department of
signal processing

https://sp.utia.cas.cz

Application Note

ÚTIA Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

Compilation of AI 3.0 models for Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8V

Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, Raissa Likhonina
kadlec@utia.cas.cz, zdenek.pohl@utia.cas.cz, *kohoutl@utia.cas.cz, likhonina@utia.cas.cz*

Revision history

Rev.	Date	Author	Description
v01	20.5.2024	J.K.	Vitis 2023.2, Petalinux 2023.2, AI 3.5 runtime, AI 3.0 models for AMD DPU DPUCZDX8V
v02	5.12.2024	J.K.	Update for bring-up script released by Trenz electronic 25.11.2024 TE0820-test_board-vivado_2023.2-build_4_20241125214948.zip
v03	15.02.2025	J.K.	Fixed typos.
v04	16.02.2025	J.K.	Fixed references.

Contents

Acknowledgement

The EECONE project is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Funding Authorities from involved countries under grant agreement no. 101112065.

https://zs.utia.cas.cz/index.php?ids=projects/eecone

https://eecone.com/eecone/home/

1 Introduction

EECONE project https://eecone.com/eecone/home/ work package 4, task 4.3 is investigating measures to support second life of electronics due to modular design.

Work package 4 task 4.4 is investigating measures to support extension of life of electronics due to methodology of support used custom platform to adapt for the in-time-evolving design tools and embedded Linux PetaLinux operating system.

UTIA AV CR, v.v.i. (Institute of Information Theory and Automation of the Czech Academy of Sciences, in short UTIA) is not-for profit research institute located in Prague, Czech Republic. UTIA is involved as partner in both tasks, T4.3 and T4.4.

Both EECONE tasks require specification of comparable reference systems which are based on modular HW with potential for “second life” by reuse of modules or use cost optimized PCB HW without modularity.

Systems (with HW modularity or low cost single PCB) should be capable to perform similar challenging tasks. Systems have to be capable to accelerate in HW AI inference algorithms with video camera input for edge application like person detection, face detection, car-make or car-type detection and graphical output to on X11 desktop of a remote PC connected by wired Ethernet in a local network.

Systems should also support remote monitoring and remote user control from a PC connected by wired Ethernet in a local network.

The investigated measures and methodologies to support “second life” of electronic modules (T4.3) and measures to support extension of life of electronics (T4.4) due to methodology of support used custom platform to adapt for the in-time-evolving design tools and embedded Linux PetaLinux operating system. We target developers designing the final commercial, AI inference based edge applications, mainly in the area of home automation.

Based on these requirements UTIA have selected two types of systems:

- Low cost systems. See [2], [3]
- Modul based systems. See [4,] [5] and [8], [9].

Both compared types of systems use STMicroelectronic STM32H573I-DK board for:

- Local system control on small graphical touch screen display.
- Remote system control from www browser based on www-server or secure communication based on mqtt client. Board is supported by STMicroelectronic CubeMX SW framework and also by NetXDuo SW framework on top of ThreadX OS and FileX SW package.

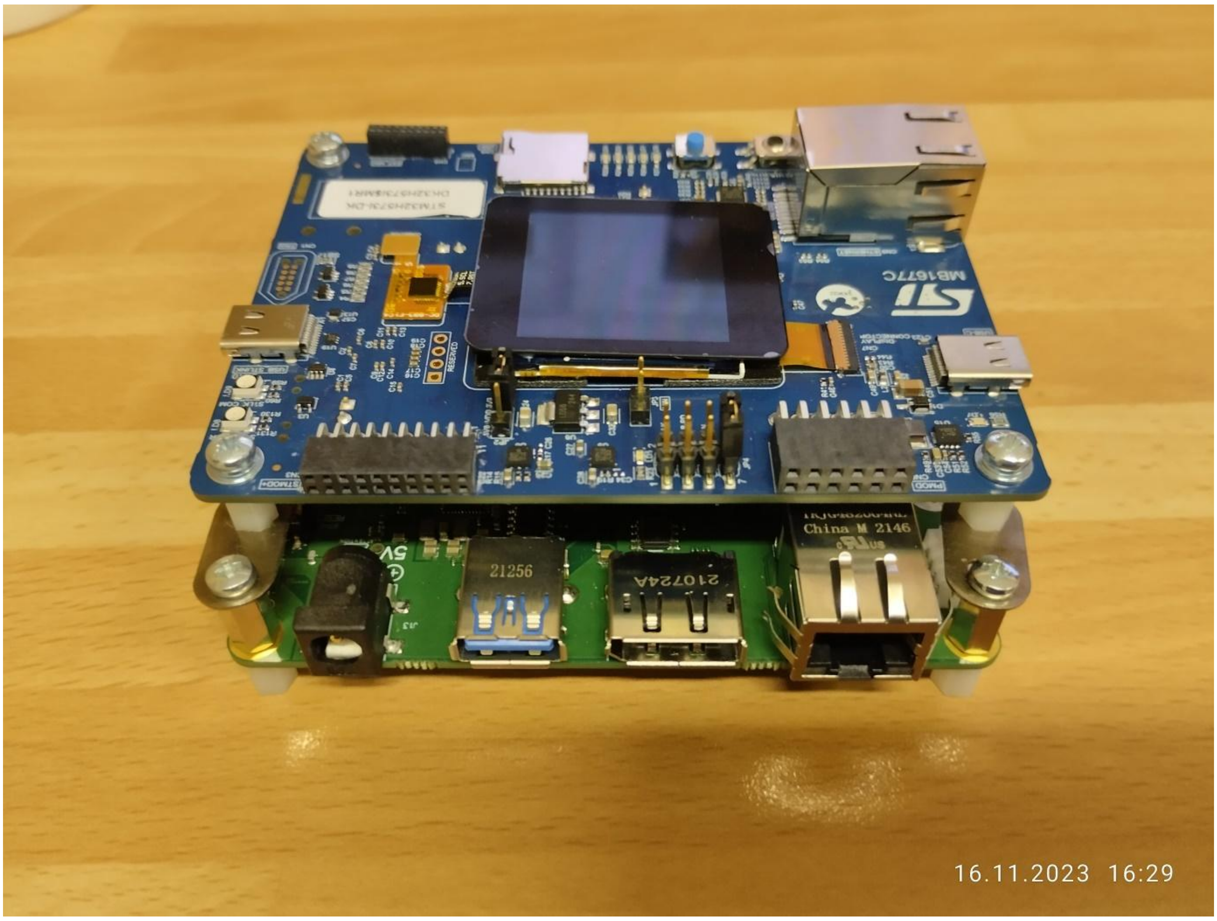
The MCU used on STM32H573I-DK board is a 40 nm chip with 32 bit ARM M33 MCU operating with 250 MHz clock, 2 MBytes of program flash memory and 640 KBytes of RAM.

Compared systems use 16 nm AMD ZynqUltrascale+ device with 64 bit ARM A53 Microprocessor and programmable logic in the same device and Petalinux OS.

