Application Note

TI Deep Learning Library Upgrade Solution

Kangjia Dong and Anshu Jian

ABSTRACT

This application note demonstrates how to upgrade the TIDL in the old version of the SDK to the latest version. These upgrades include fixing some bugs in the old version of TIDL and supporting some new features. It is recommended for customers to be on the latest TIDL so as to get all the bug fixes done along with new features added in new SDK releases.

Table of Contents

List of Figures

Trademarks

1 Introduction

Deep learning algorithms based on deep neural networks (DNNs) are used in many industries, such as robotics, industrial, and automotive. More and more DNN-based machine learning algorithms are used in ADAS products, such as lane line detection, traffic light recognition, pedestrian recognition and other basic ADAS functions are implemented using DNN algorithms. TI's latest generation automotive processor TDA4VM integrates high-performance computing unit C7x Digital Signal Processor (DSP) and Deep-learning Matrix Multiply Accelerator (MMA), which can efficiently perform some basic deep learning operators such as convolution calculations and matrix transformations. These DNN neural network algorithms usually require a lot of computation, and the C7x and MMA in TI TDA4 series processors can accelerate the computation of some DNN operators to achieve fast inference and recognition results. TI Deep Learning Library (TIDL) is a software ecosystem based on deep learning algorithms for TI platforms, which can quickly deploy some common deep learning algorithm models to TI embedded platforms.

TDA4xx is a multi-core heterogeneous SOC launched by TI. It has built-in TI's latest generation C7X DSP and Matrix Multiply Accelerator (MMA), which can efficiently process digital signals, especially good at processing audio, millimeter wave radar, lidar, and camera data. For the current hot AI technology, TI has also integrated related AI operators to facilitate customers to quickly deploy deep learning models. The software part of C7 mainly depends on the following parts in the whole SOC development process

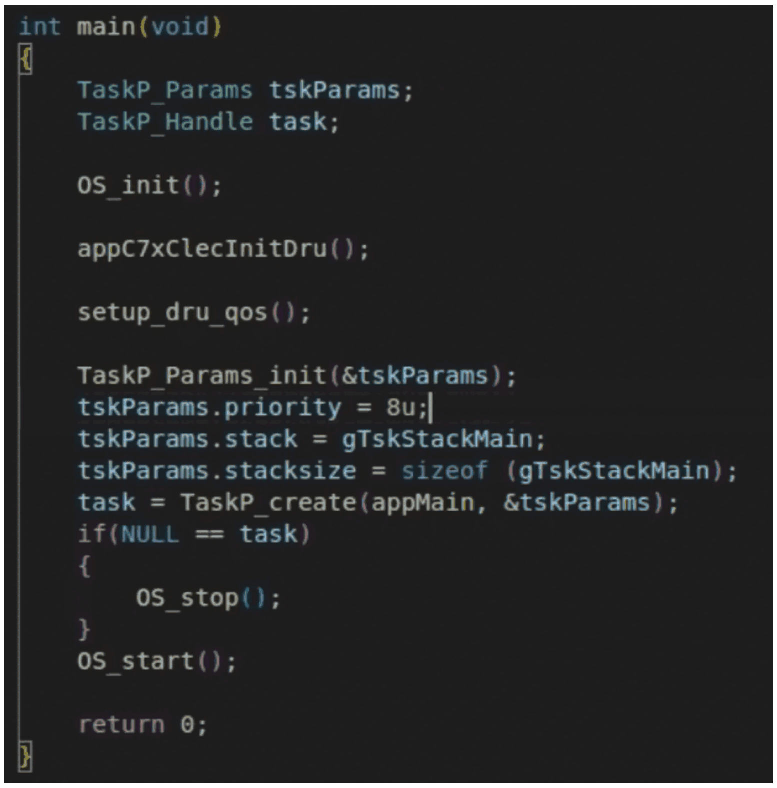
- Compiler of C7X DSP: It mainly provides some inline library functions and functions such as compiling, linking, and forming executable files written by developers.
- Operating system: used for DSP's own task management, memory management, and so forth.
- Low-level related drivers: mainly the code that depends on the implementation of related functions, such as DRU, UDMA, timer and other drivers.
- TIOVX middleware: Provide a calling interface of multi-core heterogeneous accelerators for the entire SOC, and perform parallel computing of heterogeneous accelerators through pipeline settings, thereby accelerating data processing.
- TIDL: TIDL is a comprehensive software product for acceleration of Deep Neural Networks (DNNs) on TI's embedded devices.
- Memory segment management between multi-core heterogeneous: TDA4xx has multiple built-in cores suitable for different application scenarios, and each core can access the DDR space, so it is necessary to manage the memory space to avoid memory space conflicts.

1.1 C7X Compiler

Different C7x DSP compiler versions are provided in different versions of the SDK. The new version of the compiler has some original bug fixes and compilation optimizations, and also provide some new inline functions to use, so basically the new version of the compiler is used. Older versions of software code can be compiled, but older versions of compilers sometimes cannot compile newer versions of software code. Therefore, when upgrading the TIDL software, the compiler needs to be upgraded to a new version, otherwise there will be problems that cannot be compiled. In the use of the compiler, a manual is provided to introduce the use of the compiler, especially the introduction of the relevant compilation instructions. For details, see the *C7000 C/C++Optimizing Compiler Users Guide*.

1.2 Operating System

The C7X DSP is a 64-bit processor with an MMU, but usually a real-time operating system is run. The current SDK version before 8.0 runs the SYSBIOS real-time operating system provided by TI. For details, see the *TI-RTOS Kernel (SYS/BIOS) User's Guide* , Subsequent SDK versions run the Freertos operating system by default. For details, see PDK4.2. The code related to the application layer is not highly dependent on the underlying operating system, so there is no need to upgrade the system to a new version when doing a TIDL upgrade. The main function of the entire operating system is located in *ti-processor-sdk-rtos-j721e-evm-08_04_00_02/ vision_apps/platform/j721e/rtos/c7x_1/main.c*



```
int main(void)
{
    TaskP_Params tskParams;
    TaskP_Handle task;

    OS_init();

    appC7xClecInitDru();

    setup_dru_qos();

    TaskP_Params_init(&tskParams);
    tskParams.priority = 8u;
    tskParams.stack = gTskStackMain;
    tskParams.stacksize = sizeof (gTskStackMain);
    task = TaskP_create(appMain, &tskParams);
    if(NULL == task)
    {
        OS_stop();
    }
    OS_start();

    return 0;
}
```

