AMD Tech Day

AI#2 - From AMD Vitis AI to Model Deployment on KV260: A Streamlined Workflow

Dimitrios Kolosov, Senior Field Application Engineer - FPGA Specialist

AVNET SILICA

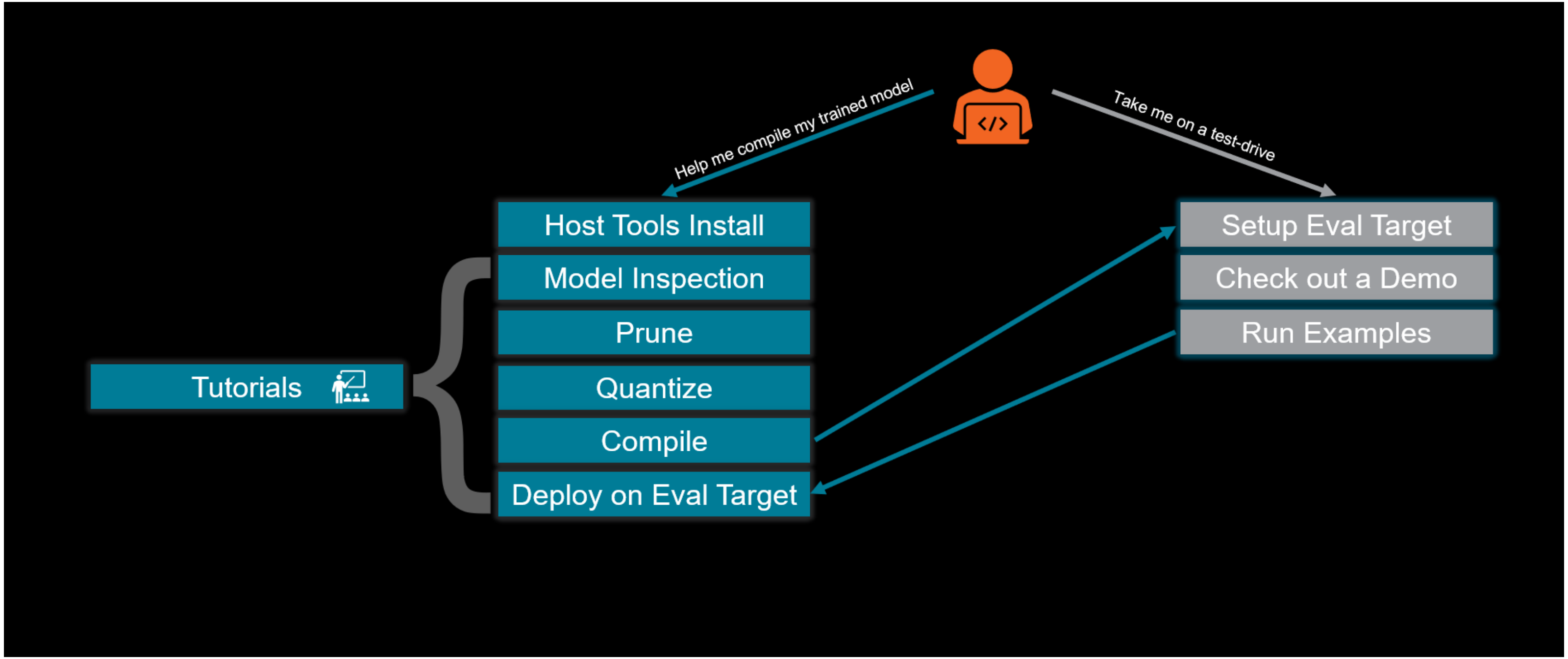
8th November, 2023

Agenda

- Getting Started with Vitis AI and Model Zoo
- Downloading a model from Vitis AI (ResNet50 Example)
- Edge Deployment on KRIA KV260
- Q&A

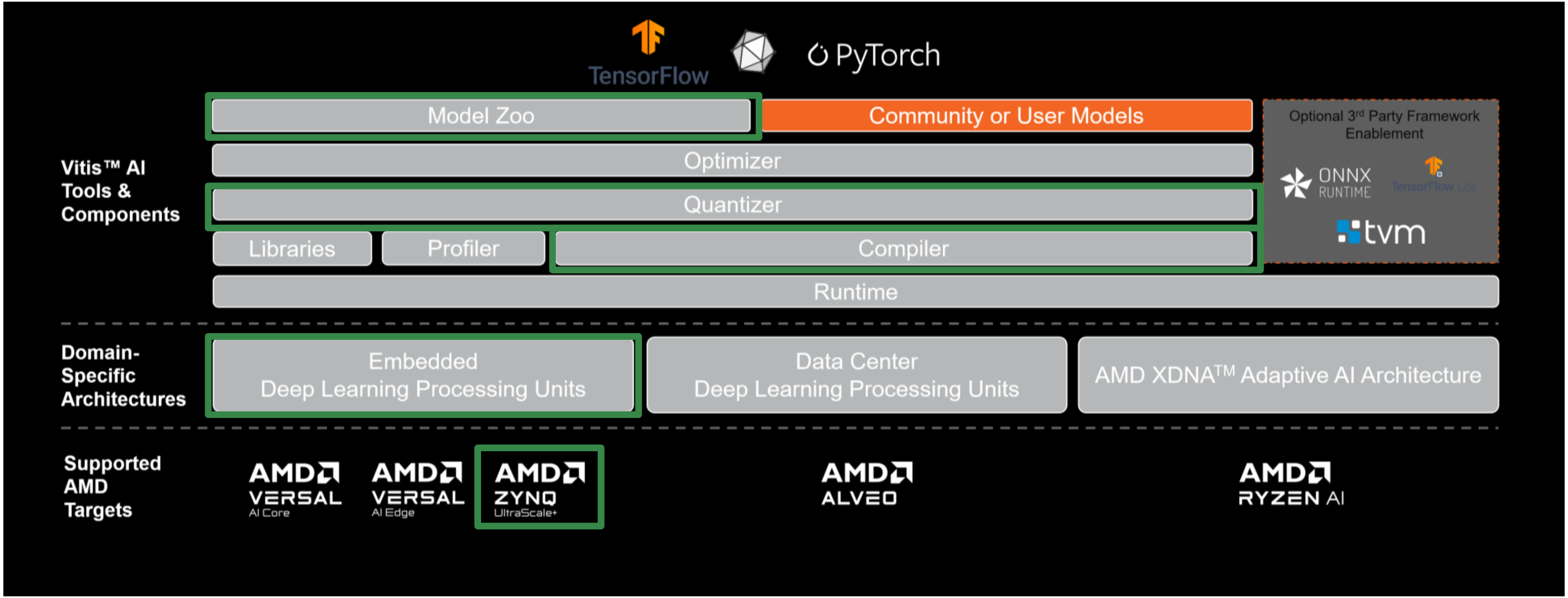
Getting Started with Vitis AI and Model Zoo

Where to start evaluation?