- Low-cost systems have an AMD ZynqUltrascale+ device and DDR4 with all peripheral interfaces soldered on a single, low cost PCB
- Module-based systems have an AMD ZynqUltrascale+ device and DDR4 soldered on an 4x5 cm module connected by connectors to a carrier board with all peripheral interfaces

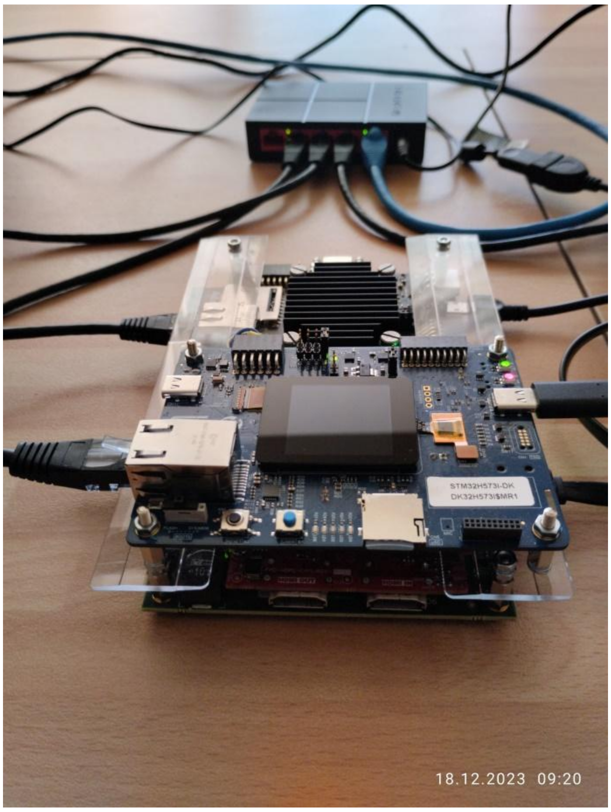
1.1 Low cost systems used by UTIA in EECONE T4.3 and T4.4

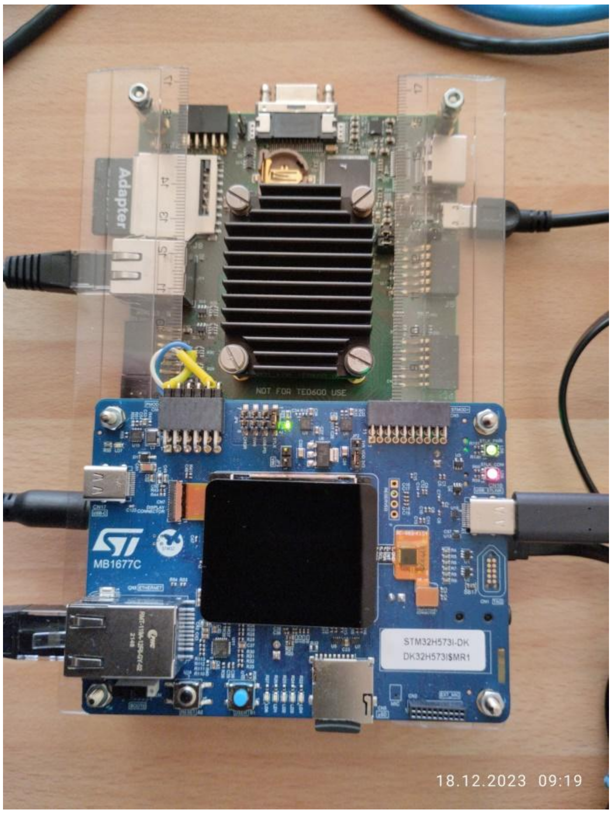
[1]	STM32H573I-DK	https://www.st.com/en/evaluation-tools/stm32h573i-dk.html	Local or remote system control (www-server or secure mqtt client) for [2], [3]
[2]	TE0802-02-1BEV2-A	https://shop.trenz-electronic.de/en/TE0802-02-1BEV2-A-MPSoC-Development-Board-with-AMD-Zynq-UltraScale-ZU1EG-and-1-GB-LPDDR4?c=474	AMD Vitis AI 3.0 AMD DPU in PL USB camera, remote X11 desktop
[3]	TE0802-02-2AEV2-A	MPSoC Development Board mit AMD Zynq™ UltraScale+™ ZU2 und 1 GB LPDDR4 | Trenz Electronic GmbH Online Shop (EN) (trenz-electronic.de)	AMD Vitis AI 3.0 AMD DPU in PL USB camera, remote X11 desktop



1.2 Module based systems used by UTIA in EECONE T4.3 and T4.4

[1] **[7]**	STM32H573I-DK TE0701-06 Carrier Board for Trenz Electronic 4 x 5 Modules TE0821 or TE0820	https://www.st.com/en/evaluation-tools/stm32h573i-dk.html https://shop.trenz-electronic.de/en/TE0701-06-Carrier-Board-for-Trenz-Electronic-4-x-5-Modules?c=261	Local or remote system control (www-server or secure mqtt client) for 2-1, 2-2 Carrier Board for range of 4x5 cm modules [3], [4].
[4] **[8]**	TE0821 Module: **17 module types** (to be supported)	https://shop.trenz-electronic.de/en/Products/Trenz-Electronic/TE08XX-Zynq-UltraScale/TE0821-Zynq-UltraScale/	AMD Vitis AI 3.0 AMD DPU in PL USB camera FULL HD HDMI display or remote X11 desktop
[5] **[9]**	TE0820 Module: **115 module types** **119 module types**	https://shop.trenz-electronic.de/en/Products/Trenz-Electronic/TE08XX-Zynq-UltraScale/TE0821-Zynq-UltraScale/	AMD Vitis AI 3.0 AMD DPU in PL USB camera FULL HD HDMI display or remote X11 desktop





This application note [10] and the accompanying evaluation package describe compilation of AI 3.0 models for different configurations of AMD DPUs for systems [8] and [9] for Vitis 2023.2.1 and Petalinux 2023.2. It is available for free public download from UTIA server dedicated to UTIA contributions to EECONE project:
https://zs.utia.cas.cz/index.php?ids=projects/eecone

This application note [10] and the accompanying evaluation package will be also available for free public download in format of wiki tutorial on Trenz-Electronic wiki server:
https://wiki.trenz-electronic.de/display/PD/Vitis+AI+and+Vitis+Acceleration+Tutorials+with+Trenz+Electronic+Modules

1.3 Objective of This Application Note and Evaluation Package

This application note [26] and the accompanying evaluation package describe compilation (in Vitis 2023.2.1 and Petalinux 2023.2. of AI 3.0 models for inference acceleration on DPUs of edge systems with Vitis 2023.2.1 and Petalinux 2023.2:

- Low-cost system [2] with zu1eg device: AMD DPU in configuration B512.
- Low-cost system [3] with zu2cg device: AMD DPU in configuration B512 or B1024.
- Module-based system [8]: AMD DPU in configurations: B512, B1024, or B1600.
- Module-based system [9]: AMD DPU in configurations: B512, B1024, B1600 and B4096.

The AMD DPU engine is unchanged in Vitis 2022.2 and in Vitis 2023.2.1 The DPU is present in the Vitis AI 3.0 repository. The general models are also unchanged and present in Vitis-AI 3.0 repository. The demo SW source code applications are compatible with Petalinux 2023.2 and AI 3.5 runtime libraries present in the Vitis AI 3.5 repository.

That is why we need to download and use both packages: Vitis AI 3.0 and Vitis AI 3.5.

2 Install packages

2.1 Install Vitis AI 3.0

Download the Vitis-AI 3.0 repository.

In browser, open page:

https://github.com/Xilinx/Vitis-AI/tree/3.0

Click on green Code button and download Vitis-AI-3.0.zip file.
Unzip to

```
~/work/Vitis-AI-3.0
```

The directory contains the Vitis-AI 3.0 framework.

Download the Vitis-AI 3.5 repository.

In browser, open page:

https://github.com/Xilinx/Vitis-AI/tree/master

Click on green Code button and download Vitis-AI-master.zip file.
Unzip to

```
~/work/Vitis-AI-3.5
```

The directory contains the Vitis-AI 3.5 framework.