Figure 1-1. C7X OS Main Function

1.3 Drivers

In addition to running the operating system, C7X DSP may also use hardware resources such as peripherals, so it needs to call the underlying drivers, such as DMA, IPC, and so forth. The underlying related peripheral related drivers are generally stored in the pdk_jacinto_08_04_00_2 directory in the RTOS SDK. You can view the libraries and related drivers that may be used *ti-processor-sdk-rtos-j721e-evm-08_04_00_02/vision_apps/platform/j721e/rtos/concerto_a72_inc.mak* , The driver codes of different versions of peripherals are not very different, so when upgrading TIDL-related software, there is no need to upgrade this. If the previous version of TIDL has related underlying driver dependencies, you can add an additional driver library file to the following three files:

- *ti-processor-sdk-rtos-j721e-evm-08_04_00_02/vision_apps/platform/j721e/rtos/concerto_a72_inc.mak*
- *ti-processor-sdk-rtos-j721e-evm-08_04_00_02/vision_apps/platform/j721e/rtos/c7x_1/concerto_c7x_1_inc.mak*
- *ti-processor-sdk-rtos-j721e-evm-08_04_00_02/vision_apps/platform/j721e/rtos/c7x_1/ concerto.mak*

Add the path where the underlying driver library is located through the IDIRS variable, and add specific library files through *STATIC_LIBS*, *PTK_LIBS*, *TIOVX_LIBS*, *TIDL_LIBS*, *ADDITIONAL_STATIC_LIBS*.

1.4 TIOVX

The TDA4XX processor needs to use the middleware TIOVX to realize the parallel computing of the heterogeneous processors, so the C7x DSP firmware depends on TIOVX. By default, different TIOVX versions have encapsulated TIDL as a target node, so when upgrading TIDL, you do not need to consider upgrading TIOVX. It should be noted that the version after 8.0 introduces the mechanism of TVM deployment algorithm, that is, the unsupported layer is placed on the Arm® side for calculation, and the version after 8.0 supports this mechanism. If you want to use this mechanism in versions before 8.0, you need to upgrade TIOVX. For more details, see the TIOVX usage documentation: *TIOVX User Guide*.

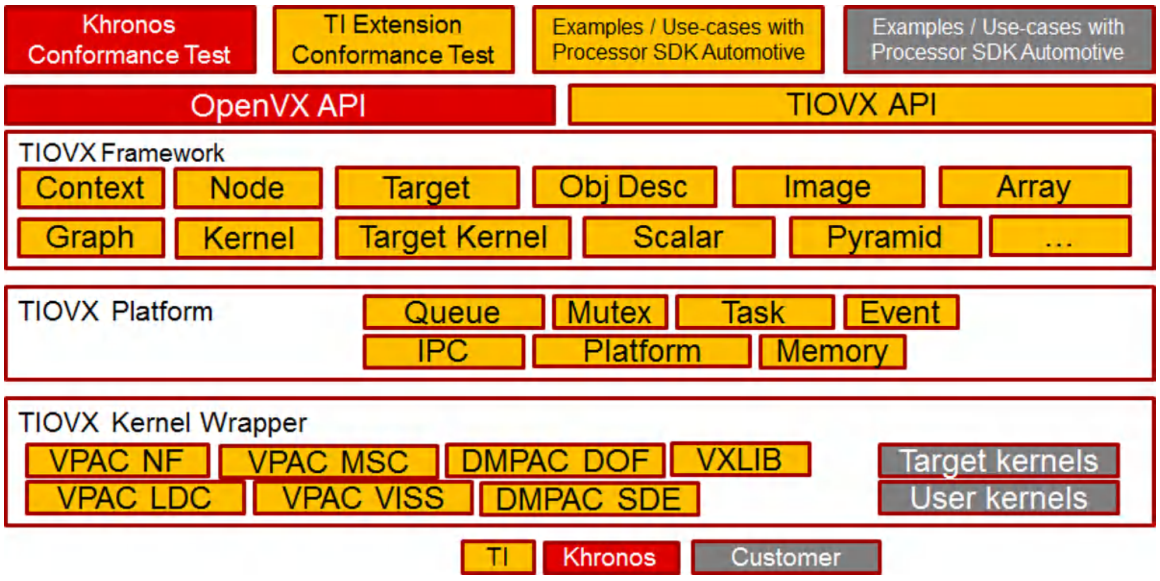


Figure 1-2. TIOVX Block Diagram

1.5 TIDL

TIDL, as an important software library for deploying AI algorithms for TI embedded devices, is iterated and updated in each software version. Each iterative update is mainly to fix bugs in the old version and support the new features of the current deep learning algorithm. TIDL is mainly divided into two parts. The first part is to read, quantify, and convert models such as Tensorflow and ONNX of the current mainstream algorithm framework into the model format that TIDL can execute on the PC side. This part depends on the external open source libraries Opencv, Flatbufffers, and Protobuf, and different versions of the SDK have different version requirements for the three libraries. Therefore, when upgrading, you need to confirm whether to upgrade according to the TIDL User Guide. The second part is to infer the converted model on the board side. The board side inference is mainly to accelerate the calculation of operators. Some operators are calculated using C7X DSP, which is implemented by TIDL, and the other part is calculated using MMA, such as volume Product calculation, Fourier transform are calculated using MMALIB. Therefore, when TIDL is upgraded, MMALIB needs to be upgraded. MMALIB usage reference MMALIB User Guide.