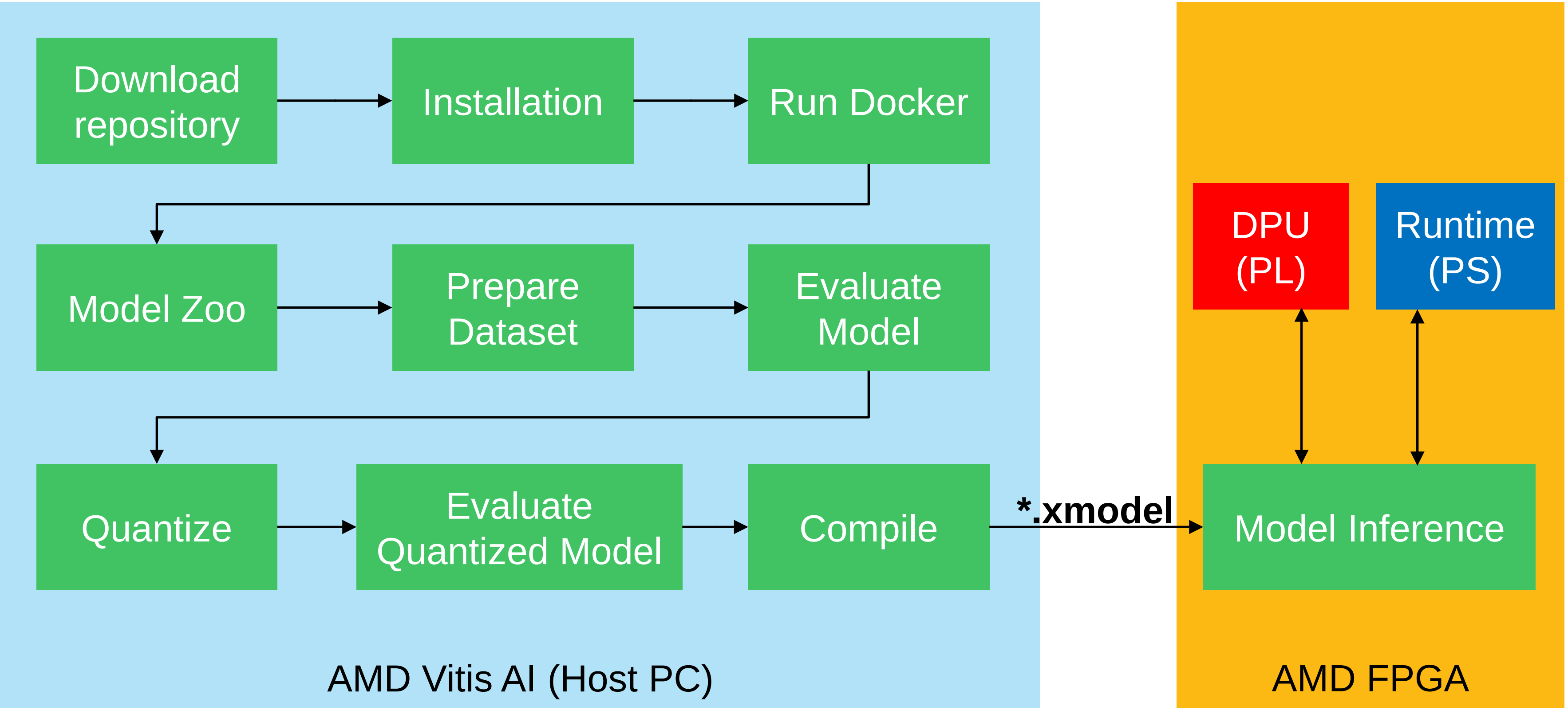


- Vitis AI — Vitis™ AI 3.5 documentation (xilinx.github.io)
- GitHub - Xilinx/Vitis-AI: Vitis AI is Xilinx's development stack for AI inference on Xilinx hardware platforms, including both edge devices and Alveo cards.

AMD Vitis™ AI Integrated Development Environment



Vitis-AI Flow: Get started with Model Zoo



Downloading a model from Vitis AI (ResNet50 Example)

Vitis-AI Docker (v3.0 tag)

GitHub - Xilinx/Vitis-AI: Vitis AI is Xilinx's development stack for AI inference on Xilinx hardware platforms, including both edge devices and Alveo cards.

- **Clone GitHub repository, e.g.:**
  git clone https://github.com/Xilinx/Vitis-AI
  cd Vitis-AI && git checkout v3.0

- **Build CPU or GPU docker image, e.g.:**
  ./docker/docker_build_gpu.sh

- **Run docker, e.g.:**
  ./docker_run.sh xilinx/vitis-ai-gpu:3.0.0.001

DOCKER_TYPE (-t)	TARGET_FRAMEWORK (-f)	Desired Environment
cpu	pytorch	PyTorch cpu-only
	tf2	TensorFlow 2 cpu-only
	tf1	TensorFlow 1.15 cpu-only
gpu	pytorch	PyTorch CUDA-gpu
	opt_pytorch	PyTorch with AI Optimizer CUDA-gpu
	tf2	TensorFlow 2 CUDA-gpu
	opt_tf2	TensorFlow 2 with AI Optimizer CUDA-gpu
	tf1	TensorFlow 1.15 CUDA-gpu
	opt_tf1	TensorFlow 1.15 with AI Optimizer CUDA-gpu
rocm	pytorch	PyTorch ROCm-gpu
	opt_pytorch	PyTorch with AI Optimizer ROCm-gpu
	tf2	TensorFlow 2 ROCm-gpu
	opt_tf2	TensorFlow 2 with AI Optimizer ROCm-gpu

Vitis-AI Docker Preview

```
==========
== CUDA ==
==========

CUDA Version 11.3.1

Container image Copyright (c) 2016-2022, NVIDIA CORPORATION & AFFILIATES. All rights reserved.

This container image and its contents are governed by the NVIDIA Deep Learning Container License.
By pulling and using the container, you accept the terms and conditions of this license:
https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license

A copy of this license is made available in this container at /NGC-DL-CONTAINER-LICENSE for your convenience.

Setting up dkolosov 's environment in the Docker container...
usermod: no changes
Running as vitis-ai-user with ID 0 and group 0

Docker Image Version: 3.0.0.001   (GPU)
Vitis AI Git Hash: 9e7bea642
Build Date: 2023-02-02
WorkFlow: tf2

vitis-ai-user@dev:/workspace$
```

Vitis-AI Model Zoo – Downloading models

Over 100+ models from various frameworks (TF1.x, TF2.x, PyTorch)

Download available models, using provided python script Vitis-AI/model_zoo/downloader.py

1. Select framework (tf1, tf2, pytorch)
2. Select models (all or specific ones)
3. Select target device (GPU, MPSOC platforms, versal platforms, etc)

Vitis-AI Model Zoo – Downloading ResNet50

```
vitis-ai-user@dev:/workspace$ cd model_zoo/
vitis-ai-user@dev:/workspace/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:tf2
chose model
0 : all
1 : tf2_efficientnet-b0_imagenet_224_224_0.78G_3.0
2 : tf2_resnet50_imagenet_224_224_7.76G_3.0
3 : tf2_erfnet_cityscapes_512_1024_54G_3.0
4 : tf2_efficientnet-lite_imagenet_224_224_0.77G_3.0
5 : tf2_2d-unet_nuclei_128_128_5.31G_3.0
6 : tf2_yolov3_coco_416_416_65.9G_3.0
7 : tf2_inceptionv3_imagenet_299_299_11.5G_3.0
8 : tf2_mobilenetv3_imagenet_224_224_132M_3.0
9 : tf2_mobilenetv1_imagenet_224_224_1.15G_3.0
input num:2
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:1
tf2_resnet50_imagenet_224_224_7.76G_3.0.zip
                                              100.0%|100%
done
vitis-ai-user@dev:/workspace/model_zoo$
```

Choose framework

Choose model

Choose type

Vitis-AI Model Zoo Preview

AMD

Vitis AI 3.5 Model Zoo

Copyright (c) 2023 Advanced Micro Devices, Inc.

COPY CSV JSON PRINT

SEARCH:

Use the search function in the upper right to locate a model

Task	Market Specialization	Application	Framework	Vitis-AI Model Name Zoo Name	License Restriction(s)	Copyleft Model Zoo	Model Architecture	Model Research Publication	Dataset
	Industrial Vision / Robotics	Interest Point Detection and Description	TensorFlow	tf_superpoint_3.5		No	SuperPoint	https://arxiv.org/abs/2107.03601	COCO 2014
Depth Estimation	Industrial Vision / Robotics	Binocular depth estimation	PyTorch	pt_fadnet_0.65_3.5		No	FADNet	https://arxiv.org/abs/2003.10758	Sceneflow
Depth Estimation	Industrial Vision / Robotics	Stereo Depth Estimation	PyTorch	pt_fadnet_3.5		No	FADNet	https://arxiv.org/abs/2003.10758	Sceneflow
Depth Estimation	Industrial Vision / Robotics	Stereo Depth Estimation	PyTorch	pt_fadnetv2_0.51_3.5		No	FADNet	https://arxiv.org/abs/2003.10758	Sceneflow
Depth Estimation	Industrial Vision / Robotics	Stereo Depth Estimation	PyTorch	pt_fadnetv2_3.5		No	FADNet	https://arxiv.org/abs/2003.10758	Sceneflow
Image Classification		General	PyTorch	pt_inceptionv3_0.3_3.5	Non-Commercial Use Only	No	Inception-v3	https://arxiv.org/abs/1512.00567	ILSVRC2012
Image Classification		General	PyTorch	pt_inceptionv3_0.4_3.5	Non-Commercial Use Only	No	Inception-v3	https://arxiv.org/abs/1512.00567	ILSVRC2012
Image Classification		General	PyTorch	pt_inceptionv3_0.5_3.5	Non-Commercial Use Only	No	Inception-v3	https://arxiv.org/abs/1512.00567	ILSVRC2012
Image Classification		General	PyTorch	pt_inceptionv3_0.6_3.5	Non-Commercial Use Only	No	Inception-v3	https://arxiv.org/abs/1512.00567	ILSVRC2012
Image Classification		General	PyTorch	pt_inceptionv3_3.5	Non-Commercial Use Only	No	Inception-v3	https://arxiv.org/abs/1512.00567	ILSVRC2012
Image Classification		General	PyTorch	pt_OFA-depthwise-res50_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_OFA-resnet50_0.88_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_OFA-resnet50_0.74_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_OFA-resnet50_0.45_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_OFA-resnet50_0.60_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_OFA-resnet50_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_resnet50_0.3_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_resnet50_0.4_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_resnet50_0.5_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012
Image Classification		General	PyTorch	pt_resnet50_0.6_3.5	Non-Commercial Use Only	No	ResNet50	https://arxiv.org/abs/1512.03385	ILSVRC2012