Starting point for exploration of these Vitis AI 3.0 examples is this Xilinx www page.

https://xilinx.github.io/Vitis-AI/3.0/html/index.html

2.2 Vitis AI 3.0 Images and Videos

Download the AI 3.0 support archive archive with images:

https://www.xilinx.com/bin/public/openDownload?filename=vitis_ai_library_r3.0.0_images.tar.gz

Download the AI 3.0 support archive with videos:

https://www.xilinx.com/bin/public/openDownload?filename=vitis_ai_library_r3.0.0_video.tar.gz

Unzip and untar content to:

```
~/apps
~/samples
~/samples_onnx
```

These directories contain support material for AI 3.0 examples. Move content to:

```
~/work/Vitis-AI-3.0/examples/vai_library/apps

~/work/Vitis-AI-3.0/examples/vai_library/samples

~/work/Vitis-AI-3.0/examples/vai_library/samples_onnx
```

2.3 Install Docker

In Ubuntu 20.04, install docker

```
sudo apt install docker.io
```

Test the docker instalation:

```
sudo docker run hello-world
```

2.4 Install Docker Images for Vitis AI 3.5

Install precompiled docker images

```
sudo docker pull xilinx/vitis-ai-tensorflow-cpu:latest
```

```
sudo docker pull xilinx/vitis-ai-tensorflow2-cpu:latest
```

```
sudo docker pull xilinx/vitis-ai-pytorch-cpu:latest
```

Version 3.5 is `latest`. It is suitable for compilation of models for the installed Vitis AI 3.0 DPUs present in [22], [23], [24], [25] systems.

3 Vitis AI 3.0 Model Downloads

3.1 Python Downloader

Vitis AI 3.0 models can be downloaded with python tool `downloader.py`

```
cd ~/work/Vitis-AI-3.0/model_zoo
```

Example:
Select pytorch model framework
input: `pt`
Select model (pt_resnet50_imagenet_224_224_0.4_4.9G_3.0)
input num: `1`
chose model (all)
Input num: `0`

```
devel@ubuntu:~/work/Vitis-AI-3.0/model_zoo$ python3 downloader.py
Tip:
you need to input framework and model name, use space divide such as tf
vgg16
tf:tensorflow1.x  tf2:tensorflow2.x  cf:caffe  dk:darknet  pt:pytorch
all: list all model
input:pt
chose model
0 : all
1 : pt_resnet50_imagenet_224_224_0.4_4.9G_3.0
2 : pt_face-mask-detection_512_512_0.67G_3.0
3 : pt_inceptionv3_imagenet_299_299_0.5_5.7G_3.0
4 : pt_yolov5-large_coco_640_640_109.6G_3.0
5 : pt_fadnet_sceneflow_576_960_0.65_154G_3.0
6 : pt_psmnet_sceneflow_576_960_0.68_696G_3.0
7 : pt_squeezenet_imagenet_224_224_1.12G_3.0
8 : pt_OFA-resnet50_imagenet_160_160_0.88_1.8G_3.0
9 : pt_yolov6m_coco_640_640_82.4G_3.0
10 : pt_resnet50_imagenet_224_224_0.3_5.8G_3.0
11 : pt_OFA-depthwise-res50_imagenet_176_176_1.29G_3.0
12 : pt_salsanext_semantic-kitti_64_2048_0.6_20.4G_3.0
13 : pt_SOLO_coco_640_640_214G_3.0
14 : pt_resnet50_imagenet_224_224_0.5_4.1G_3.0
15 : pt_vehicle-type-classification_CarBodyStyle_224_224_3.64G_3.0
16 : pt_pmg_grocerystore_224_224_2.28G_3.0
17 : pt_SemanticFPN-mobilenetv2_cityscapes_512_1024_10G_3.0
18 : pt_bert-large_SQuADv1.1_384_246.42G_3.0
19 : pt_salsanextv2_semantic-kitti_64_2048_0.75_33.27G_3.0
20 : pt_inceptionv3_imagenet_299_299_0.6_4.5G_3.0
21 : pt_resnet50_imagenet_224_224_0.7_2.5G_3.0
22 : pt_vehicle-color-classification_VCoR_224_224_3.64G_3.0
```

```
23 : pt_inceptionv3_imagenet_299_299_11.4G_3.0
24 : pt_DRUNet_Kvasir_528_608_0.8G_3.0
25 : pt_vehicle-make-classification_VMMR_224_224_3.64G_3.0
26 : pt_OFA-yolo_coco_640_640_0.3_34.72G_3.0
27 : pt_MaskRCNN_coco_800_800_240G_3.0
28 : pt_OFA-resnet50_imagenet_224_224_15.0G_3.0
29 : pt_yolov5s6_coco_1280_1280_17G_3.0
30 : pt_OFA-rcan_DIV2K_360_640_40.5G_3.0
31 : pt_SESR-S_DIV2K_360_640_10.2G_3.0
32 : pt_resnet50_imagenet_224_224_0.6_3.3G_3.0
33 : pt_resnet50_imagenet_224_224_8.2G_3.0
34 : pt_CFLOW_LIDC_128_128_10.42G_3.0
35 : pt_yolox-nano_coco_416_416_1G_3.0
36 : pt_yolov4csp_coco_640_640_121G_3.0
37 : pt_OFA-yolo_coco_640_640_48.88G_3.0
38 : pt_OFA-resnet50_imagenet_192_192_0.74_3.6G_3.0
39 : pt_3D-UNET_kits19_128_128_128_1065.44G_3.0
40 : pt_bert-tiny_SQuADv1.1_384_453M_3.0
41 : pt_xilinxSR_360_640_DIV2K_364.88G_3.0
42 : pt_OFA-resnet50_imagenet_224_224_0.45_8.2G_3.0
43 : pt_inceptionv3_imagenet_299_299_0.3_8G_3.0
44 : pt_OFA-yolo_coco_640_640_0.5_24.62G_3.0
45 : pt_inceptionv3_imagenet_299_299_0.4_6.8G_3.0
46 : pt_movenet_coco_192_192_0.5G_3.0
47 : pt_fadnet_sceneflow_576_960_441G_3.0
48 : pt_ENet_cityscapes_512_1024_11.3G_3.0
49 : pt_fadnetv2_sceneflow_576_960_412G_3.0
50 : pt_SemanticFPN_cityscapes_256_512_10.56G_3.0
51 : pt_fadnetv2_sceneflow_576_960_0.51_201G_3.0
52 : pt_pointpillars_kitti_12000_100_11.2G_3.0
53 : pt_OFA-resnet50_imagenet_224_224_0.60_6.0G_3.0
54 : pt_yolov5-nano_coco_640_640_4.6G_3.0
55 : pt_CLOCs_kitti_3.0
56 : pt_HRNet_cityscapes_1024_2048_378G_3.0
57 : pt_bert-base_SQuADv1.1_384_70.66G_3.0
input num:1
chose model type
0: all
1 : GPU
2 : zcu102 & zcu104 & kv260
3 : vck190
4 : vck5000-DPUCVDX8H-4pe
5 : vck5000-DPUCVDX8H-6pe-aieDWC
6 : vck5000-DPUCVDX8H-6pe-aieMISC
7 : vck5000-DPUCVDX8H-8pe
input num:0
pt_resnet50_imagenet_224_224_0.4_4.9G_3.0.zip

                                              100.0%|100%
```

```
resnet50_pruned_0_4_pt-zcu102_zcu104_kv260-r3.0.0.tar.gz
                                              100.0%|100%
resnet50_pruned_0_4_pt-vck190-r3.0.0.tar.gz
                                              100.0%|100%
resnet50_pruned_0_4_pt-vck5000-DPUCVDX8H-4pe-r3.0.0.tar.gz
                                              100.0%|100%
resnet50_pruned_0_4_pt-vck5000-DPUCVDX8H-6pe-aieDWC-r3.0.0.tar.gz
                                              100.0%|100%
resnet50_pruned_0_4_pt-vck5000-DPUCVDX8H-6pe-aieMISC-r3.0.0.tar.gz
                                              100.0%|100%
resnet50_pruned_0_4_pt-vck5000-DPUCVDX8H-8pe-r3.0.0.tar.gz
                                              100.0%|100%
done
devel@ubuntu:~/work/Vitis-AI-3.0/model_zoo$
```