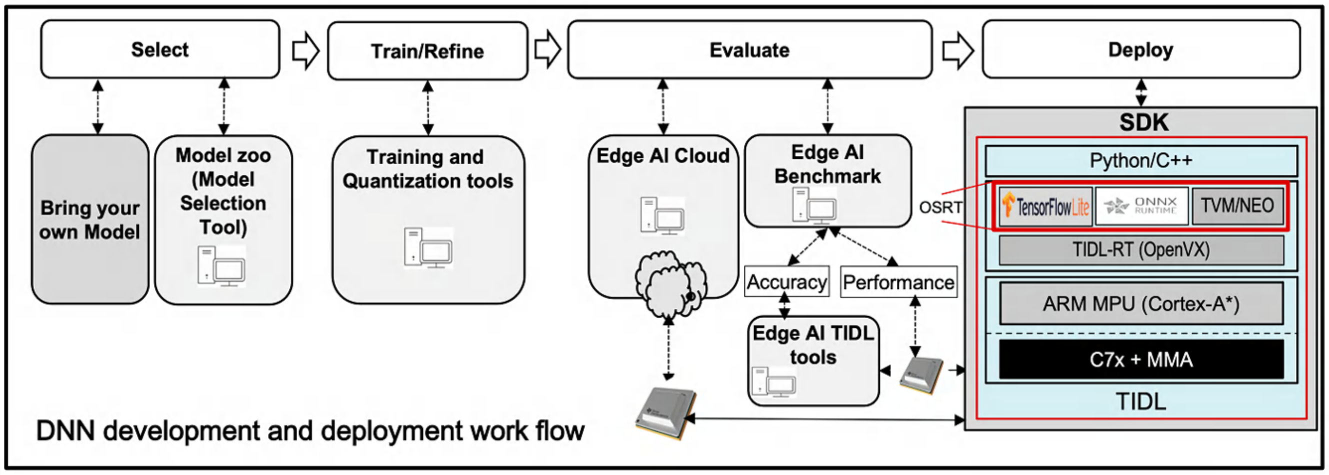
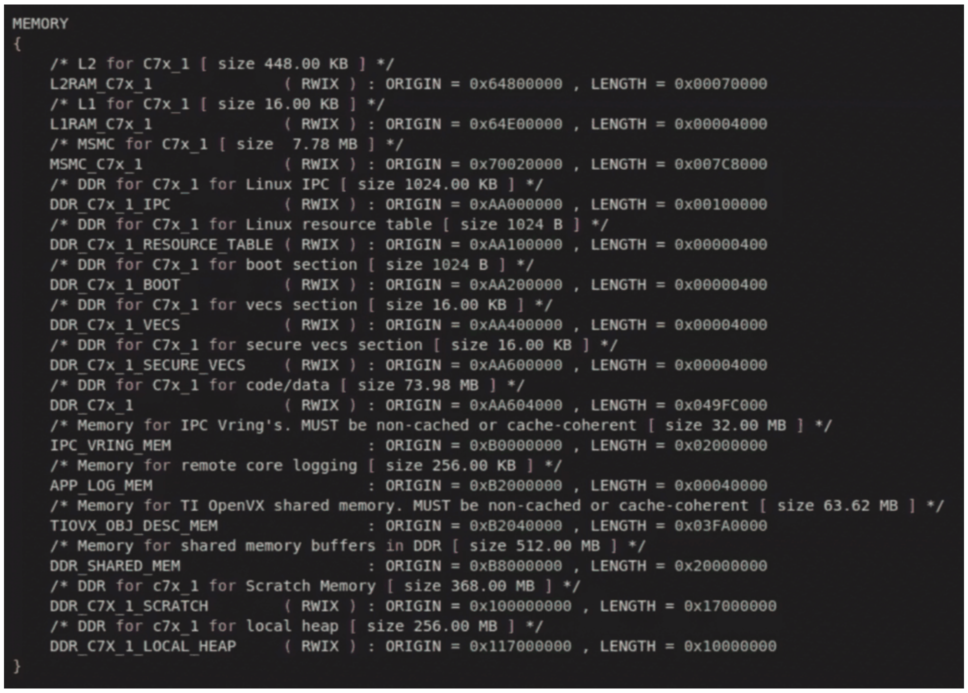


Figure 1-3. TIDL Block Diagram

1.6 Memory Segment Management

TDA4X is a multi-core heterogeneous SOC, multiple cores share the same DDR, and the 32-bit processor can only access the low 2GB *0x0000_8000_0000~ 0x0000_FFFF_FFFF* space, such as R5F, C66, and so forth. But 64-bit processors can access the full memory space, such as A72, C7X. In order to avoid conflicts in memory usage, you need to divide and manage the DDR memory space. In the SDK, the memory of each processor is divided and managed through python scripts. For details, see *ti-processor-sdk-rtos-j721e-evm-08_04_00_02/vision_apps/platform/j721e/rtos/gen_linker_mem_map.py* , The memory management of A72 is not in this document. If the A72 runs Linux, the memory management is set in DTS. Reference documentation for the use of memory management tools Understanding and updating SDK memory map for J721E. The memory segment of C7X DSP is divided as follows:



```
MEMORY
{
    /* L2 for C7x_1 [ size 448.00 KB ] */
    L2RAM_C7x_1              ( RWIX ) : ORIGIN = 0x64800000 , LENGTH = 0x00070000
    /* L1 for C7x_1 [ size 16.00 KB ] */
    L1RAM_C7x_1              ( RWIX ) : ORIGIN = 0x64E00000 , LENGTH = 0x00004000
    /* MSMC for C7x_1 [ size  7.78 MB ] */
    MSMC_C7x_1               ( RWIX ) : ORIGIN = 0x70020000 , LENGTH = 0x007C8000
    /* DDR for C7x_1 for Linux IPC [ size 1024.00 KB ] */
    DDR_C7x_1_IPC            ( RWIX ) : ORIGIN = 0xAA000000 , LENGTH = 0x00100000
    /* DDR for C7x_1 for Linux resource table [ size 1024 B ] */
    DDR_C7x_1_RESOURCE_TABLE ( RWIX ) : ORIGIN = 0xAA100000 , LENGTH = 0x00000400
    /* DDR for C7x_1 for boot section [ size 1024 B ] */
    DDR_C7x_1_BOOT           ( RWIX ) : ORIGIN = 0xAA200000 , LENGTH = 0x00000400
    /* DDR for C7x_1 for vecs section [ size 16.00 KB ] */
    DDR_C7x_1_VECS           ( RWIX ) : ORIGIN = 0xAA400000 , LENGTH = 0x00004000
    /* DDR for C7x_1 for secure vecs section [ size 16.00 KB ] */
    DDR_C7x_1_SECURE_VECS    ( RWIX ) : ORIGIN = 0xAA600000 , LENGTH = 0x00004000
    /* DDR for C7x_1 for code/data [ size 73.98 MB ] */
    DDR_C7x_1                ( RWIX ) : ORIGIN = 0xAA604000 , LENGTH = 0x049FC000
    /* Memory for IPC Vring's. MUST be non-cached or cache-coherent [ size 32.00 MB ] */
    IPC_VRING_MEM                     : ORIGIN = 0xB0000000 , LENGTH = 0x02000000
    /* Memory for remote core logging [ size 256.00 KB ] */
    APP_LOG_MEM                       : ORIGIN = 0xB2000000 , LENGTH = 0x00040000
    /* Memory for TI OpenVX shared memory. MUST be non-cached or cache-coherent [ size 63.62 MB ] */
    TIOVX_OBJ_DESC_MEM                : ORIGIN = 0xB2040000 , LENGTH = 0x03FA0000
    /* Memory for shared memory buffers in DDR [ size 512.00 MB ] */
    DDR_SHARED_MEM                    : ORIGIN = 0xB8000000 , LENGTH = 0x20000000
    /* DDR for c7x_1 for Scratch Memory [ size 368.00 MB ] */
    DDR_C7X_1_SCRATCH        ( RWIX ) : ORIGIN = 0x100000000 , LENGTH = 0x17000000
    /* DDR for c7x_1 for local heap [ size 256.00 MB ] */
    DDR_C7X_1_LOCAL_HEAP     ( RWIX ) : ORIGIN = 0x117000000 , LENGTH = 0x10000000
}
```