[Vitis AI Model Zoo — Vitis™ AI 3.5 documentation (xilinx.github.io)]

Pose Detection

NLP

Image Classification

Object Detection

Depth Estimation

Super Resolution

Semantic Segmentation

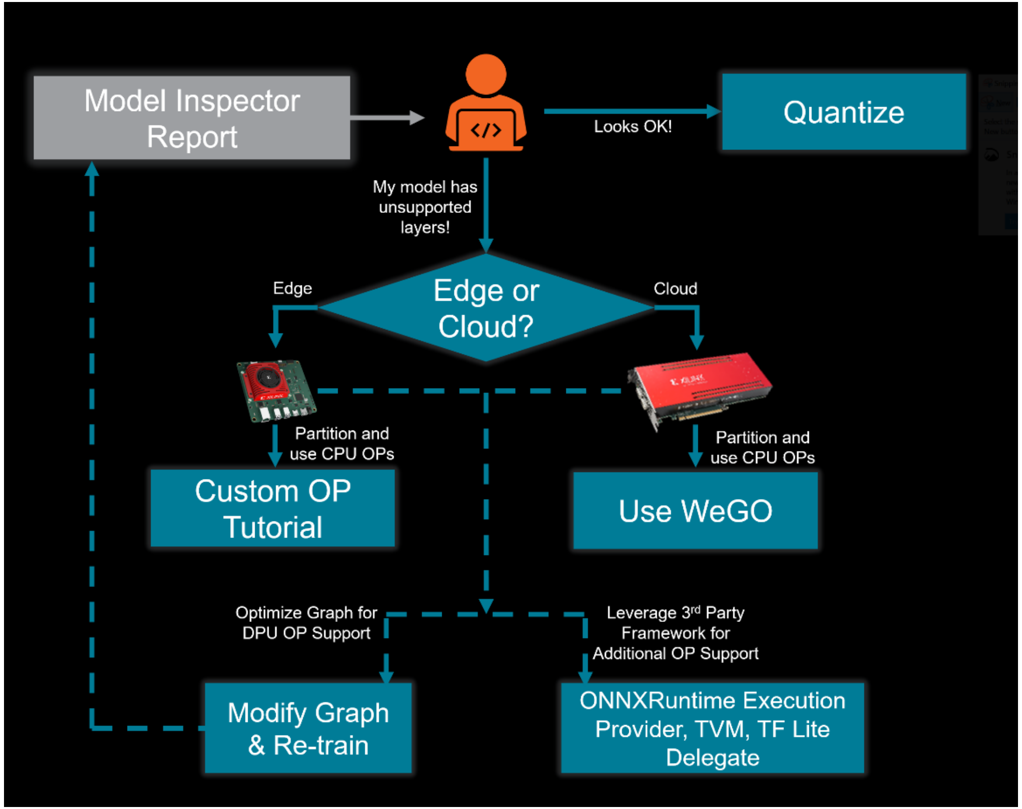
Industrial Vision/Robotics

Prepare Dataset (for ResNet50)

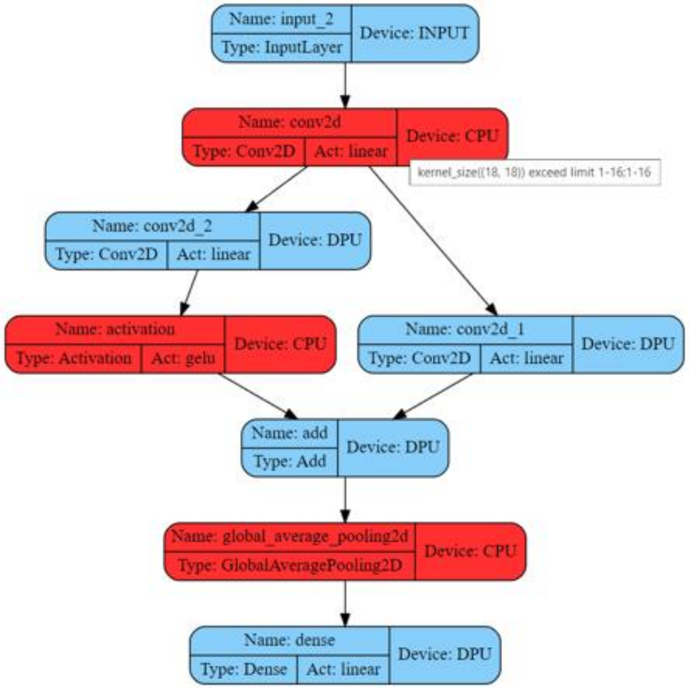
- **Download Dataset from [ImageNet (image-net.org)]**
  - Need account registration
  - DL validation set from ILSVRC-2012

- **Prepare Dataset**
  1. Unzipping and placing into the right directory
  2. Sort images into class type folder (bash script can be provided by Avnet Silica)
  3. Pre-Process dataset using bash scripts provided with model zoo download

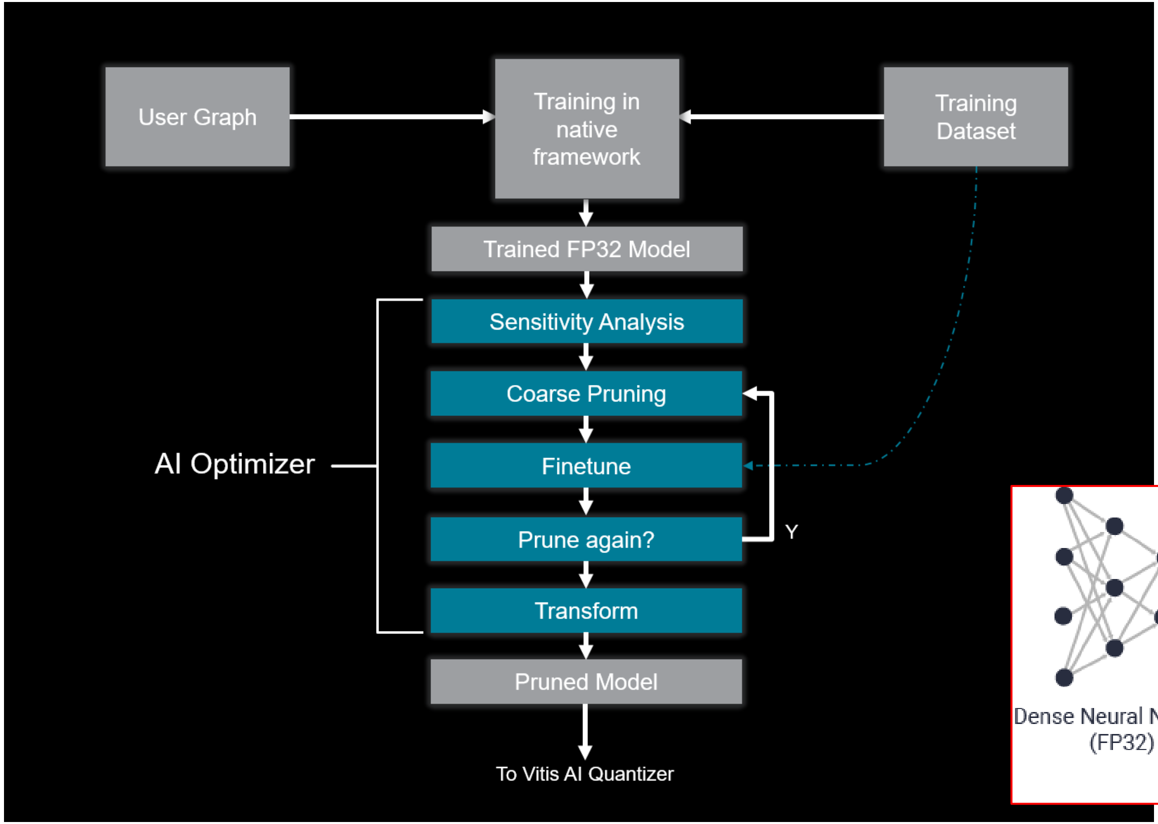
(Optional) Model Inspector

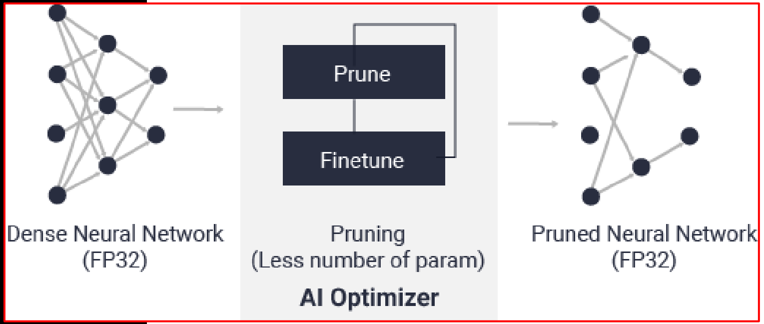


Example:



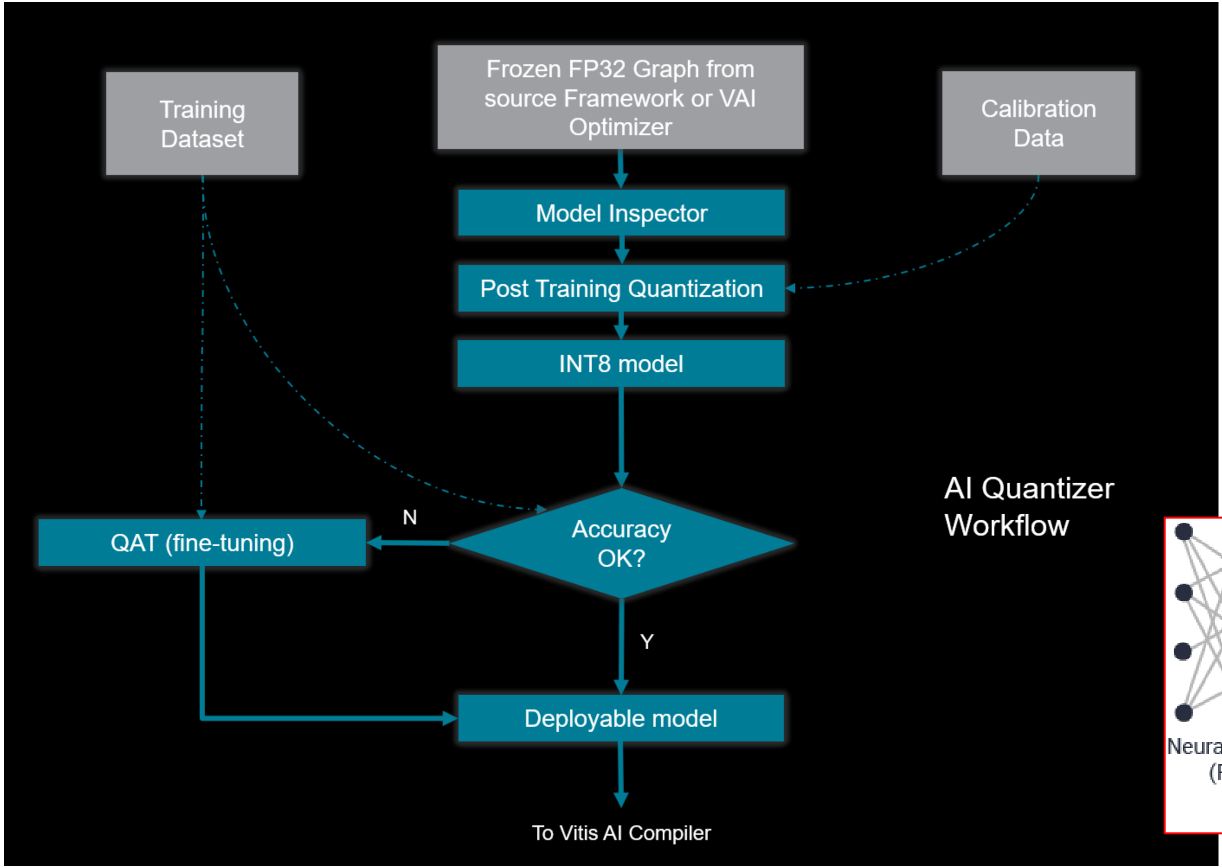
(Optional) Vitis AI Optimizer Pruning

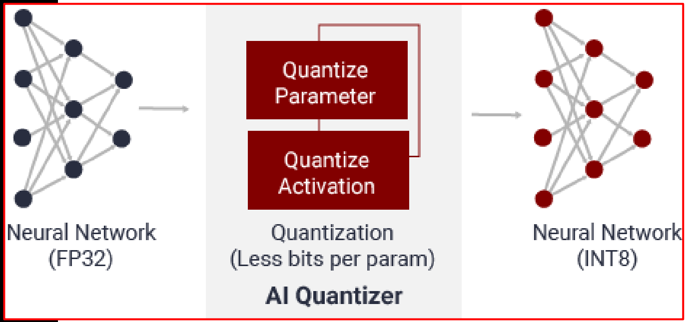




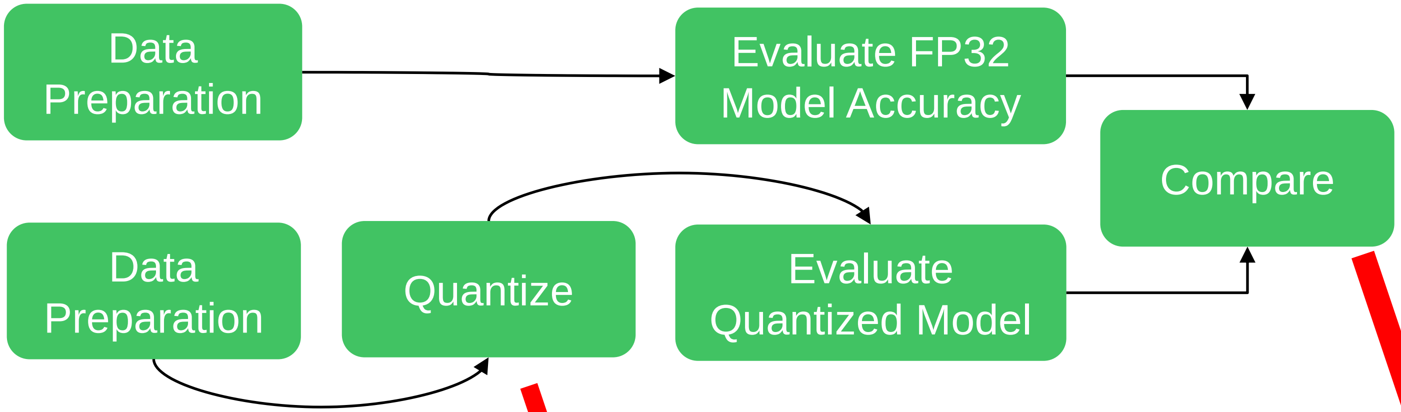
AVNET SILICA

Vitis AI Quantizer Workflow





Evaluation Steps




```
# run quantization
quantizer = vitis_quantize.VitisQuantizer(orig_model)
ptq_model = quantizer.quantize_model(calib_dataset=x_train[0:CALIB_SIZE])
```

```
[VAI INFO] Update custom_layer_type: []
[VAI INFO] Start CrossLayerEqualization...
10/10 [==============================] - 0s 40ms/step
[VAI INFO] CrossLayerEqualization Done.
[VAI INFO] Start Quantize Calibration...
32/32 [==============================] - 1s 10ms/step
[VAI INFO] Quantize Calibration Done.
[VAI INFO] Start Post-Quantize Adjustment...
[VAI INFO] Post-Quantize Adjustment Done.
[VAI INFO] Quantization Finished.
```

```
Original Accuracy: 98.99%
PTQ Accuracy:      99.00% - Difference: -0.01
```

