face recognition using Google's FaceNet.....	14
4.3.1	Code checkout.....	14
4.3.2	Download face dataset	14
4.3.3	Download pretrained model	15
4.3.4	LFW dataset alignment	15
4.3.5	FaceNet validation using LFW.....	15
5.	Bibliography.....	15



1.3. Software and hardware compatibility matrix

Table 1 Software and hardware compatibility matrix

HW/SW	Version
Board	LS1046ARDB
Layerscape SDK (LSDK)	18.06
Bazel	0.15.2
Tensorflow	1.9.0

2. LSDK installation

Refer to (NXP Semiconductors, 2018) for the detailed LSDK-1806 installation procedure on LS1046ARDB boards.

2.1 LSDK download and compilation

See the below link to download LSDK-18.06:

<https://nxp.flexnetoperations.com/control/frse/download?agree=Accept&element=10174317>

Product Download

Layerscape Software Development Kit

Files

License Keys

Notes

[? Download Help](#)

To get started with flexbuild, refer to the Quick-start Guide at [LSDK knowledge-center link](#)

<input type="checkbox"/>	+ File Description	File Size	File Name
<input checked="" type="checkbox"/>	+ flexbuild_lsdk1806	63.7 KB	flexbuild_lsdk1806.tgz

Download Selected Files

Download the following tar ball to an X86 host machine running Ubuntu 16.02 or newer version.

- flexbuild_lsdk1806.tgz

Execute the following commands below to compile the LSDK userland using Ubuntu 18.04 docker container.

```
$tar zxvf flexbuild_lsdk1806.tgz
$cd flexbuild/
$source setup.env
```

Enable Tensorflow Machine Learning Framework on LS1046 with Layerscape SDK, Application Notes, Rev. 1.1, 07/2018

```

$ flex-builder docker
root@fbubuntu:~# source setup.env
root@fbubuntu:~# wget
http://www.nxp.com/lgfiles/sdk/lsdk1806/app\_components\_arm64.tgz
root@fbubuntu:~# wget
http://www.nxp.com/lgfiles/sdk/lsdk1806/bootpartition\_arm64\_lts\_4.14.tgz
root@fbubuntu:~# wget http://www.nxp.com/lgfiles/sdk/lsdk1806/arm64-modules-4.14.47.tgz
root@fbubuntu:~# flex-builder -i mkrfs -a arm64
root@fbubuntu:~# tar xvzf app_components_arm64.tgz -C build/apps
root@fbubuntu:~# tar xvzf arm64-modules-4.14.47.tgz -C
build/rfs/rootfs_ubuntu_bionic_arm64/lib/modules
root@fbubuntu:~# flex-builder -i merge-component -a arm64
root@fbubuntu:~# flex-builder -i compressrfs -a arm64
root@fbubuntu:~# exit

$ ls -l build/images/
total 976204
-rw-r--r-- 1 root root      26994 Jul 28 13:43 linux-lib-firmware.tgz
-rw-r--r-- 1 root root 999599805 Jul 28 13:48
rootfs_ubuntu_bionic_arm64_201807281844.tgz
lrwxrwxrwx 1 root root         43 Jul 28 13:49 rootfs_ubuntu_bionic_arm64.tgz ->
rootfs_ubuntu_bionic_arm64_201807281844.tgz
b42705@mahshev:~/LSDK_1806/flexbuild$

```

2.2 Flash LSDK composite firmware

To run LSDK-1803 flash the board with the compatible firmware.

Retrieve the composite firmware image from the following URL and move it to the TFTP directory on the host:

```

$cd <tftp-dir>
$wget
http://www.nxp.com/lgfiles/sdk/lsdk1806/firmware\_ls1046ardb\_uboot\_qspiboot.img

```

Power cycle the LS1046ARDB board and boot to U-Boot prompt in the primary bank. From the U-Boot prompt, one can flash the composite firmware image using the commands below and reboot to the alternative bank:

```
=> tftp a0000000 firmware_ls1046ardb_uboot_qspiboot.img
=> sf probe 0:1
=> sf erase 0 +$filesize && sf write 0xa0000000 0 $filesize
=> cpld reset altbank
```

The board shall be able to boot to the default TinyDistro Linux prompt, as shown below section.