Delete model packages for other architectures:

```
./resnet50_pruned_0_4_pt-vck190-r3.0.0.tar.gz
./resnet50_pruned_0_4_pt-vck5000-DPUCVDX8H-4pe-r3.0.0.tar.gz
./resnet50_pruned_0_4_pt-vck5000-DPUCVDX8H-6pe-aieDWC-r3.0.0.tar.gz
./resnet50_pruned_0_4_pt-vck5000-DPUCVDX8H-6pe-aieMISC-r3.0.0.tar.gz
./resnet50_pruned_0_4_pt-vck5000-DPUCVDX8H-8pe-r3.0.0.tar.gz
```

Models for AMD DPU configuration B4096 are present in:

```
./resnet50_pruned_0_4_pt-zcu102_zcu104_kv260-r3.0.0.tar.gz
```

Models for other AMD DPU configurations have to be compiled from data present in `zip` archive:

```
./pt_resnet50_imagenet_224_224_0.4_4.9G_3.0.zip
```

unzip this archive to directory

```
./pt_resnet50_imagenet_224_224_0.4_4.9G_3.0
```

3.2 Arch.json File

An `arch.json` is produced by Vivado in the process of compilation of the AMD DPU.

In case of TE0821 system with ID=3, [4] the `arch.json` file describing the AMD DPU B1600 configuration has been created in:

```
~/work/te0821_3_240/test_board_dpu_trd/dpu_trd_system_hw_link/Hardware/
dpu.build/link/vivado/vpl/prj/prj.gen/sources_1/bd/zusys/ip/zusys_DPUCZ
DX8G_1_0/arch.json
```

Copy the `arch.json` file for the selected AMD DPU configuration to

```
~/work/Vitis-AI-3.0/model_zoo/arch.json
```

4 Compile Pytorch Models

Copy fie `compile_model.sh` from the repository associated to this application note to the directory

```
~/work/Vitis-AI-3.0/model_zoo/compile_model.sh
```

Change mode to executable for file `compile_model.sh`

```
Chmod +x ~/work/Vitis-AI- 3.0/model_zoo/compile_model.sh
```

4.1 Compile First Pytorch Model for AMD DPU in B1600 Configuration

Change directory to

```
~/work/Vitis-AI-3.0
```

Start `pytorch` docker

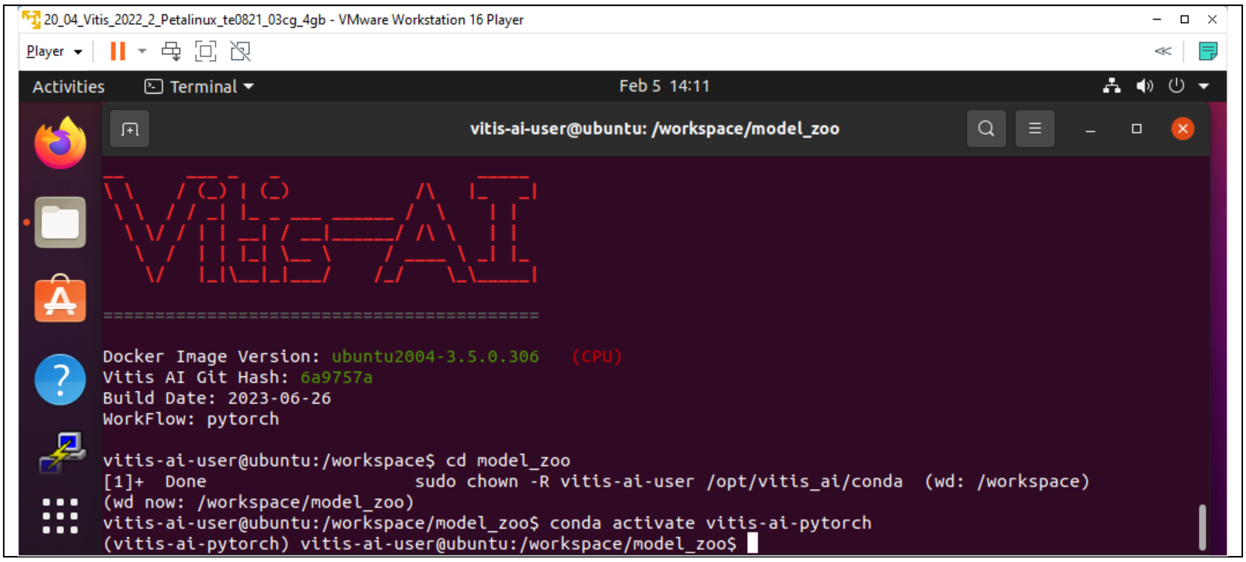
```
sudo ./docker_run.sh xilinx/vitis-ai-pytorch-cpu:latest
```

In `pytorch` docker WorkFlow, change directory to `model_zoo`

```
cd model_zoo
```

Start `pytorch` compiler ftramework

```
conda activate vitis-ai-pytorch
```


20_04_Vitis_2022_2_Petalinux_te0821_03cg_4gb - VMware Workstation 16 Player
Player
Activities
Terminal
Feb 5 14:11
vitis-ai-user@ubuntu: /workspace/model_zoo

```
==========================================

Docker Image Version: ubuntu2004-3.5.0.306   (CPU)
Vitis AI Git Hash: 6a9757a
Build Date: 2023-06-26
WorkFlow: pytorch

vitis-ai-user@ubuntu:/workspace$ cd model_zoo
[1]+  Done                    sudo chown -R vitis-ai-user /opt/vitis_ai/conda  (wd: /workspace)
(wd now: /workspace/model_zoo)
vitis-ai-user@ubuntu:/workspace/model_zoo$ conda activate vitis-ai-pytorch
(vitis-ai-pytorch) vitis-ai-user@ubuntu:/workspace/model_zoo$
```

Parameters for model compilation script `compile_model.sh` :

1.	`pt`	`pytorch model`
2.	`resnet50_pruned_0_4_pt`	`Model name`
3.	`pt_resnet50_imagenet_224_224_0.4_4.9G_3.0`	`Directory`
4.	`ResNet_0_int.xmodel`	`Quantized model`

`Pytorch model` is indicated by `pt`
`Model name` can be found in the readme file in `classification` directory:

```
/home/devel/work/Vitis-AI-3.0/examples/vai_library/samples/classification/readme
```

Part of `classification/readme` file content

```
Valid model name:
    …
    resnet50_pruned_0_4_pt
    …
```

`Directory` is name of the unzipped directory associated to the model
`Quantized model` is name of file `<name>.xmodel` located in directory associated to the model insubdirectory `quantized`

```
/work/Vitis-AI-3.0/model_zoo/pt_resnet50_imagenet_224_224_0.4_4.9G_3.0/quantized/ResNet_0_int.xmodel
```

