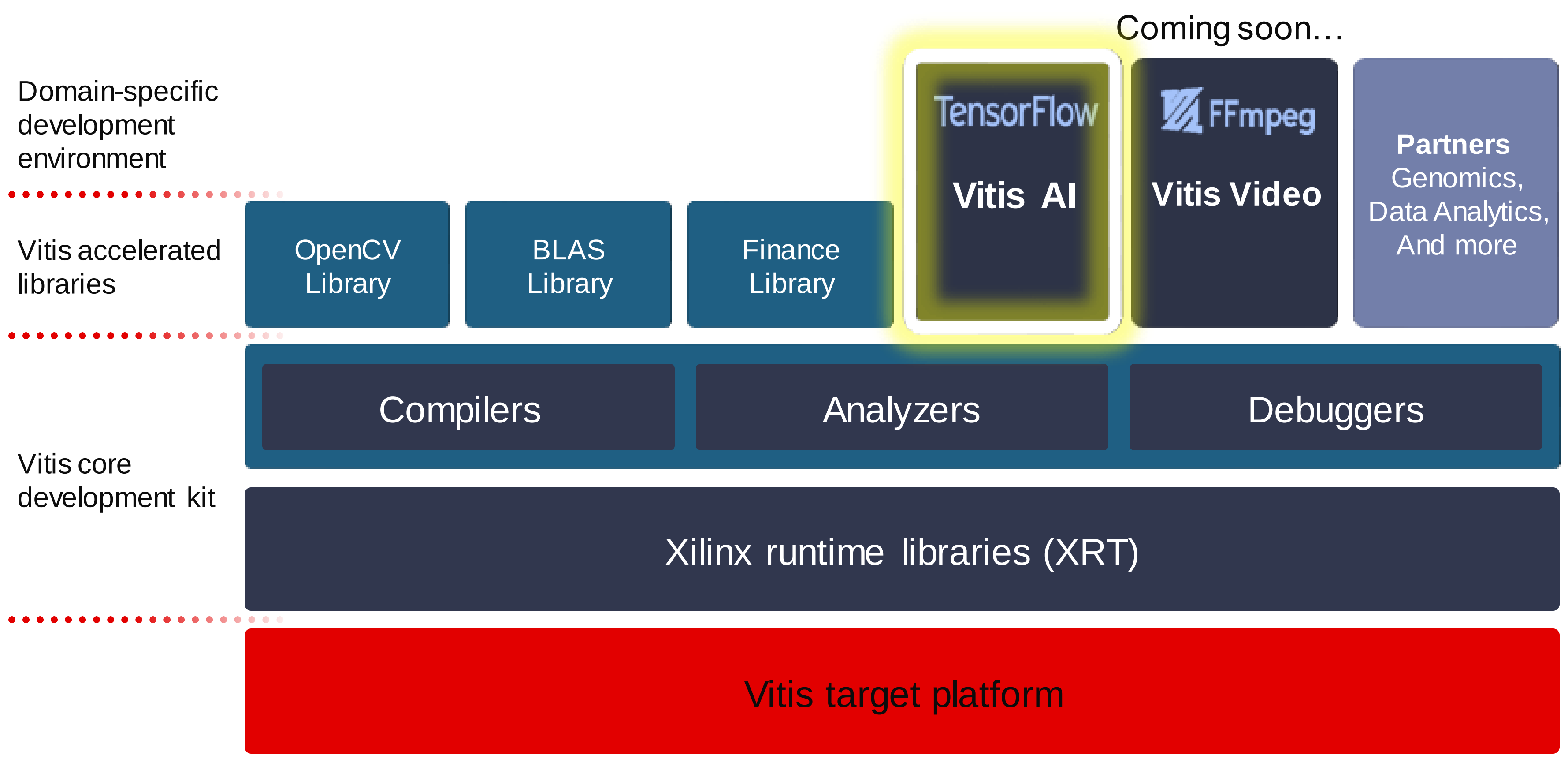
Vitis™ AI start to finish