Figure 1-4. C7X Memory Map

2 TIDL Upgrade

This example demonstrates how to upgrade TIDL from SDK 7.1 to 8.4 on TDA4VM. The upgrade methods for other versions are similar. This upgrade mainly includes the upgrade of TIDL, MMALIB, and C7 compilers, mainly to fix some bugs. The specific bug fixes can be viewed in the TIDL release note and MMALIB release note of each version. There are some new operators and new features are support such as batch processing, and so forth.

2.1 RTOS SDK Changes

1. You need to remove the old version of the software, delete or move it to another directory, this example directly deletes the related software that the old version of TIDL depends on:
    a. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11$ rm -rf ./ti-cgt-c7000_1.4.0.LTS*
    b. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11$ rm -rf ./tidl_j7_01_03_00_11*
    c. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11$ rm -rf ./mmalib_01_03_00_06*
2. Copy the TIDL related software in the 8.4SDK to the 7.1SDK directory:
    a. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11$ cp ../../ti-8.4/ti-processor-sdk-rtos-j721e-evm-08_04_00_02/ti-cgt-c7000_3.0.0.STS ./ti-cgt-c7000_1.4.0.LTS -r*
    b. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11$ cp ../../ti-8.4/ti-processor-sdk-rtos-j721e-evm-08_04_00_02/mmalib_02_04_00_06 ./mmalib_01_03_00_06 -r*
    c. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11$ cp ../../ti-8.4/ti-processor-sdk-rtos-j721e-evm-08_04_00_02/tidl_j721e_08_04_00_12 ./tidl_j7_01_03_00_11 -r*

3. The path of the library generated by TIDL and MMALIB in 8.4SDK has changed, so the path needs to be modified during the upgrade. The details are as follows:

```
diff --git a/tidl_j7_01_03_00_11/makerules/config.mk b/tidl_j7_01_03_00_11/makerules/config.mk
index 6a67f858..af1c084e 100644
--- a/tidl_j7_01_03_00_11/makerules/config.mk
+++ b/tidl_j7_01_03_00_11/makerules/config.mk
@@ -99,20 +99,20 @@ BUILD_LIDAR_PREPROC    ?= 0
 ifdef SystemRoot
 PSDK_INSTALL_PATH ?= C:\ti
 else
-PSDK_INSTALL_PATH ?= /ti/j7presi/workarea/
+PSDK_INSTALL_PATH = $(pwd)/../../
 endif

 ifdef SystemRoot
-BIOS_PATH              ?="$(PSDK_INSTALL_PATH)\bios_6_83_02_07"
-XDCTOOLS_PATH          ?="$(PSDK_INSTALL_PATH)\xdctools_3_61_04_40_core"
+BIOS_PATH              ?="$(PSDK_INSTALL_PATH)\bios_6_83_00_18"
+XDCTOOLS_PATH          ?="$(PSDK_INSTALL_PATH)\xdctools_3_61_03_29_core"
 IVISION_PATH          ?="$(PSDK_INSTALL_PATH)\ivision"
 UTILS_PATH            ?="$(PSDK_INSTALL_PATH)\ccs910\ccs\utils\cygwin"
 TIDL_PROTOBUF_PATH    ?="$(PSDK_INSTALL_PATH)\protobuf-3.11.3_msvc_2015_x64"
 TIDL_OPENCV_PATH      ?=$(PSDK_INSTALL_PATH)\opencv_3.1.0_msvc_2015_x64\opencv\sources
-DSP_TOOLS              ?=$(PSDK_INSTALL_PATH)\ti-cgt-c7000_3.0.0.STS
+DSP_TOOLS              ?=$(PSDK_INSTALL_PATH)\ti-cgt-c7000_1.4.0.LTS
 TIDL_FLATBUF_PATH     ?=$(PSDK_INSTALL_PATH)\flatbuffers-1.12.0
-MMALIB_PATH            ?=$(PSDK_INSTALL_PATH)\mmalib_02_04_00_05
-PDK_INSTALL_PATH       ?=$(PSDK_INSTALL_PATH)\pdk_jacinto_08_04_00_21\packages
+MMALIB_PATH            ?=$(PSDK_INSTALL_PATH)\mmalib_01_03_00_06
+PDK_INSTALL_PATH       ?=$(PSDK_INSTALL_PATH)\pdk_jacinto_07_01_00_45\packages
 CONCERTO_ROOT         ?=$(PSDK_INSTALL_PATH)\vision_apps\concerto
 TIOVX_PATH            ?=$(PSDK_INSTALL_PATH)\tiovx
 VISION_APPS_PATH      ?=$(PSDK_INSTALL_PATH)\vision_apps
@@ -122,12 +122,12 @@ TIDL_GRAPHVIZ_PATH  ?="$(PSDK_INSTALL_PATH)\graphviz-2.38_x64"
 CUDA_PATH ?= "C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v9.0/"
 CUDNN_PATH ?= "D:/work/cuDNN/cudnn-9.0-v7.0/"
 else
-BIOS_PATH              ?="$(PSDK_INSTALL_PATH)/bios_6_83_02_07"
-XDCTOOLS_PATH          ?="$(PSDK_INSTALL_PATH)/xdctools_3_61_04_40_core"
+BIOS_PATH              ?="$(PSDK_INSTALL_PATH)/bios_6_83_00_18"
+XDCTOOLS_PATH          ?="$(PSDK_INSTALL_PATH)/xdctools_3_61_03_29_core"
 IVISION_PATH          ?="$(PSDK_INSTALL_PATH)/ivision"
-DSP_TOOLS              ?="$(PSDK_INSTALL_PATH)/ti-cgt-c7000_3.0.0.STS"
-MMALIB_PATH            ?="$(PSDK_INSTALL_PATH)/mmalib_02_04_00_05"
-PDK_INSTALL_PATH       ?="$(PSDK_INSTALL_PATH)/pdk_jacinto_08_04_00_21/packages"
+DSP_TOOLS              ?="$(PSDK_INSTALL_PATH)/ti-cgt-c7000_1.4.0.LTS"
+MMALIB_PATH            ?="$(PSDK_INSTALL_PATH)/mmalib_01_03_00_06"
+PDK_INSTALL_PATH       ?="$(PSDK_INSTALL_PATH)/pdk_jacinto_07_01_00_45/packages"
 CONCERTO_ROOT         ?=$(PSDK_INSTALL_PATH)/vision_apps/concerto
 TIOVX_PATH            ?=$(PSDK_INSTALL_PATH)/tiovx
 VISION_APPS_PATH      ?=$(PSDK_INSTALL_PATH)/vision_apps
@@ -161,7 +161,7 @@ CORE ?= dsp

 # Default RTOS is FREERTOS
 # Supported values: FREERTOS, SYSBIOS
-RTOS ?= FREERTOS
+RTOS ?= SYSBIOS

diff --git a/vision_apps/apps/basic_demos/app_tirtos/concerto_c7x_inc.mak b/vision_apps/apps/
basic_demos/app_tirtos/concerto_c7x_inc.mak
index 4c30e077..38530b96 100755
--- a/vision_apps/apps/basic_demos/app_tirtos/concerto_c7x_inc.mak
+++ b/vision_apps/apps/basic_demos/app_tirtos/concerto_c7x_inc.mak
@@ -17,8 +17,8 @@ LDIRS += $(PDK_PATH)/packages/ti/drv/sciclient/lib/j721e/c7x_1/$(TARGET_BUILD)/
 LDIRS += $(TIOVX_PATH)/lib/$(TARGET_PLATFORM)/$(TARGET_CPU)/$(TARGET_OS)/$(TARGET_BUILD)
 LDIRS += $(PTK_PATH)/lib/$(TARGET_PLATFORM)/$(TARGET_CPU)/$(TARGET_OS)/$(TARGET_BUILD)
 LDIRS += $(VISION_APPS_PATH)/lib/$(TARGET_PLATFORM)/$(TARGET_CPU)/$(TARGET_OS)/$(TARGET_BUILD)
-LDIRS += $(MMALIB_PATH)/lib/release
-LDIRS += $(TIDL_PATH)/lib/dsp/algo/release
+LDIRS += $(MMALIB_PATH)/lib/C7100/release
+LDIRS += $(TIDL_PATH)/lib/j721e/dsp/algo/release

 STATIC_LIBS += app_utils_mem
 STATIC_LIBS += app_utils_console_io
```