2.3 Deploy Ubuntu userland on LS1046

The detailed instructions are available from Section 4.1.3.2 in (NXP Semiconductors, 2018).

From TinyDistro Linux:

```
root@TinyDistro:~# udhcpc -i eth0
root@TinyDistro:~# flex-installer -i pf -d /dev/mmcblk0
root@TinyDistro:~# cd /run/media/mmcblk0p3
root@TinyDistro:/run/media/mmcblk0p3# scp
<user>@<builder_host>:flexbuild/bootpartition_arm64_lts_4.14.tgz .
root@TinyDistro:/run/media/mmcblk0p3# scp
<user>@<builder_host>:flexbuild/build/images/rootfs_ubuntu_bionic_arm64_20180728184
4.tgz .
root@TinyDistro:/run/media/mmcblk0p3# flex-installer -i install -b
bootpartition_arm64_lts_4.14.tgz -r rootfs_ubuntu_bionic_arm64_201807281844.tgz -d
/dev/mmcblk0
```

On completing the newly built Ubuntu userland, reboot the board and the system shall boot to the full-blown Linux prompt, as shown below.

```
U-Boot 2018.03-gb3f98d438e (Jun 23 2018 - 14:48:59 +0800)

SoC: LS1046AE Rev1.0 (0x87070010)
Clock Configuration:
  CPU0 (A72):1800 MHz  CPU1 (A72):1800 MHz  CPU2 (A72):1800 MHz
  CPU3 (A72):1800 MHz
  Bus:      700 MHz  DDR:      2100 MT/s  FMAN:      800 MHz
Reset Configuration Word (RCW):
  00000000: 0e150012 10000000 00000000 00000000
  00000010: 11335559 40005012 40025000 c1000000
  00000020: 00000000 00000000 00000000 00238800
  00000030: 20124000 00003101 00000096 00000001
Model: LS1046A RDB Board
Board: LS1046ARDB, boot from QSPI vBank 4
```

```
CPLD: V2.3
PCBA: V2.0
SERDES Reference Clocks:
SD1_CLK1 = 156.25MHZ, SD1_CLK2 = 100.00MHZ
I2C: ready
DRAM: Initializing DDR...using SPD
Detected UDIMM 18ASF1G72AZ-2G3B1
7.9 GiB (DDR4, 64-bit, CL=15, ECC on)
    DDR Chip-Select Interleaving Mode: CS0+CS1
SEC0: RNG instantiated
PPA Firmware: Version LSDK-18.06
SEC Firmware: 'loadables' present in config
loadables: 'trustedOS@1'
Using SERDES1 Protocol: 4403 (0x1133)
Using SERDES2 Protocol: 21849 (0x5559)
NAND: 512 MiB
MMC: FSL_SDHC: 0
Loading Environment from SPI Flash... SF: Detected s25fs512s with page size 256
Bytes, erase size 256 KiB, total 64 MiB
*** Warning - bad CRC, using default environment

Failed (-5)
EEPROM: NXID v1
In: serial
Out: serial
Err: serial
SATA link 0 timeout.
AHCI 0001.0301 32 slots 1 ports 6 Gbps 0x1 impl SATA mode
flags: 64bit ncq pm clo only pmp fbss pio slum part ccc apst
Found 0 device(s).
SCSI: Net: SF: Detected s25fs512s with page size 256 Bytes, erase size 256 KiB,
total 64 MiB
Fman1: Uploading microcode version 106.4.18
PCIe0: pcie@3400000 Root Complex: no link
PCIe1: pcie@3500000 Root Complex: no link
PCIe2: pcie@3600000 Root Complex: no link
FM1@DTSEC3 [PRIME], FM1@DTSEC4, FM1@DTSEC5, FM1@DTSEC6, FM1@TGEC1, FM1@TGEC2
Hit any key to stop autoboot: 0
scanning bus for devices...
Found 0 device(s).
```

Enable Tensorflow Machine Learning Framework on LS1046 with Layerscape SDK, Application Notes, Rev. 1.1, 07/2018