This file name `<name>` can be different for different models but the location alvais in the subdirectory `quantized` and the extension is alvais `.xmodel`

Command for compilation of pytorch model:

```
./compile_model.sh pt resnet50_pruned_0_4_pt
pt_resnet50_imagenet_224_224_0.4_4.9G_3.0 ResNet_0_int.xmodel
```

Compilation listing:

```
(vitis-ai-pytorch) vitis-ai-
user@ubuntu:/workspace/model_zoo$ ./compile_model.sh pt
resnet50_pruned_0_4_pt pt_resnet50_imagenet_224_224_0.4_4.9G_3.0
ResNet_0_int.xmodel
**************************************************
* VITIS_AI Compilation - Xilinx Inc.
**************************************************
[UNILOG][INFO] Compile mode: dpu
[UNILOG][INFO] Debug mode: null
[UNILOG][INFO] Target architecture: DPUCZDX8G_ISA1_B1600
[UNILOG][INFO] Graph name: ResNet_0, with op num: 417
[UNILOG][INFO] Begin to compile...
[UNILOG][INFO] Total device subgraph number 3, DPU subgraph number 1
[UNILOG][INFO] Compile done.
[UNILOG][INFO] The meta json is saved to
"/workspace/model_zoo/./compiled_output_pt/resnet50_pruned_0_4_pt/meta.
json"
[UNILOG][INFO] The compiled xmodel is saved to
"/workspace/model_zoo/./compiled_output_pt/resnet50_pruned_0_4_pt/resne
t50_pruned_0_4_pt.xmodel"
[UNILOG][INFO] The compiled xmodel's md5sum is
8977ac80ac31133b9a957c20d55643d4, and has been saved to
"/workspace/model_zoo/./compiled_output_pt/resnet50_pruned_0_4_pt/md5su
m.txt"
(vitis-ai-pytorch) vitis-ai-user@ubuntu:/workspace/model_zoo$
```

Directory `compiled_output_pt` is created. It contains compiled model files for the AMD DPU in configuration B1600 in the directory `resnet50_pruned_0_4_pt`

```
./compiled_output_pt/resnet50_pruned_0_4_pt
```

One file is needed in the compiled model. File is present in archive with precompiled model for the AMD DPU in B4096 configurations.

Open archive

```
resnet50_pruned_0_4_pt-zcu102_zcu104_kv260-r3.0.0.tar.gz
```

copy file

```
resnet50_pruned_0_4_pt/resnet50_pruned_0_4_pt.prototxt
```

to file

```
./compiled_output_pt/resnet50_pruned_0_4_pt/resnet50_pruned_0_4_pt.prot
otxt
```

General classification Model: `pt_resnet50_imagenet_224_224_0.4_4.9G_3.0` is compiled for AMD DPU in B1600 configuration. It contains 4 files:

```
~/work/Vitis-AI-3.0/model_zoo/compiled_output_pt/resnet50_pruned_0_4_pt/md5sum.txt

~/work/Vitis-AI-3.0/model_zoo/compiled_output_pt/resnet50_pruned_0_4_pt/meta.json

~/work/Vitis-AI-3.0/model_zoo/compiled_output_pt/resnet50_pruned_0_4_pt/resnet50_pruned_0_4_pt.xmodel

~/work/Vitis-AI-3.0/model_zoo/compiled_output_pt/resnet50_pruned_0_4_pt/resnet50_pruned_0_4_pt.prototxt
```

4.2 Compile Other Pytorch Models for AMD DPU in B1600 Configuration

Perform similar compilation steps for other 9 seleced models:

```
Face detection Model:
pt_face-mask-detection_512_512_0.67G_3.0


Vehicle make Model:
pt_vehicle-make-classification_VMMR_224_224_3.64G_3.0
Vehicle type Model:
pt_vehicle-type-classification_CarBodyStyle_224_224_3.64G_3.0


Vehicle color Model:
pt_vehicle-color-classification_VCoR_224_224_3.64G_3.0
General classification Model:
pt_resnet50_imagenet_224_224_8.2G_3.0
General classification Model:
pt_resnet50_imagenet_224_224_0.3_5.8G_3.0
General classification Model:
pt_resnet50_imagenet_224_224_0.4_4.9G_3.0 (allready compiled)
General classification Model:
pt_resnet50_imagenet_224_224_0.5_4.1G_3.0
General classification Model:
pt_resnet50_imagenet_224_224_0.6_3.3G_3.0
General classification Model:
pt_resnet50_imagenet_224_224_0.7_2.5G_3.0
```

Find the expected model names in sample application `readme` files:

```
~/work/Vitis-AI-3.0/examples/vai_library/samples/yolov4/readme

~/work/Vitis-AI-3.0/examples/vai_library/samples/vehicleclassification/readme
```

```
~/work/Vitis-AI-3.0/examples/vai_library/samples/classification/readme
```

Result of compilations is folder (with all 10 model subfolders targeting the AMD DPU in B1600 configuration)

```
./compiled_output_pt/*
```

Compres it as .zip file. It will be used to target Vitis AI 3.0 demos on system [4] with AMD DPU configuration B1600

Copy this archive into directory indicating the AMD DPU configuration like:

```
~/work/B1600/compiled_output_pt.zip
```

Delete directory

```
~/work/Vitis-AI-3.0/model_zoo/compiled_output_pt/*
```

4.3 Compile Pytorch Models for other AMD DPU Configurations

To target other systems [2], [3] and [25], the arch.json describing the AMD DPU configurat ion has to be replaced.

Copy the `arch.json` file for the selected AMD DPU configuration to

```
~/work/Vitis-AI-3.0/model_zoo/arch.json
```

Repeat:

- Compilation for system [2], AMD DPU in B0512 configuration will be created.
- Compilation for system [3], AMD DPU in B1024 configuration will be created.

Allready available:

- Compilation for system [4], AMD DPU in B1600 configuration will be created.
- In case of system [25], the AMD DPU in B4096 configuration all needed files are present in the download packages archives for `zcu102_zcu104_kv260` targets.

To target tested systems [22], [23], [24] and [25] use these created `.zip` archives:

```
~/work/B0512/compiled_output_pt.zip
~/work/B1024/compiled_output_pt.zip
~/work/B1600/compiled_output_pt.zip
~/work/B4096/compiled_output_pt.zip
```

4.4 Sample Vitis AI 3.5 applications

Demos from the Vitis AI 3.5 library can be compiled in Vitis 2023.2 for Petalinux 2023.2 run-time on the test boards. Demos can use models from the Vitis AI 3.0 library for the DPU HW with different sizes of the DPUs defined in Vitis AI 3.0. This is possible due to unchanged internal structure of these DPUs.

SW examples from the Vitis AI 3.5 package to be moved to the target systems [22], [23], [24] and [25] are:

```
~/work/Vitis-AI-3.5/examples/vai_library/samples/yolov4
~/work/Vitis-AI-3.5/examples/vai_library/samples/vehicleclassification
~/work/Vitis-AI-3.5/examples/vai_library/samples/classification
```

Compress them to:

```
~/work/yolov4.zip
~/work/vehicleclassification.zip
~/work/classification.zip
```

Archives to be moved to the target systems [22], [23], [24] and [25] are:

```
~/work/B0512/compiled_output_pt.zip
~/work/B1024/compiled_output_pt.zip
~/work/B1600/compiled_output_pt.zip
~/work/B4096/compiled_output_pt.zip
~/work/yolov4.zip
~/work/vehicleclassification.zip
~/work/classification.zip
```

4.5 Compile Tensorflow Models

Process of compilation models is similar to pytorch model. Change directory to:

```
~/work/Vitis-AI-3.0
```

Start `tensorflow` docker

```
sudo ./docker_run.sh xilinx/vitis-ai-tensorflow-cpu:latest
```

In `tensorflow` docker WorkFlow, change directory to `model_zoo`

```
cd model_zoo
```

Start `tensorflow` compiler ftramework

```
conda activate vitis-ai-tensorflow
```

There are only 3 parameters for model compilation script `compile model.sh` :

1. `tf tensorflow model`
2. `Model name`
3. `Directory`

The `tensorflow model` is indicated by `tf`
`Model name` can be found in the `readme` file in the sample application.
`Directory` is name of the unzipped directory associated to the model.