TIOVX has made some upgrade changes in version 8.4 compared to version 7.1, so it needs to be modified to adapt to 7.1.

```
diff --git a/tidl_j7_01_03_00_11/ti_dl/rt/src/a72/concerto.mak b/tidl_j7_01_03_00_11/
ti_dl/rt/src/a72/concerto.mak
index 33c15839..978f94da 100644
--- a/tidl_j7_01_03_00_11/ti_dl/rt/src/a72/concerto.mak
+++ b/tidl_j7_01_03_00_11/ti_dl/rt/src/a72/concerto.mak
@@ -20,20 +20,18 @@ LDIRS += $(LINUX_FS_PATH)/usr/lib

 TIOVX_LIBS  =
 TIOVX_LIBS += vx_framework
-TIOVX_LIBS += vx_platform_psdk_j7
+TIOVX_LIBS += vx_platform_psdk_j7_linux
 TIOVX_LIBS += vx_kernels_host_utils
 TIOVX_LIBS += vx_kernels_tidl

-ifeq ($(TARGET_PLATFORM), J7)
-TIOVX_LIBS += vx_kernels_tvm
-endif
+

 TIOVX_LIBS += vx_kernels_openvx_core
 TIOVX_LIBS += vx_utils

 VISION_APPS_UTILS_LIBS += app_utils_console_io
 VISION_APPS_UTILS_LIBS += app_utils_ipc
-VISION_APPS_UTILS_LIBS += app_rtos_linux_mpu1_common
+VISION_APPS_UTILS_LIBS += app_tirtos_linux_mpu1_common
 VISION_APPS_UTILS_LIBS += app_utils_remote_service
 VISION_APPS_UTILS_LIBS += app_utils_mem
 VISION_APPS_UTILS_LIBS += app_utils_perf_stats
diff --git a/tidl_j7_01_03_00_11/ti_dl/rt/src/concerto_common.mak b/tidl_j7_01_03_00_11/
ti_dl/rt/src/concerto_common.mak
index 8c09bc08..e63f0fbc 100644
--- a/tidl_j7_01_03_00_11/ti_dl/rt/src/concerto_common.mak
+++ b/tidl_j7_01_03_00_11/ti_dl/rt/src/concerto_common.mak
@@ -4,13 +4,8 @@ CFLAGS += -fPIC -Wno-int-to-pointer-cast -Wno-stringop-truncation -Wno-format-ov
 CPPFLAGS += -fPIC --std=c++11

 CSOURCES    += ../tidl_rt_ovx.c
-CSOURCES    += ../tidl_rt_utils.c
-CSOURCES    += ../tidl_rt_ovx_debug_utils.c
-CPPSOURCES  += ../tidl_rt_ovx_datamove.cpp

-ifeq ($(TARGET_PLATFORM), J7)
-CSOURCES    += ../tvm_rt_ovx.c
-endif
+

 IDIRS :=
 IDIRS += $(IVISION_PATH)
diff --git a/tidl_j7_01_03_00_11/ti_dl/rt/src/tidl_rt_ovx.c b/tidl_j7_01_03_00_11/ti_dl/rt/src/
tidl_rt_ovx.c
index 3a659729..234d6109 100644
--- a/tidl_j7_01_03_00_11/ti_dl/rt/src/tidl_rt_ovx.c
+++ b/tidl_j7_01_03_00_11/ti_dl/rt/src/tidl_rt_ovx.c
@@ -67,13 +67,6 @@
 #include <tivx_utils_graph_perf.h>
 #include <tivx_utils_tidl_trace.h>

-#if   defined (SOC_J721E)
-#include <TI/tivx_soc_j721e.h>
-#elif defined (SOC_J721S2)
-#include <TI/tivx_soc_j721s2.h>
-#elif defined (SOC_J784S4)
-#include <TI/tivx_soc_j784s4.h>
-#endif

 #include <stdio.h>
 #include <stdint.h>
@@ -86,6 +79,69 @@
 #include <math.h>
 #include "itidl_rt.h"
 #include "tidl_rt_ovx_utils.h"
+/*! \brief Target name for DSP_C7_1
+ *         This target task is first in priority.
+ *         Each of the C7X targets are assigned a different
+ *         task priority.  Subsequent C7X targets are assigned
+ *         lower priority than the preceding target (i.e.,
+ *         \ref TIVX_TARGET_DSP_C7_1_PRI_2 is lower priority
+ *         than \ref TIVX_TARGET_DSP_C7_1_PRI_1).  Therefore,
+ *         the \ref TIVX_TARGET_DSP_C7_1_PRI_2 target will be
+ *         preempted by \ref TIVX_TARGET_DSP_C7_1_PRI_1 once
+ *         the higher priority target is unblocked to execute.
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1    "DSP_C7-1"
+
+/*! \brief Target name for DSP_C7_1
+ *         This target task is first in priority.
+ *         This aliases to the same task as \ref TIVX_TARGET_DSP_C7_1
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1_PRI_1    TIVX_TARGET_DSP_C7_1
+
+/*! \brief Target name for DSP_C7-1_PRI_2
+ *         This target task is second in priority
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1_PRI_2    "DSP_C7-1_PRI_2"
+
+/*! \brief Target name for DSP_C7-1_PRI_3
+ *         This target task is third in priority
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1_PRI_3    "DSP_C7-1_PRI_3"
+
+/*! \brief Target name for DSP_C7-1_PRI_4
+ *         This target task is fourth in priority
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1_PRI_4    "DSP_C7-1_PRI_4"
+
+/*! \brief Target name for DSP_C7-1_PRI_5
+ *         This target task is fifth in priority
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1_PRI_5    "DSP_C7-1_PRI_5"
+
+/*! \brief Target name for DSP_C7-1_PRI_6
+ *         This target task is sixth in priority
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1_PRI_6    "DSP_C7-1_PRI_6"
+
+/*! \brief Target name for DSP_C7-1_PRI_7
+ *         This target task is seventh in priority
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1_PRI_7    "DSP_C7-1_PRI_7"
+
+/*! \brief Target name for DSP_C7-1_PRI_8
+ *         This target task is eighth in priority
+ * \ingroup group_tivx_ext_targets
+ */
+#define TIVX_TARGET_DSP_C7_1_PRI_8    "DSP_C7-1_PRI_8"
+
 extern char* strdup(const char*);

 #define TIVX_TIDL_TRACE_DATA_SIZE  (256 * 1024 * 1024)
diff --git a/tiovx/kernels_j7/include/TI/j7_tidl.h b/tiovx/kernels_j7/include/TI/j7_tidl.h
index 855672b0..654df4f8 100644
--- a/tiovx/kernels_j7/include/TI/j7_tidl.h
+++ b/tiovx/kernels_j7/include/TI/j7_tidl.h
@@ -106,6 +106,9 @@ typedef struct{
   /**TIDL input/output buffer descriptor*/
   sTIDL_IOBufDesc_t ioBufDesc;

+  /** Flag to enable optimization ivision alg activate, default it is disabled */
+  vx_uint32 optimize_ivision_activation;
+
 }tivxTIDLJ7Params;
```