LSDK installation

```
Device 0: not available
switch to partitions #0, OK
mmc0 is current device
Scanning mmc 0:1...
Scanning mmc 0:2...
Found U-Boot script /ls1046ardb_boot.scr
716 bytes read in 242 ms (2 KiB/s)
## Executing script at 80000000
21119488 bytes read in 1225 ms (16.4 MiB/s)
28231 bytes read in 144 ms (191.4 KiB/s)
## Flattened Device Tree blob at 90000000
   Booting using the fdt blob at 0x90000000
   Using Device Tree in place at 0000000090000000, end 0000000090019e46

Starting kernel ...

[ 0.000000] Booting Linux on physical CPU 0x0
[ 0.000000] Linux version 4.14.47 (jenkins@saturn) (gcc version 7.3.0
(Ubuntu/Linaro 7.3.0-16ubuntu3)) #1 SMP PREEMPT Sat Jun 23 14:46:49 CST 2018
[ 0.000000] Boot CPU: AArch64 Processor [410fd082]
[ 0.000000] Machine model: LS1046A RDB Board
[ 0.000000] earlycon: uart8250 at MMIO 0x00000000021c0500 (options '')
[ 0.000000] bootconsole [uart8250] enabled
[ 0.000000] efi: Getting EFI parameters from FDT:
[ 0.000000] efi: UEFI not found.

... ..
[ OK ] Started Getty on tty1.
[ OK ] Reached target Login Prompts.
[ OK ] Started LSB: HPA's tftp server.
[ OK ] Started OpenBSD Secure Shell server.
[ OK ] Started LSB: Mount debugfs on /sys/kernel/debug.
[ OK ] Started Ubuntu FAN network setup.
[ OK ] Started Dispatcher daemon for systemd-networkd.
[ OK ] Started LSB: Load kernel modules needed to enable cpufreq scaling.
      Starting LSB: set CPUFreq kernel parameters...
[ OK ] Started LSB: set CPUFreq kernel parameters.
      Starting LSB: Set sysfs variables from /etc/sysfs.conf...
[ OK ] Started LSB: Set sysfs variables from /etc/sysfs.conf.
[ OK ] Started Virtualization daemon.
```

Enable Tensorflow Machine Learning Framework on LS1046 with Layerscape SDK, Application Notes, Rev. 1.1, 07/2018

```
[ OK ] Started LXC network bridge setup.
      Starting LXC Container Initialization and Autoboot Code...
      Starting Suspend/Resume Running libvirt Guests...
[ OK ] Started Suspend/Resume Running libvirt Guests.
[ OK ] Started LXC Container Initialization and Autoboot Code.
```

```
Ubuntu 18.04.1 LTS localhost ttyS0
```

```
localhost login:
```

```
Password:
```

```
Login incorrect
```

```
localhost login: root
```

```
Password:
```

```
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.14.47 aarch64)
```

```
root@localhost:~# dhclient
```

2.4 Create user ID and sudo permission

```
# dhclient
# adduser <userID>
# usermod -aG sudo <userID>
# date -set <date-str>
```

3. Tensorflow installation

3.1 Prerequisite package installation

Execute the commands below, to install the required packages:

```
$sudo apt-get install python-numpy python-dev python-pip python-wheel
$sudo apt-get install build-essential openjdk-8-jdk python zip
$sudo apt-get install python-opencv libblas-dev liblapack-dev
$sudo apt-get install gfortran libpng-dev libjpeg-dev
$sudo pip install Pillow scipy sklearn twisted autobahn flask flask_restful
service_identity imutils mock
```

3.2 Bazel installation

Bazel is the compilation tool for Tensorflow. The Bazel release packages can be downloaded from <https://github.com/bazelbuild/bazel/releases>. Bazel-0.15.2 is used in this application note.

```
$wget https://github.com/bazelbuild/bazel/releases/download/0.15.2/bazel-0.15.2-
dist.zip
$mkdir bazel-0.15.2 && cd bazel-0.15.2
$unzip ../bazel-0.15.2-dist.zip
$./compile.sh
$sudo cp output/bazel /usr/local/bin/
b42705@localhost:~$ bazel version
WARNING: --batch mode is deprecated. Please instead explicitly shut down your Bazel
server using the command "bazel shutdown".
Build label: 0.15.2- (@non-git)
Build target: bazel-out/arm-
opt/bin/src/main/java/com/google/devtools/build/lib/bazel/BazelServer_deploy.jar
Build time: Thu Jul 26 19:08:55 2018 (1532632135)
Build timestamp: 1532632135
Build timestamp as int: 1532632135
b42705@localhost:~$
```