ResNet50 Quantization Results Example

- **Compare results**

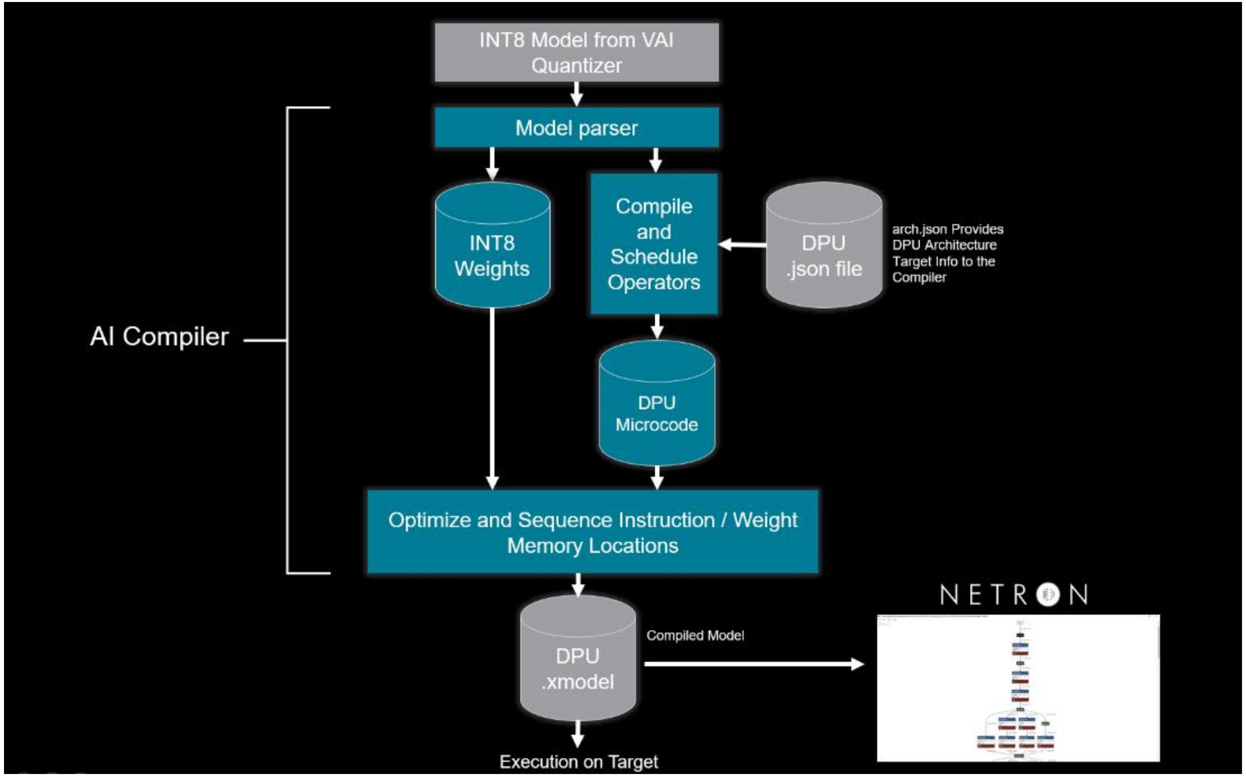
	Accuracy	Accuracy Top5	Size
(FP32) float h5	75.10%	93.10%	100 MB
(INT8) quant h5	75.90%	93.10%	100 MB
(INT8) *.xmodel	-	-	26 MB

- **Online Model Zoo Details & Performance**
  https://xilinx.github.io/Vitis-AI/3.0/html/docs/reference/ModelZoo_Github_web.htm

Compile for target AMD Hardware

Requirements:

- Quantized Model
- DPU *.json file



AVNET SILICA

Vitis AI Compiler for TensorFlow2

```
vai_c_tensorflow2 -m /PATH/TO/quantized.h5
                  -a /PATH/TO/arch.json
                  -o /OUTPUTPATH
                  -n netname
```

Outputs:
netname.xmodel
meta.json

```
(vitis-ai-tensorflow) Vitis-AI /workspace > vai_c_tensorflow2 -h
**************************************************
* VITIS_AI Compilation - Xilinx Inc.
**************************************************
usage: vai_c_tensorflow2 [-h] [-m MODEL] [-a ARCH] [-o OUTPUT_DIR]
                         [-n NET_NAME] [-e OPTIONS]

optional arguments:
  -h, --help            show this help message and exit
  -m MODEL, --model MODEL
                        h5 model file
  -a ARCH, --arch ARCH  json file
  -o OUTPUT_DIR, --output_dir OUTPUT_DIR
                        output directory
  -n NET_NAME, --net_name NET_NAME
                        prefix-name for the outputs
  -e OPTIONS, --options OPTIONS
                        extra options. Use --options '{"input_shape":
                        "1,224,224,3"}' to specify input shape manually, or
if
                        there are more than 1 inputs, use --options
                        '{"input_shape": {"data_op0": "1,224,224,3",
                        "data_op1": "1,112,112,3"}}'. Use --options
                        '{"batchsize": 4}' to modify the batchsize. Use
                        --options '{"plugins": "plugin0,plugin1"}' to
specify
                        plugin libraries. Use --options '{"output_ops":
                        "op_name0,op_name1"}' to specify output ops
(vitis-ai-tensorflow) Vitis-AI /workspace >
```

Edge Deployment on KRIA KV260

How can you get started?

Pre-built images:

- PetaLinux with DPU core
- Ubuntu 22.04 + PYNQ

Custom image, with Vivado or Vitis flow:

- Adding DPU to your hardware image
- Vitis AI Libraries / drivers
  - Offline install (during build time via PetaLinux configuration)
  - Online install (during run time via dnf utility)

Starting with Pre-built PetaLinux Image

Requirements:

- Download corresponding SD Card image with DPU (UG1354)
- Flash SD card, e.g. with BalenaEtcher
- [KRIA only] Flash latest firmware version (QSPI)
- Boot KV260
- **Download test images/videos (UG1354) and copy to KV260**
- Copy *.xmodel (custom or from Model Zoo) to KV260

Starting with PetaLinux

Check DPU settings with: **xdputil query**

- IP Version
- VAI Version
- DPU Architecture
- DPU Frequency
- Fingerprint ID
- # cores

```
root@xilinx-kv260-starterkit-20222:~# xdputil query
{
    "DPU IP Spec":{
        "DPU Core Count":1,
        "IP version":"v4.1.0",
        "generation timestamp":"2022-11-30 19-15-00",
        "git commit id":"ce8dd1",
        "git commit time":2022113019,
        "regmap":"1to1 version"
    },
    "VAI Version":{
        "libvaip-core.so":"Xilinx vaip Version: 1.0.0-a176db67b19f94b0a31f9d24ef80322efe4494ad  2022-12-27-01:24:22 ",
        "libvart-runner.so":"Xilinx vart-runner Version: 3.0.0-2efa5fe1e56c2b2c8a7e71e9fc1636242dd50a9f  2022-12-27-00:47:05 ",
        "libvitis_ai_library-dpu_task.so":"Xilinx vitis_ai_library dpu_task Version: 3.0.0-1cccff04dc341c4a6287226828f90aed56005f4f  2022-12-20 10:29:01 [UT
C] ",
        "libxir.so":"Xilinx xir Version: xir-9204ac72103092a7b253a0c23ec7471481656940 2022-12-27-00:46:16",
        "target_factory":"target-factory.3.0.0 860ed0499ab009084e2df3004eeb9ae710c26351"
    },
    "kernels":[
        {
            "DPU Arch":"DPUCZDX8G_ISA1_B4096",
            "DPU Frequency (MHz)":300,
            "IP Type":"DPU",
            "Load Parallel":2,
            "Load augmentation":"enable",
            "Load minus mean":"disable",
            "Save Parallel":2,
            "XRT Frequency (MHz)":300,
            "cu_addr":"0xa0010000",
            "cu_handle":"0xaaaacfe29490",
            "cu_idx":0,
            "cu_mask":1,
            "cu_name":"DPUCZDX8G:DPUCZDX8G_1",
            "device_id":0,
            "fingerprint":"0x101000056010407",
            "name":"DPU Core 0"
        }
    ]
}
root@xilinx-kv260-starterkit-20222:~# 
```


AVNET SILICA

Quick Benchmarking Test

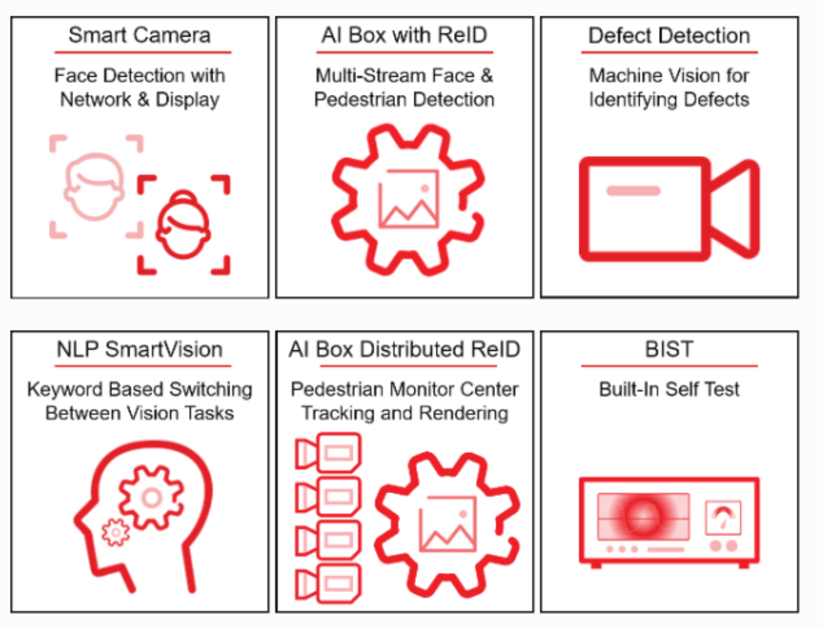
Test any model, with: xdputil benchmark <model.xmodel> <num of threads>