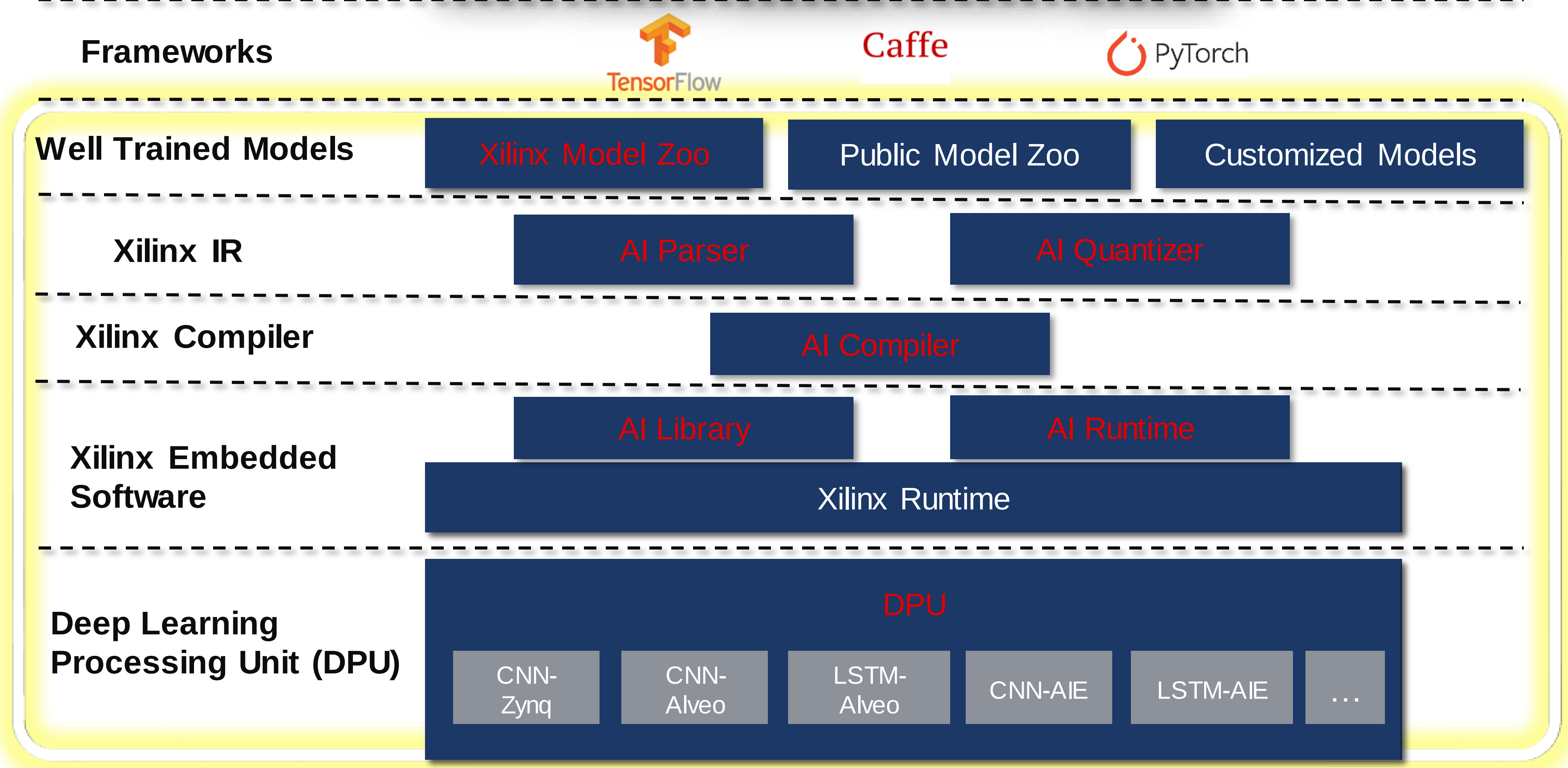
Fan Zhang Ph.D
Software and AI Technical Marketing