3.3 Tensorflow installation

Tensorflow is Google's machine learning framework. The most authoritative reference on Tensorflow is available at <https://www.tensorflow.org/>. The following is the verified procedure to compile Tensorflow from source for LS1046A and other ARMv8 based Layerscape devices

3.3.1 Source download and patching

Download source as shown below:

```
$git clone https://github.com/tensorflow/tensorflow
$cd tensorflow
$git checkout v1.9.0
$git apply --verbose --ignore-whitespace --reject tf.diff
```

NOTE: The Tensorflow source shall be patched with the following diff file to avoid compilation issue on aarch64.

```
$ cat tf.diff
diff --git a/tensorflow/contrib/lite/build_def.bzl
b/tensorflow/contrib/lite/build_def.bzl
index b9e40cc50c..eb013dfa62 100644
--- a/tensorflow/contrib/lite/build_def.bzl
+++ b/tensorflow/contrib/lite/build_def.bzl
@@ -14,8 +14,8 @@ def tflite_copts():
     "-O3",
     ],
     str(Label("//tensorflow:android_arm")): [
-        "-mfpu=neon",
-        "-mfloat-abi=softfp",
+#         "-mfpu=neon",
+#         "-mfloat-abi=softfp",
         "-std=c++11",
         "-O3",
     ],
diff --git a/tensorflow/contrib/lite/kernels/internal/BUILD
b/tensorflow/contrib/lite/kernels/internal/BUILD
index 0a5223b235..2eff76dc24 100644
--- a/tensorflow/contrib/lite/kernels/internal/BUILD
+++ b/tensorflow/contrib/lite/kernels/internal/BUILD
@@ -21,7 +21,7 @@ HARD_FP_FLAGS_IF_APPLICABLE = select({
  NEON_FLAGS_IF_APPLICABLE = select({
    ":arm": [
      "-O3",
-     "-mfpu=neon",
+#     "-mfpu=neon",
    ],
    ":armeabi-v7a": [
```

Enable Tensorflow Machine Learning Framework on LS1046 with Layerscape SDK, Application Notes, Rev. 1.1, 07/2018

```

    "-O3",
diff --git a/third_party/png.BUILD b/third_party/png.BUILD
index 17c5449cc0..7b095cbc36 100644
--- a/third_party/png.BUILD
+++ b/third_party/png.BUILD
@@ -42,6 +42,7 @@ cc_library(
    ],
    includes = ["."],
    linkopts = ["-lm"],
+   copts = ["-DPNG_ARM_NEON_OPT=0"],
    visibility = ["//visibility:public"],
    deps = ["@zlib_archive//:zlib"],
)

```

3.3.2 Configuration

Configure Bazel as shown below:

```

b42705@localhost:~/sw_perf/tensorflow$ ./configure
WARNING: --batch mode is deprecated. Please instead explicitly shut down your Bazel
server using the command "bazel shutdown".
You have bazel 0.15.2- (@non-git) installed.
Please specify the location of python. [Default is /usr/bin/python]:

Found possible Python library paths:
  /usr/local/lib/python2.7/dist-packages
  /usr/lib/python2.7/dist-packages
Please input the desired Python library path to use.  Default is
[/usr/local/lib/python2.7/dist-packages]

Do you wish to build TensorFlow with jemalloc as malloc support? [Y/n]:
jemalloc as malloc support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Google Cloud Platform support? [Y/n]:
Google Cloud Platform support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Hadoop File System support? [Y/n]: n
No Hadoop File System support will be enabled for TensorFlow.