1x thread: xdputil benchmark resnet50.xmodel 1

```
root@xilinx-kv260-starterkit-20222:~# xdputil benchmark resnet50.xmodel 1
Nov 19 09:27:25 xilinx-kv260-starterkit-20222 kernel: [drm] ERT_EXEC_WRITE is obsoleted, use ERT_START_KEY_VAL
WARNING: Logging before InitGoogleLogging() is written to STDERR
I1119 09:27:25.233786  2201 test_dpu_runner_mt.cpp:474] shuffle results for batch...
I1119 09:27:25.234745  2201 performance_test.hpp:73] 0% ...
I1119 09:27:31.234948  2201 performance_test.hpp:76] 10% ...
I1119 09:27:37.235215  2201 performance_test.hpp:76] 20% ...
I1119 09:27:43.235513  2201 performance_test.hpp:76] 30% ...
I1119 09:27:49.235724  2201 performance_test.hpp:76] 40% ...
I1119 09:27:55.235949  2201 performance_test.hpp:76] 50% ...
I1119 09:28:01.236171  2201 performance_test.hpp:76] 60% ...
I1119 09:28:07.236421  2201 performance_test.hpp:76] 70% ...
I1119 09:28:13.236657  2201 performance_test.hpp:76] 80% ...
I1119 09:28:19.236877  2201 performance_test.hpp:76] 90% ...
I1119 09:28:25.237103  2201 performance_test.hpp:76] 100% ...
I1119 09:28:25.237231  2201 performance_test.hpp:79] stop and waiting for all threads terminated....
I1119 09:28:25.243063  2201 performance_test.hpp:85] thread-0 processes 5938 frames
I1119 09:28:25.243116  2201 performance_test.hpp:93] it takes 5851 us for shutdown
I1119 09:28:25.243145  2201 performance_test.hpp:94] FPS= 98.9528 number of frames= 5938 time= 60.0084 seconds.
I1119 09:28:25.243213  2201 performance_test.hpp:96] BYEBYE
Test PASS.
root@xilinx-kv260-starterkit-20222:~#
```


AVNET SILICA

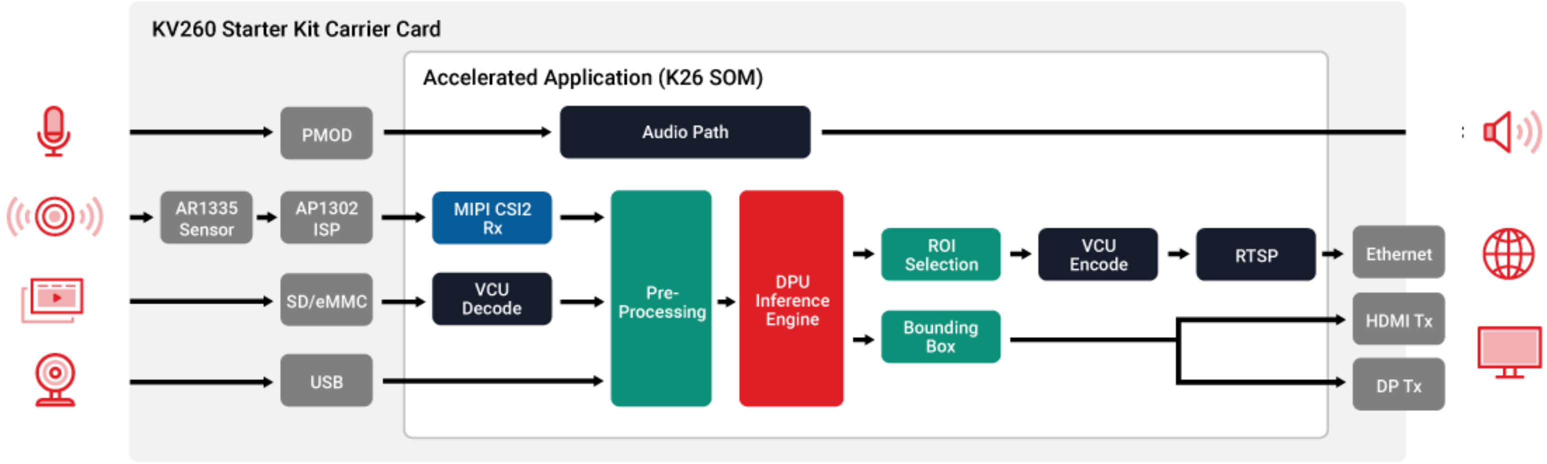
Kria KV260 Vision AI Starter Kit Applications



- [Xilinx / KV260 / kv260-demos-workflow-example · GitLab (avnet.com)]
- [Kria KV260 Vision AI Starter Kit Applications — Kria™ KV260 2022.1 documentation (xilinx.github.io)]

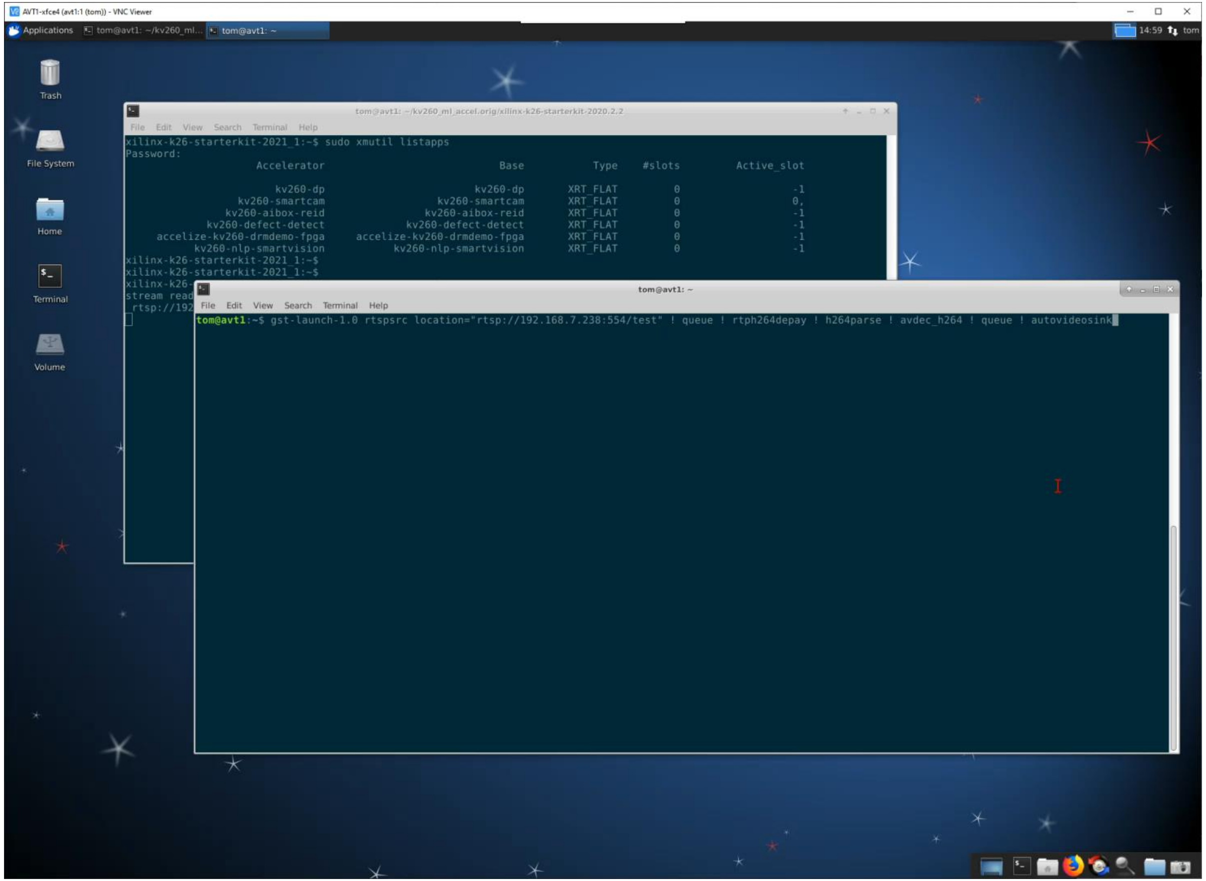
Smart Camera App

- 4K resolution with H.264/H.265 encode
- HDMI or DisplayPort Out
- Face & pedestrian detection model support
- ADAS model support