4.6 Compile Tensorflow2 Models

Process of compilation models is similar to pytorch model. Change directory to:

```
~/work/Vitis-AI-3.0
```

Start `tensorflow` docker

```
sudo ./docker_run.sh xilinx/vitis-ai-tensorflow2-cpu:latest
```

In `tensorflow` docker WorkFlow, change directory to `model_zoo`

```
cd model_zoo
```

Start `tensorflow2` compiler ftramework

```
conda activate vitis-ai-tensorflow2
```

There are only 3 parameters for model compilation script `compile model.sh` :

1. `tf2 tensorflow2 model`
2. `Model name`
3. `Directory`

`Tensorflow2 model` is indicated by `tf2`
`Model name` can be found in the `readme` file in the sample application.
`Directory` is name of the unzipped directory associated to the model.

5 Test Sample Vitis AI 3.5 Applications

5.1 Install Archives on Systems [22]-[25] with Petalinux 2023.2

Use `ssh` to copy Vitis AI 3.5 examples from PC :

```
~/work/yolov4.zip
~/work/vehicleclassification.zip
~/work/classification.zip
```

to edge evaluation board

```
~/yolov4.zip
~/vehicleclassification.zip
~/classification.zip
```

Use `ssh` to copy compiled Vitis 3.0 model archives from PC:

```
~/work/B0512/compiled_output_pt.zip
~/work/B1024/compiled_output_pt.zip
~/work/B1600/compiled_output_pt.zip
~/work/B4096/compiled_output_pt.zip
```

To edge evaluation board

```
~/B0512/compiled_output_pt.zip
~/B1024/compiled_output_pt.zip
~/B1600/compiled_output_pt.zip
~/B4096/compiled_output_pt.zip
```

We target one of evaluation boards [22], [23], [24] or [25] with Petalinux 2023.2.

On target board, unzip Vitis AI 3.5 sample applications from

```
~/yolov4.zip
~/vehicleclassification.zip
~/classification.zip
```

to:

```
~/yolov4
~/vehicleclassification
~/classification
```

On target board, copy models from the relevant directory:
[22]: `<config> = B0512`
[23]: `<config> = B1024`
[24]: `<config> = B1600`
[25]: `<config> = B4096`

From:

```
~/<config>/compiled_output_pt/face_mask_detection_pt
```

to:

```
~/yolov4/face_mask_detection_pt
```

From:

```
~/<config>/compiled_output_pt/vehicle_make_resnet18_pt
~/<config>/compiled_output_pt/vehicle_type_resnet18_pt
```

to:

```
~/vehicleclassification/vehicle_make_resnet18_pt
~/vehicleclassification/vehicle_type_resnet18_pt
```

From:

```
~/<config>/compiled_output_pt/chen_color_resnet18_pt
~/<config>/compiled_output_pt/resnet50_pt
~/<config>/compiled_output_pt/resnet50_pruned_0_3_pt
~/<config>/compiled_output_pt/resnet50_pruned_0_4_pt
~/<config>/compiled_output_pt/resnet50_pruned_0_5_pt
~/<config>/compiled_output_pt/resnet50_pruned_0_6_pt
~/<config>/compiled_output_pt/resnet50_pruned_0_7_pt
```

to:

```
~/classification/chen_color_resnet18_pt
~/classification/resnet50_pt
~/classification/resnet50_pruned_0_3_pt
~/classification/resnet50_pruned_0_4_pt
~/classification/resnet50_pruned_0_5_pt
~/classification/resnet50_pruned_0_6_pt
~/classification/resnet50_pruned_0_7_pt
```

5.2 Compile Vitis AI 3.5 Applications on Systems [22]-[25]

On target board, compile all sample applications:

```
cd ~/yolov4
chmod +x build.sh
./build,sh

cd ~/vehicleclassification
chmod +x build.sh
./build,sh

cd ~/classification
chmod +x build.sh
./build,sh
```

5.3 Export Path to the DPU

On low-cost systems TE0802 [2] or TE0802 [3], use export command:

```
export XLNX_VART_FIRMWARE=/run/media/mmcblk0p1/dpu.xclbin
```

On module-based systems supporting modules TE0821 [24] or modules TE0820 [25], use export command:

```
export XLNX_VART_FIRMWARE=/run/media/mmcblk1p1/dpu.xclbin
```

5.4 Test Vitis AI 3.5 Application on Systems [22]-[25]

Test Vitis AI 3.5 applications with commands described in sample application `readme` files:

```
~/work/Vitis-AI-3.5/examples/vai_library/samples/yolov4/readme

~/work/Vitis-AI-3.5/
examples/vai_library/samples/vehicleclassification/readme

~/work/Vitis-AI-3.0/examples/vai_library/samples/classification/readme
```

Change directory to

```
~/yolov4
```

Test face mask detection with input from file `sample_face_mask.jpg`

```
./test_jpeg_yolov4 face_mask_detection_pt sample_face_mask.jpg
```

Output: face mask detection coordinates on terminal.

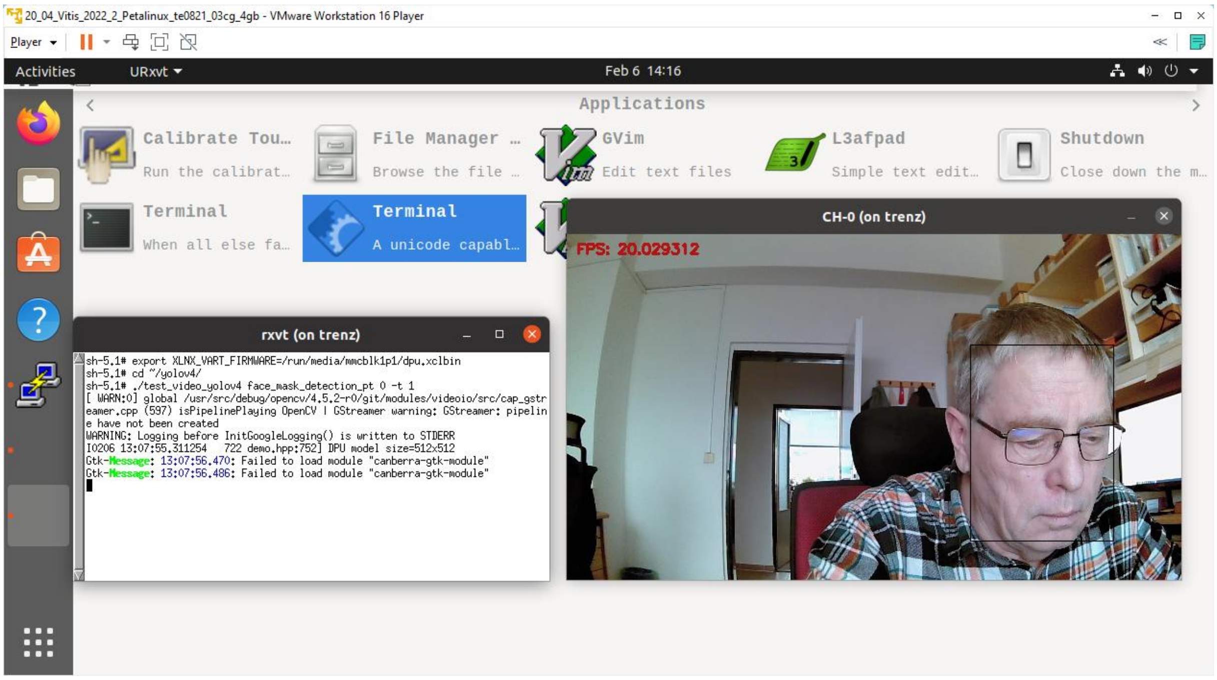
Test performance of face mask detection with input from files listed in `test_performance_face_mask.list` file.

```
./test_performance_yolov4 face_mask_detection_pt
test_performance_face_mask.list -s 60 -t 3
```