```

```
Do you wish to build TensorFlow with Amazon S3 File System support? [Y/n]: n
No Amazon S3 File System support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Apache Kafka Platform support? [Y/n]: n
No Apache Kafka Platform support will be enabled for TensorFlow.

Do you wish to build TensorFlow with XLA JIT support? [y/N]:
No XLA JIT support will be enabled for TensorFlow.

Do you wish to build TensorFlow with GDR support? [y/N]:
No GDR support will be enabled for TensorFlow.

Do you wish to build TensorFlow with VERBS support? [y/N]:
No VERBS support will be enabled for TensorFlow.

Do you wish to build TensorFlow with OpenCL SYCL support? [y/N]:
No OpenCL SYCL support will be enabled for TensorFlow.

Do you wish to build TensorFlow with CUDA support? [y/N]:
No CUDA support will be enabled for TensorFlow.

Do you wish to download a fresh release of clang? (Experimental) [y/N]:
Clang will not be downloaded.

Do you wish to build TensorFlow with MPI support? [y/N]:
No MPI support will be enabled for TensorFlow.

Please specify optimization flags to use during compilation when bazel option "--
config=opt" is specified [Default is -march=native]:

Would you like to interactively configure ./WORKSPACE for Android builds? [y/N]:
Not configuring the WORKSPACE for Android builds.

Preconfigured Bazel build configs. You can use any of the below by adding "--
config=<>" to your build command. See tools/bazel.rc for more details.
    --config=mkl          # Build with MKL support.
    --config=monolithic  # Config for mostly static monolithic build.
Configuration finished
```

**Enable Tensorflow Machine Learning Framework on LS1046 with Layerscape SDK, Application Notes, Rev. 1.1,
07/2018**

3.3.3 Compilation and installation

```

$bazel build --config=opt //tensorflow/tools/pip_package:build_pip_package
...
Target //tensorflow/tools/pip_package:build_pip_package up-to-date:
  bazel-bin/tensorflow/tools/pip_package/build_pip_package
INFO: Elapsed time: 10189.229s, Critical Path: 189.74s
INFO: 5463 processes: 5463 local.
INFO: Build completed successfully, 6098 total actions

b42705@localhost:~/sw_perf/tensorflow$ bazel-
bin/tensorflow/tools/pip_package/build_pip_package /tmp/tensorflow_pkg
Sun Jul 29 00:03:56 UTC 2018 : === Preparing sources in dir: /tmp/tmp.isJCGxBofn
~/sw_perf/tensorflow ~/sw_perf/tensorflow
~/sw_perf/tensorflow
Sun Jul 29 00:04:12 UTC 2018 : === Building wheel
warning: no files found matching '*.dll' under directory '*'
warning: no files found matching '*.lib' under directory '*'
warning: no files found matching '*.h' under directory
'tensorflow/include/tensorflow'
warning: no files found matching '*' under directory 'tensorflow/include/Eigen'
warning: no files found matching '*.h' under directory 'tensorflow/include/google'
warning: no files found matching '*' under directory
'tensorflow/include/third_party'
warning: no files found matching '*' under directory
'tensorflow/include/unsupported'
Sun Jul 29 00:04:35 UTC 2018 : === Output wheel file is in: /tmp/tensorflow_pkg

b42705@localhost:~$ sudo pip install /tmp/tensorflow_pkg/tensorflow-1.9.0-cp27-
cp27mu-linux_aarch64.whl
... ..
Installing collected packages: protobuf, astor, backports weakref, futures,
markdown, tensorboard, termcolor, gast, grpcio, absl-py, tensorflow
  Running setup.py install for termcolor ... done
  Running setup.py install for gast ... done
  Running setup.py install for grpcio ... done
  Running setup.py install for absl-py ... done
Successfully installed absl-py-0.3.0 astor-0.7.1 backports.weakref-1.0.post1
futures-3.2.0 gast-0.2.0 grpcio-1.13.0 markdown-2.6.11 protobuf-3.6.0 tensorboard-
1.9.0 tensorflow-1.9.0 termcolor-1.1.0
b42705@localhost:~$

```

4. Tensorflow validation on LSDK

4.1 Hello World!

```
b42705@localhost:~$ python
Python 2.7.15rc1 (default, Apr 15 2018, 21:51:34)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> tf.__version__
'1.9.0'
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
Hello, TensorFlow!
>>> quit()
```

4.2 Image classification using GoogLenet

This sample uses the pretrained GoogLeNet from Google's Tensorflow model repository to test image classification and validate Tensorflow compilation and installation.

```
$git clone https://github.com/tensorflow/models.git tensorflow-models
b42705@localhost:~/sw_perf/tensorflow-models$ git log
commit 22e248cebd6a556594f5631f5213bf13fd6f5792 (HEAD -> master, origin/master,
origin/HEAD)
Merge: 696b69a4 a2dca9f5
Author: Lukasz Kaiser <lukaszkaizer@users.noreply.github.com>
Date: Thu Jul 26 16:13:32 2018 -0700

Merge pull request #4885 from norouzi/master

Adding keypointnet to research/models.
$cd tutorials/image/imagenet/
b42705@localhost:~/sw_perf/tensorflow-models/tutorials/image/imagenet$ python
classify_image.py
>> Downloading inception-2015-12-05.tgz 100.0%
Successfully downloaded inception-2015-12-05.tgz 88931400 bytes.
2018-07-29 00:20:03.484427: W tensorflow/core/framework/op_def_util.cc:346] Op
BatchNormWithGlobalNormalization is deprecated. It will cease to work in GraphDef
version 9. Use tf.nn.batch_normalization().
```