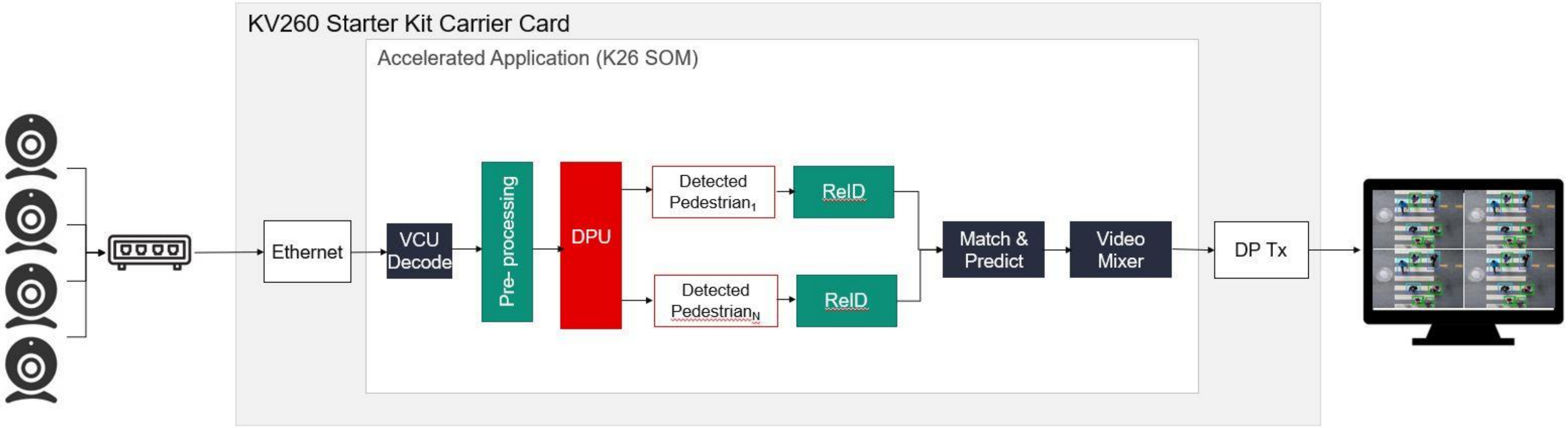
AVNET SILICA

Smart Camera App

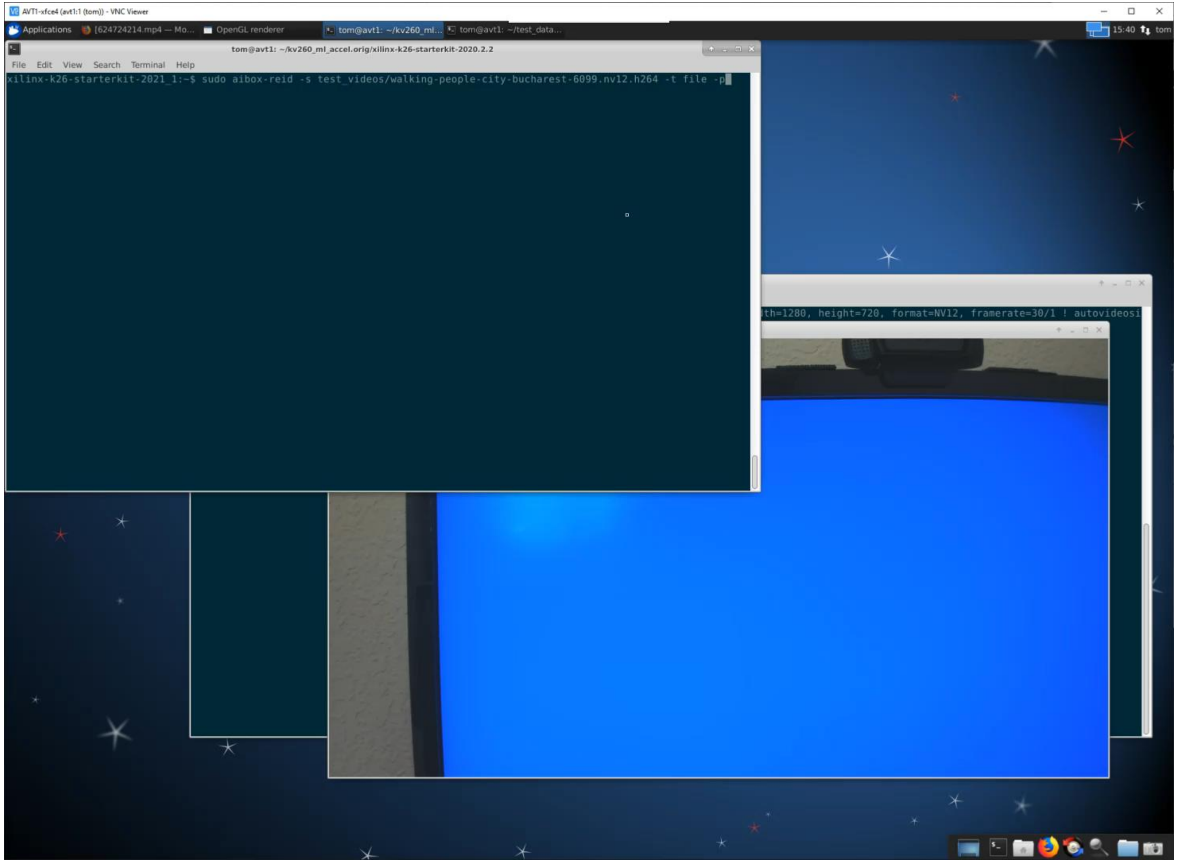


AIBox-ReID App

- Up to 4 input streams from IP cameras
- H.264/H.265 decoding @ 1080p
- HDMI or DisplayPort Out
- Pedestrian detection & ReID model support