The 8.4 TIDL depends on the *app_utils_init* library, so it needs to be copied to the 7.1SDK:

- *ti-processor-sdk-rtos-j721e-evm-07_01_00_11$ cp ./../ti-8.4/ti-processor-sdk-rtos-j721e-evm-08_04_00_02/ vision_apps/utils/app_init ./vision_apps/utils/ -r*

In addition to modifying the relevant paths, the latest version of TIDL code has been added, causing the memory segment to need to be re-allocated, otherwise an error will be reported using the default memory allocation of 7.1SDK:

1. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11/vision_apps/tools/PyTI_PSDK_RTOS$ pip3 install -e . --user*
2. Modify *vision_apps/apps/basic_demos/app_tirtos/tirtos_linux/gen_linker_mem_map.py* as follows:

```
diff --git a/vision_apps/apps/basic_demos/app_tirtos/tirtos_linux/gen_linker_mem_map.py b/
vision_apps/apps/basic_demos/app_tirtos/tirtos_linux/gen_linker_mem_map.py
index eba9d84b..871fe762 100755
--- a/vision_apps/apps/basic_demos/app_tirtos/tirtos_linux/gen_linker_mem_map.py
+++ b/vision_apps/apps/basic_demos/app_tirtos/tirtos_linux/gen_linker_mem_map.py
@@ -200,7 +200,7 @@ c7x_1_ddr_vecs_size = 16*KB;
 c7x_1_ddr_secure_vecs_addr = c7x_1_ddr_resource_table_addr + 5*MB;
 c7x_1_ddr_secure_vecs_size = 16*KB;
 c7x_1_ddr_addr = c7x_1_ddr_secure_vecs_addr + c7x_1_ddr_secure_vecs_size;
-c7x_1_ddr_size = 16*MB - (c7x_1_ddr_addr-c7x_1_ddr_ipc_addr);
+c7x_1_ddr_size = 32*MB - (c7x_1_ddr_addr-c7x_1_ddr_ipc_addr);

 #
 # DDR memory allocation for various shared memories
@@ -256,7 +256,7 @@ c66x_2_ddr_scratch_addr    = c66x_2_ddr_local_heap_addr +
c66x_2_ddr_local_heap_
 c66x_2_ddr_scratch_size    = 48*MB;

 c7x_1_ddr_local_heap_addr  = c66x_2_ddr_scratch_addr + c66x_2_ddr_scratch_size;
-c7x_1_ddr_local_heap_size  = 256*MB;
+c7x_1_ddr_local_heap_size  = 240*MB;
 c7x_1_ddr_scratch_addr     = c7x_1_ddr_local_heap_addr + c7x_1_ddr_local_heap_size;
 c7x_1_ddr_scratch_size     = ddr_mem_size - 32*MB - (c7x_1_ddr_scratch_addr - ddr_mem_addr);
```

3. Execute the following command to generate a new memory management segment *ti-processor-sdk-rtos-j721e-evm-07_01_00_11/vision_apps/apps/basic_demos/app_tirtos/tirtos_linux$ ./gen_linker_mem_map.py*.