Enable Tensorflow Machine Learning Framework on LS1046 with Layerscape SDK, Application Notes, Rev. 1.1, 07/2018

```
giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (score = 0.89107)
indri, indris, Indri indri, Indri brevicaudatus (score = 0.00779)
lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (score =
0.00296)
custard apple (score = 0.00147)
earthstar (score = 0.00117)
b42705@localhost:~/sw_perf/tensorflow-models/tutorials/image/imagenet$
```

4.3 Face recognition using Google's FaceNet

This sample demonstrates the Tensorflow based implementation of Google's face recognition algorithm, FaceNet, on LSDK.

An opensource implementation of the FaceNet algorithm is available at

<https://github.com/davidsandberg/facenet>

4.3.1 Code checkout

```
$Git clone https://github.com/davidsandberg/facenet.git
b42705@localhost:~/sw_perf/apps$ cd facenet/
b42705@localhost:~/sw_perf/apps/facenet$ git log
commit 096ed770f163957c1e56efa7feeb194773920f6e (HEAD -> master, origin/master,
origin/HEAD)
Merge: e9d4e8e fa08fa8
Author: David Sandberg <david.o.sandberg@gmail.com>
Date: Mon Apr 16 19:32:11 2018 +0200
```

4.3.2 Download face dataset

The Labelled Face in the Wild (LFW) dataset is used, available from

<http://vis-www.cs.umass.edu/lfw/lfw.tgz>

```
$mkdir -p datasets/lfw/raw && cd datasets/
$wget http://vis-www.cs.umass.edu/lfw/lfw.tgz
$tar zxvf lfw.tgz -C lfw/raw --strip-components=1
```

4.3.3 Download pretrained model

The pretrained FaceNet model is available from

<https://drive.google.com/file/d/1EXPBSXwTaqrSC0OhUdXNmKSh9qJUQ55-/view>

```
$mkdir -p models/facenet
$unzip 20180402-114759.zip
```

4.3.4 LFW dataset alignment

The detailed instructions of validating FaceNet against the LFW dataset is available from (Sanberg, 2018). Here are the actual commands run on LS1046ARDB:

```
$export PYTHONPATH=$PWD/src
b42705@localhost:~/sw_perf/apps/facenet$ for N in {1..4}; do python
src/align/align_dataset_mtcnn.py datasets/lfw/raw datasets/lfw/lfw_mtcnpy_160 --
image_size 160 --margin 32 --random_order --gpu_memory_fraction 0.25 & done
```

4.3.5 FaceNet validation using LFW

```
b42705@localhost:~/sw_perf/apps/facenet$ python src/validate_on_lfw.py
datasets/lfw/lfw_mtcnpy_160 models/facenet/20180402-114759 --distance_metric 1 --
use_flipped_images --subtract_mean --use_fixed_image_standardization
Model directory: models/facenet/20180402-114759
Metagraph file: model-20180402-114759.meta
Checkpoint file: model-20180402-114759.ckpt-275

... ..
Accuracy: 0.99650+-0.00302
Validation rate: 0.98567+-0.00967 @ FAR=0.00100
Area Under Curve (AUC): 1.000
Equal Error Rate (EER): 0.004
```

5. Bibliography

<https://github.com/davidsandberg/facenet,2018>
NXP Semiconductors 2018 *Layerscape Software Development Kit 18.06 Documentation*

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address:
nxp.com/SalesTermsandConditions..

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, Freescale, the Freescale logo, CodeWarrior, Layerscape, PowerQUICC, QorIQ, CoreNet, and QUICC Engine are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, Cortex, and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2018 NXP B.V.

Document Number: AN12209
Rev. 1.1
07/2018