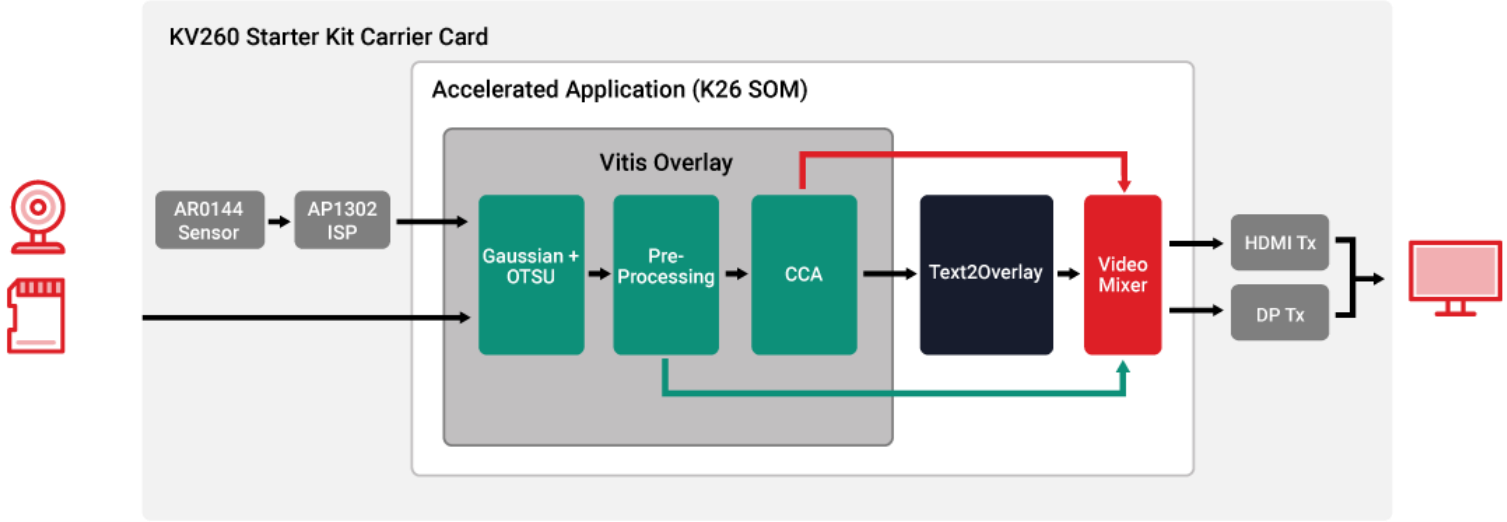
AIBox-ReID App



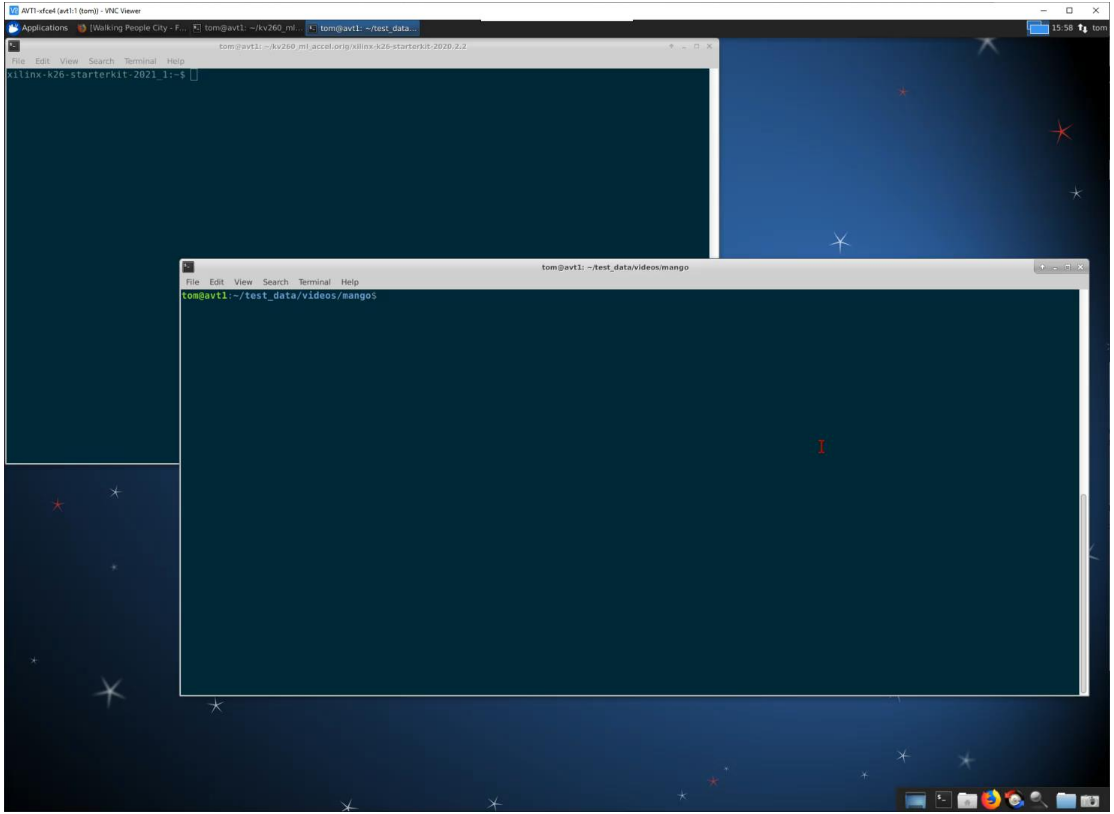
AVNET SILICA

Defect Detect App

- Low latency defect detection pipeline
- Uses traditional image processing algorithms instead of ML
- Leverages the Vitis Vision Library
- HDMI or DisplayPort Out

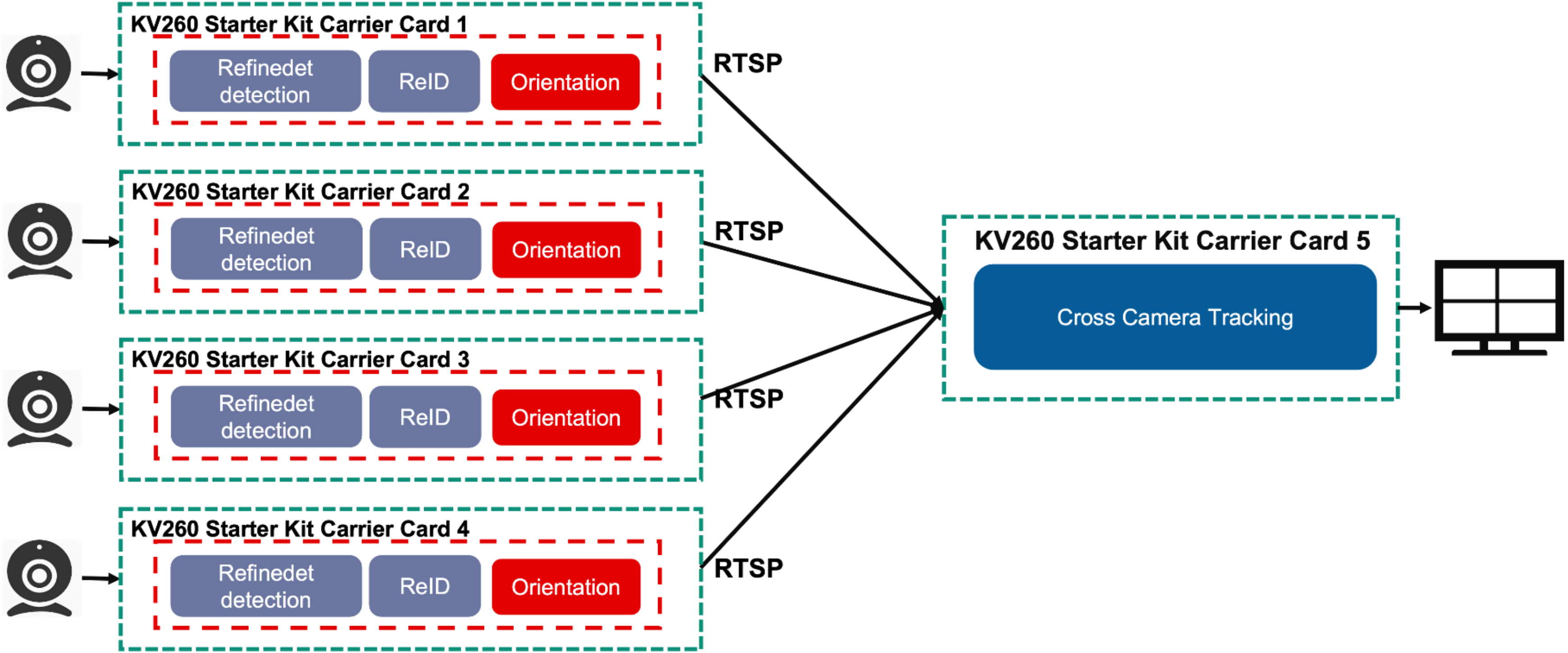


Defect Detect App



AVNET SILICA

AI Box Distributed ReID App



Q&A

Typical Questions

Are all the components of Vitis AI free?

Yes!

As of the 3.5 release all components are free!

For releases <3.5, the Vitis AI Optimizer does require a separate license which can be obtained free-of-charge upon request.

Is Vitis AI a separate download?

Yes! Users can get started by cloning the Vitis AI GitHub repository:

GitHub - Xilinx/Vitis-AI: Vitis AI is Xilinx's development stack for AI inference on Xilinx hardware platforms, including both edge devices and Alveo cards.

What Xilinx Target Device Families and Platforms does Vitis AI Support?

Vitis AI DPUs are available for:

- Zynq 7000
- Zynq Ultrascale+ MPSoC
- Versal AI Edge
- Versal AI Core

How does FPGA compare to CPU and GPU acceleration?

FPGA accelerated networks can run up to 90x faster as compared to CPU. FPGA accelerated networks are on par with GPU accelerated networks for throughput critical applications yet provide support for more custom applications.

FPGA accelerated networks are far superior to GPU accelerated networks for latency critical applications such as autonomous driving.

Lot of scope in reducing on a system level power / cost and creating scalable designs.

Is it possible to deploy the DPUCZ using Yocto flows, or even Ubuntu, rather than PetaLinux?

Yes!

What is important to consider is that each release of the Vitis AI tool and the DPUCZ IP is provided with drivers and a runtime that targets a specific Linux kernel release. Misalignment between the target kernel version can pose challenges and may require extensive code changes.

My DPU implementation does not meet my latency/throughput targets. Is there anything else I can do?

Yes!

Besides modifying architecture and/or taking advantage of pruning within Vitis AI, you can also explore FINN.

FINN implements each layer of a neural network separately and creates like this a custom very low latency and high throughput design in the PL.

Thank You