Output: data about performance (in FPS) of face mask detection on terminal.
-s 60   defines length of benchmark in seconds.
-t 3   defines use of 3 SW threads (with one AMD DPU)

Test face mask detection with USB www camera input

```
./test_video_yolov4 face_mask_detection_pt 0 -t 1
```



```
sh-5.1# export XLNX_VART_FIRMWARE=/run/media/mmcblk1p1/dpu.xclbin
sh-5.1# cd ~/yolov4/
sh-5.1# ./test_video_yolov4 face_mask_detection_pt 0 -t 1
[ WARN:0] global /usr/src/debug/opencv/4.5.2-r0/git/modules/videoio/src/cap_gstr
eamer.cpp (597) isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipelin
e have not been created
WARNING: Logging before InitGoogleLogging() is written to STDERR
I0206 13:07:55.311254   722 demo.hpp:752] DPU model size=512x512
Gtk-Message: 13:07:56.470: Failed to load module "canberra-gtk-module"
Gtk-Message: 13:07:56.486: Failed to load module "canberra-gtk-module"
```

Face mask detection Vitis AI 3.0 application with camera input, DPU: B1600, system [4].

Output: Video output with face mask coordinates in video and performance (in FPS) displayed as text in video.

`0`   defines USB www camera device number
`-t 1` defines use of 1 SW thread

Face mask detection Vitis AI 3.0 application can be stopped by pointing mouse to the X11 window with video output and typing `Esc` key on the keyboard.

5.5 Other Vitis AI 3.0 Applications on Systems [22]-[25]

Vitis AI 3.0 sample test applications located in the directory `~/vehicleclassification` have analogical format for calling and parameters.

Two different AI models can be used to classify `make` or `type` of the vehicle.

Vitis AI 3.0 sample test applications located in the directory `~/classification` have analogic format.

Several different AI models can be used to

- Classify `color` of vehicle,
- Classify objects with original model
- Classify objects with model, pruned by factor of 0.3, 0.4, 0.5, 0.6 or 0.7, have gradually reduced complexity and potencially reduced classification quality.

5.6 Measured Vitis AI 3.0 Performance on Systems [22]-[25]

Mesured performance indicators are summarized in sections 5.8, 5.9 and 5.10, 5.11 for low-cost systems [22], [23] and for module-based systems [24], [25]:

- Performance in frames per second [FPS],
- Power consumption in [W],
- End to end performance in (int8) Giga operations per second [GOPs].

5.7 TE0802-02-1BEV2-A board [2], ID=4, 1EG Device, DPU: B512

Vitis AI 3.0 exampes	Performance with input from camera e2e [FPS]	Power with camera and VGA [W]	Performance with input from file e2e [FPS]	Power with input from file [W]	GigaOps with input from file e2e [Gops]
Face detection Model: pt_face-mask-detection_512_512_0.67G_3.0	17.3	6.9	30.9	6.2	20.7
Vehicle make Model: pt_vehicle-make-classification_VMMR_224_224_3.64G_3.0	18.9	7.4	22.3	6.7	81.1
Vehicle type Model: pt_vehicle-type-classification_CarBodyStyle_224_224_3.64G_3.0	19.1	7.4	22.3	6.7	81.1
Vehicle color Model: pt_vehicle-color-classification_VCoR_224_224_3.64G_3.0	18.9	7.4	22.3	6.7	81.1
General classification Model: pt_resnet50_imagenet_224_224_8.2G_3.0	8.1	7.2	9.3	6.5	76.2
General classification Model: pt_resnet50_imagenet_224_224_0.3_5.8G_3.0	9.2	7.1	11.1	6.5	64.3
General classification Model: pt_resnet50_imagenet_224_224_0.4_4.9G_3.0	9.8	7.1	12.2	6.5	59.7
General classification Model: pt_resnet50_imagenet_224_224_0.5_4.1G_3.0	10.2	7.0	12.9	6.4	52.8
General classification Model: pt_resnet50_imagenet_224_224_0.6_3.3G_3.0	11.4	7.0	14.9	6.4	49,1
General classification Model: pt_resnet50_imagenet_224_224_0.7_2.5G_3.0	11.8	6.9	16.2	6.3	40.5

Measurement conditions:

- TE0802-02-1BEV2-A board with ZU01EG device 1 GB DDR4
- DPU in B512 configuration
- USB WWW camera ETERNICO with sensor JX_F23, 1920x1080, max 20 FPS
- Keyboard RPi
- Mouse RPi
- VGA display (1280x720p60)
- Power supply 5V/4A
- Power measured at the 230V power plug

5.8 TE0802-02-2AEV2-A board [3], ID=1, 2CG Device, DPU: B1024

Vitis AI 3.0 exampes	Performance with input from camera e2e [FPS]	Power with camera and VGA [W]	Performance with input from file e2e [FPS]	Power with input from file [W]	GigaOps with input from file e2e [Gops]
Yolov4 face mask detection Model: pt_face-mask-detection_512_512_0.67G_3.0	17.3	7.2	43.9	6.9	29.4
Vehicleclassification vehicle make Model: pt_vehicle-make-classification_VMMR_224_224_3.64G_3.0	18.1	7.7	39.8.	7.6	144.8
Vehicleclassification vehicle type Model: pt_vehicle-type-classification_CarBodyStyle_224_224_3.64G_3.0	18.1	7.7	39.8	7.6	144.8
Classification vehicle color Model: pt_vehicle-color-classification_VCoR_224_224_3.64G_3.0	18.0	7.8	39.8	7.7	144.8
Classification Model: pt_resnet50_imagenet_224_224_8.2G_3.0	10.0	7.8	15.0	6.7	123.0
Classification Model: pt_resnet50_imagenet_224_224_0.3_5.8G_3.0	10.5	7.8	16.9	6.6	98.0
Classification Model: pt_resnet50_imagenet_224_224_0.4_4.9G_3.0	10.7	7.7	18.1	6.5	88.7
Classification Model: pt_resnet50_imagenet_224_224_0.5_4.1G_3.0	11.5	7.7	19.3	6.4	79.1
Classification Model: pt_resnet50_imagenet_224_224_0.6_3.3G_3.0	12.0	6.7	20.6	6.2	68.0
Classification Model: pt_resnet50_imagenet_224_224_0.7_2.5G_3.0	12.5	6.5	23.4	6.2	58.5

Measurement conditions:

- TE0802-02-2AEV2-A board with ZU02CG device 1 GB DDR4
- DPU in B1024 configuration
- USB WWW camera ETERNICO ET201 Full HD, sensor JX_F23, 1920x1080, 20 FPS
- Keyboard RPi
- Mouse RPi
- VGA display (1280x720p60)
- Power supply 5V/4A
- Power measured at the 230V power plug

5.9 TE0821-01-3AE31KA module, ID=3, TE0701-06 [8], DPU: B1600

Vitis AI 3.0 exampes	Performance input from camera e2e (-t 1) [FPS]	Power with camera e2e (-t 1) [W]	Performance input from file e2e (-t 3) [FPS]	Power with input from file e2e (-t 3) [W]	GigaOps input from file e2e (-t 3) [Gops]
Yolov4 face mask detection Model: pt_face-mask-detection_512_512_0.67G_3.0	20.0	7.7	76.1	7.6	50.9
Vehicleclassification vehicle make Model: pt_vehicle-make-classification_VMMR_224_224_3.64G_3.0	20.0	7.9	61.2	8.4	222.7
Vehicleclassification vehicle type Model: pt_vehicle-type-classification_CarBodyStyle_224_224_3.64G_3.0	20.0	7.9	61.2	8.4	222.7
Classification vehicle color Model: pt_vehicle-color-classification_VCoR_224_224_3.64G_3.0	20.0	7.9	61.2	8.4	222.7
Classification Model: pt_resnet50_imagenet_224_224_8.2G_3.0	20.0	8.8	27.5	8.6	225.5
Classification Model: pt_resnet50_imagenet_224_224_0.3_5.8G_3.0	20.0	8.5	36.0	8.6	208.8
Classification Model: pt_resnet50_imagenet_224_224_0.4_4.9G_3.0	20.0	8.4	39.7	8.5	194.5
Classification Model: pt_resnet50_imagenet_224_224_0.5_4.1G_3.0	20.0	8.3	44.1	8.5	180.8
Classification Model: pt_resnet50_imagenet_224_224_0.6_3.3G_3.0	20.0	8.2	45.4	8.4	149.8
Classification Model: pt_resnet50_imagenet_224_224_0.7_2.5G_3.0	20.0	8.0	58.3	8.4	145.7

Measurement conditions:

- TE0821-01-3AE31KA module 3cg-1e, 4GB DDR4, TE0701-06 carrier board
- DPU in B1024 configuration
- USB WWW camera ETERNICO ET201 Full HD, sensor JX_F23, 1920x1080, 20 FPS
- Keyboard RPi
- Mouse RPi
- Remote X11 desktop
- Power supply 12V/5A
- Power measured at the 230V power plug

5.10 TE0820-03-04EV-1E-2GB module, ID=84, TE0701-06 [9], DPU: B4096

Vitis AI 3.0 exampes	Performance with input from camera e2e [FPS]	Power with camera and VGA [W]	Performance input from file e2e (-t 3) [FPS]	Power with input from file e2e (-t 3) [W]	GigaOps input from file e2e (-t 3) [Gops]
Face detection Model: pt_face-mask-detection_512_512_0.67G_3.0	30.0	10.0	113	9.8	75.7
Vehicle make Model: pt_vehicle-make-classification_VMMR_224_224_3.64G_3.0	30.0	10.5	167	13.6	607.9
Vehicle type Model: pt_vehicle-type-classification_CarBodyStyle_224_224_3.64G_3.0	30.0	10.5	167	13.6	607.9
Vehicle color Model: pt_vehicle-color-classification_VCoR_224_224_3.64G_3.0	30.0	10.5	167	13.6	607.9
General classification Model: pt_resnet50_imagenet_224_224_8.2G_3.0	30.0	11.9	59.6	13.0	488.7
General classification Model: pt_resnet50_imagenet_224_224_0.3_5.8G_3.0	30.0	11.5	69.2	12.7	401.3
General classification Model: pt_resnet50_imagenet_224_224_0.4_4.9G_3.0	30.0	11.2	73.8	12.4	361.6
General classification Model: pt_resnet50_imagenet_224_224_0.5_4.1G_3.0	30.0	11.0	81.1	12.2	332.5
General classification Model: pt_resnet50_imagenet_224_224_0.6_3.3G_3.0	30.0	10.7	91.1	11.9	300.6
General classification Model: pt_resnet50_imagenet_224_224_0.7_2.5G_3.0	30.0	10.6	99.6	11.5	249.0

Measurement conditions:

- TE0820-03-04EV-1EA 2GB module with 12V FAN on TE0701-06 carrier board
- DPU in B4096 configuration
- USB WWW colour camera logi 720p, Logitech, 1280x720p30, 30 FPS
- Keyboard RPi
- Mouse RPi
- Remote X11 desktop
- Power supply 12V/5A
- Power measured at the 230V power plug

6 References

[1] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for STM32H573I-DK web server. (Application note, with evaluation package, UTIA). Published for public access from: https://zs.utia.cas.cz/index.php?ids=results&id=1_STM32H573_DK
This application and evaluation package will be based on the STM32CubeH5 Firmware Examples for STM32H5xx Series Application based on NetXDuo: **Nx_WebServer.**
This STM application provides an example of Azure RTOS NetX Duo stack usage on STM32H573G-DK board, it shows how to develop Web HTTP server based application.
https://htmlpreview.github.io/?https://raw.githubusercontent.com/STMicroelectronics/STM32CubeH5/master/Projects/STM32CubeProjectsList.html

[2] Lukáš Kohout, Jiří Kadlec, Zdeněk Pohl: Support for TE0802-02-1BEV2-A board with Vitis AI 3.0 DPU and VGA display (Application note with evaluation package, UTIA). Published for public free access from:
https://zs.utia.cas.cz/index.php?ids=results&id=2_TE0802-02-1BEV2-A_AI_3_0_VGA

[3] Lukáš Kohout, Jiří Kadlec, Zdeněk Pohl: Support for TE0802-02-2AEV2-A board with Vitis AI 3.0 DPU and VGA display (Application note, with evaluation package, UTIA). Published for public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=3_TE0802-02-2AEV2-A_AI_3_0_VGA

[4] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for module-based systems with TE0821 modules on TE0701 carrier board with Vitis AI 3.0 DPU (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=4_TE0821_AI_3_0

[5] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for module-based systems with TE0820 modules on TE0701 carrier board with Vitis AI 3.0 DPU (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=5_TE0820_AI_3_0

[6] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, Raissa Likhonina: Description of compilation of Vitis AI 3.0 models for different configurations of AMD DPUs, (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=6_TE_AI_3_0

[7] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for STM32H573I-DK V1.4.0 web server. (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=21_STM32H753_DK_V1_4_0
This application and evaluation package is based on the STM32CubeH5 ver 1.4 Firmware Examples for STM32H5xx Series Application based on NetXDuo: Nx_WebServer. https://www.st.com/en/development-tools/stm32cubeide.html

[8] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout: Support for TE0821 Modules in Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8G (Application note, with evaluation package, UTIA). Published for free public access from:
https://zs.utia.cas.cz/index.php?ids=results&id=24_TE0821_AI_3_5

[9] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, Raissa Likhonina: Support for TE0820 Modules in Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8V (Application note, with evaluation

package, UTIA). Published for free public access from: https://zs.utia.cas.cz/index.php?ids=results&id=25_TE0820_AI_3_5

[10] Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout, Raissa Likhonina: Compilation of AI 3.0 models for Vitis 2023.2, AI 3.5 SW, AI 3.0 DPUCZDX8V. (Application note, with evaluation package, UTIA). Published for free public access from: https://zs.utia.cas.cz/index.php?ids=results&id=26_TE_AI_3_5