The complete patch of the entire RTOS SDK is as follows: 0001-7.1SDK-upgrade-8.4-TIDL.patch.

2.2 TIDL PC Tool Changes

The 8.4 version of the PC tool relies on Open CV 4.1.0, protobuf 3.11.3, flatbuffers-1.12.0, and needs to be upgraded to the corresponding version. The specific compilation is as follows: TIDL build Instruction

2.3 Linux SDK Changes

The changes of the Linux SDK are mainly aimed at the modification of the C7 memory segment, which needs to correspond to it in the Linux DTS, otherwise an error occura when running on the board.
The main modified files are: *board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/boot/dts/ti/k3-j721e-som-p0.dtsi* and *board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/boot/dts/ti/k3-j721e-vision-apps.dtso*.

Patch file *0001-upgrade-TIDL-to-8.4-memory-map-changes.patch*, the complete changes are as follows:

```
diff --git a/board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/boot/dts/ti/
k3-j721e-som-p0.dtsi b/board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/
boot/dts/ti/k3-j721e-som-p0.dtsi
index 90bcc6be3..d4ea9a533 100644
--- a/board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/boot/dts/ti/k3-j721e-
som-p0.dtsi
+++ b/board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/boot/dts/ti/k3-j721e-
som-p0.dtsi
@@ -130,12 +130,12 @@

         c71_0_memory_region: c71-memory@a8100000 {
             compatible = "shared-dma-pool";
-            reg = <0x00 0xa8100000 0x00 0xf00000>;
+            reg = <0x00 0xa8100000 0x00 0x01f00000>;
             no-map;
         };
```

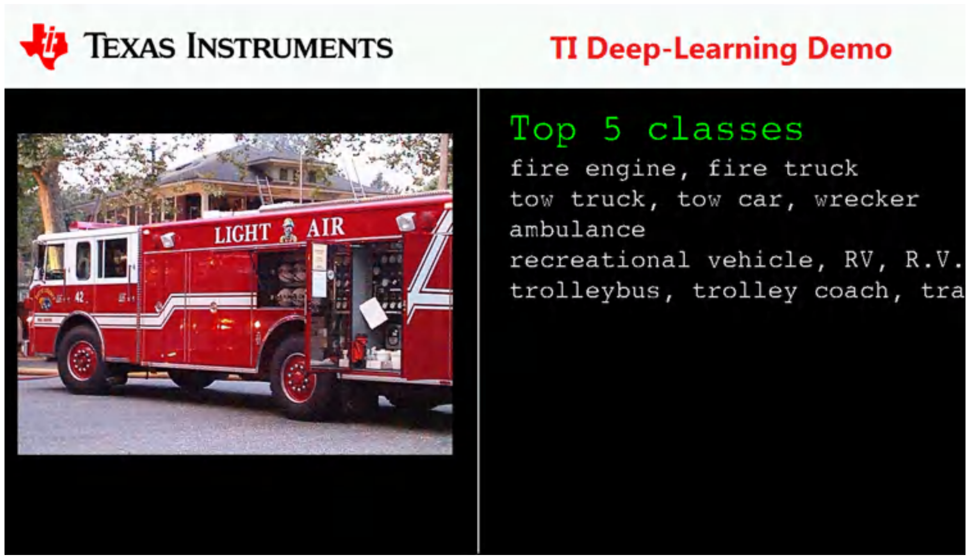
```
-         rtos_ipc_memory_region: ipc-memories@aa000000 {
-             reg = <0x00 0xaa000000 0x00 0x01c00000>;
+         rtos_ipc_memory_region: ipc-memories@ab000000 {
+             reg = <0x00 0xab000000 0x00 0x02000000>;
              alignment = <0x1000>;
              no-map;
          };
diff --git a/board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/boot/dts/ti/
k3-j721e-vision-apps.dtso b/board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/
boot/dts/ti/k3-j721e-vision-apps.dtso
index f2719997f..13c2b782f 100644
--- a/board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/boot/dts/ti/k3-j721e-
vision-apps.dtso
+++ b/board-support/linux-5.4.74+gitAUTOINC+9574bba32a-g9574bba32a/arch/arm64/boot/dts/ti/k3-j721e-
vision-apps.dtso
@@ -102,18 +102,18 @@
          reg = <0x00 0xa5900000 0x00 0x00700000>;
          no-map;
      };
-     vision_apps_memory_region: vision_apps-dma-memory@ac000000 {
+     vision_apps_memory_region: vision_apps-dma-memory@ad000000 {
          compatible = "shared-dma-pool";
-         reg = <0x00 0xac000000 0x00 0x02000000>;
+         reg = <0x00 0xad000000 0x00 0x02000000>;
          no-map;
      };
      vision_apps_shared_region: vision_apps_shared-memories {
          compatible = "dma-heap-carveout";
-         reg = <0x00 0xae000000 0x00 0x20000000>;
+         reg = <0x00 0xaf000000 0x00 0x20000000>;
      };
-     vision_apps_core_heaps: vision_apps-core-heap-memory@ce000000 {
+     vision_apps_core_heaps: vision_apps-core-heap-memory@cf000000 {
          compatible = "shared-dma-pool";
-         reg = <0x00 0xce000000 0x00 0x2d000000>;
+         reg = <0x00 0xcf000000 0x00 0x2c000000>;
          no-map;
      };
      vision_apps_mcu_r5fss0_core1_dma_memory_region: vision_apps-r5f-dma-memory@fb000000 {
--
2.17.1
```

3 Demo Verify

1. Make a pre-build SD card:
   a. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11-prebuilt$ sudo ./mk-linux-card.sh /dev/sdc*
   b. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11-prebuilt$./install_to_sd_card*
   c. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11-prebuilt$./install_data_set_to_sd_card.sh ../psdk_rtos_ti_data_set_07_01_00.tar.gz*
2. Save the previous changes, compile the SDK, and update the SD card:
   a. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11/vision_apps$ make sdk -j8*
   b. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11/vision_apps$ make linux_fs_install_sd*
   c. *ti-processor-sdk-linux-j7-evm-07_01_00_10$ make all -j8*
   d. *ti-processor-sdk-linux-j7-evm-07_01_00_10$ make install*
   e. *ti-processor-sdk-linux-j7-evm-07_01_00_10$ cp ./targetNFS/boot/* /media/$USER/rootfs/boot/*
3. Download the mobileNetv1 model for test:
   a. git clone https://github.com/shicai/MobileNet-Caffe.git
4. Copy the model to the corresponding directory to generate a model format that TIDL can execute:
   a. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11/tidl_j7_01_03_00_11/ti_dl/test/testvecs/models/public/caffe$ mkdir mobileNet1.0v1&&cd mobileNet1.0v1/*
   b. *cp ~/Desktop/MobileNet-Caffe/mobilenet.caffemodel ./*
   c. *cp ~/Desktop/MobileNet-Caffe/mobilenet_deploy.prototxt ./*
   d. *ti-processor-sdk-rtos-j721e-evm-07_01_00_11/tidl_j7_01_03_00_11/ti_dl/utils/tidlModelImport$ ./out/tidl_model_import.out ../../../ti_dl/test/testvecs/config/import/public/caffe/tidl_import_mobilenet_v1.txt*

```
Caffe Network File : ../../test/testvecs/models/public/caffe/mobileNet1.0v1/
mobilenet_deploy.prototxt
Caffe Model File   : ../../test/testvecs/models/public/caffe/mobileNet1.0v1/
mobilenet.caffemodel
TIDL Network File  : ../../test/testvecs/config/tidl_models/caffe/tidl_net_mobilenet_v1.bin
TIDL IO Info File  : ../../test/testvecs/config/tidl_models/caffe/tidl_io_mobilenet_v1_

Name of the Network :        MOBILENET

~~~~~Running TIDL in PC emulation mode to collect Activations range for each layer~~~~~

Processing config file #0 : /home/test/Desktop/pc_disk/ti_sdk/TDA4VM/ti-7.1/ti-processor-
sdk-rtos-j721e-evm-07_01_00_11/tidl_j7_01_03_00_11/ti_dl/test/testvecs/config/tidl_models/caffe/
tidl_import_mobilenet_v1.txt.qunat_stats_config.txt
 ----------------------- TIDL Process with REF_ONLY FLOW ------------------------

#    0 . .. T     280.71  .... ..... ... A :   895, 1.0000, 1.0000,    895 .... .....
#    1 . .. T     279.96  .... ..... ... A :   557, 0.5000, 0.5000,    829 .... .....
#    2 . .. T     263.92  .... ..... ... A :   442, 0.3333, 0.3333,    829 .... .....
#    3 . .. T     260.39  .... ..... ... A :   498, 0.2500, 0.2500,    751 .... .....
~~~~~Running TIDL in PC emulation mode to collect Activations range for each layer~~~~~

Processing config file #0 : /home/test/Desktop/pc_disk/ti_sdk/TDA4VM/ti-7.1/ti-processor-
sdk-rtos-j721e-evm-07_01_00_11/tidl_j7_01_03_00_11/ti_dl/test/testvecs/config/tidl_models/caffe/
tidl_import_mobilenet_v1.txt.qunat_stats_config.txt
 ----------------------- TIDL Process with REF_ONLY FLOW ------------------------

#    0 . .. T     566.09  .... ..... ... A :   895, 1.0000, 1.0000,    895 .... .....
#    1 . .. T     556.95  .... ..... ... A :   557, 0.5000, 1.0000,    733 .... .....
#    2 . .. T     553.00  .... ..... ... A :   442, 0.3333, 0.6667,    675 .... .....
#    3 . .. T     559.57  .... ..... ... A :   498, 0.2500, 0.5000,    833 .... .....
 *****************   Calibration iteration number 0 completed ************************
------------------ Network Compiler Traces -----------------------------
successful Memory allocation
****************************************************
**                ALL MODEL CHECK PASSED          **
****************************************************
```

5. Copy the newly generated model to the board side:
   a. *cp ti-processor-sdk-rtos-j721e-evm-07_01_00_11/tidl_j7_01_03_00_11/ti_dl/test/testvecs/config/tidl_models/caffe/ tidl_io_mobilenet_v1_1.bin /media/$USER/rootfs/opt/vision_apps/test_data/tivx/tidl_models/*
   b. *cp ti-processor-sdk-rtos-j721e-evm-07_01_00_11/tidl_j7_01_03_00_11/ti_dl/test/testvecs/config/tidl_models/caffe/ tidl_net_mobilenet_v1.bin /media/$USER/rootfs/opt/vision_apps/test_data/tivx/tidl_models/*
6. Run demo on TDA4EVM:
   a. *cd /opt/vision_apps*
   b. *source ./vision_apps_init.sh*
   c. *./run_app_tidl.sh*



Top 5 classes
fire engine, fire truck
tow truck, tow car, wrecker
ambulance
recreational vehicle, RV, R.V.
trolleybus, trolley coach, tra

Figure 3-1. Image Classification Result

4 Summary

This article takes 7.1SDK as an example to upgrade TIDL to version 8.4 and even though it is expected to work for other releases also but may need slight modification if you required. The upgrade methods for other versions are similar, and you can make your own if the upgraded version is the same as the version in the document, after copying the TIDL, MMALIB, and C7 compilers of 8.4SDK, you only need to use the two patch applications provided in the text to recompile the SDK. If the upgraded version is inconsistent with the text, you can refer to the steps in the text to generate the corresponding patch. It should be noted that after the upgrade, the previous version of the model cannot be run in the upgraded SDK, and the corresponding board model file must be re-imported.

5 References

- Texas Instruments: *C7000 C/C++Optimizing Compiler Users Guide*
- Texas Instruments: *TI-RTOS Kernel (SYS/BIOS) User's Guide*
- PDK4.2
- *TIOVX User Guide*
- Understanding and updating SDK memory map for J721E
- TIDL User Guide
- MMALIB User Guide
- TIDL release note
- MMALIB release note
- Vision Apps User Guide