Vitis AI: Unified AI Inference Solution Stack

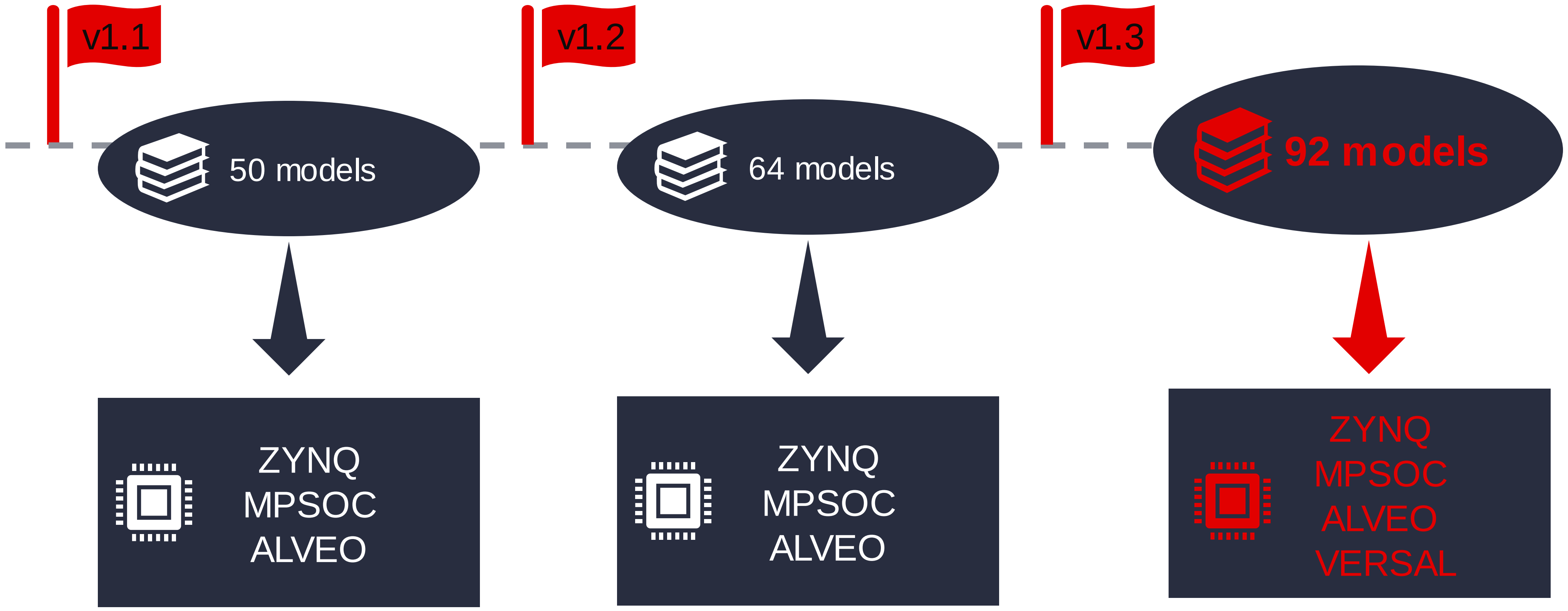

Coming soon…
Domain-specific development environment
Vitis accelerated libraries
Vitis core development kit
OpenCV Library
BLAS Library
Finance Library
TensorFlow
Vitis AI
FFmpeg
Vitis Video
Partners
Genomics, Data Analytics, And more
Compilers
Analyzers
Debuggers
Xilinx runtime libraries (XRT)
Vitis target platform

XILINX.

Vitis AI: Unified AI Inference Solution Stack



Model Zoo

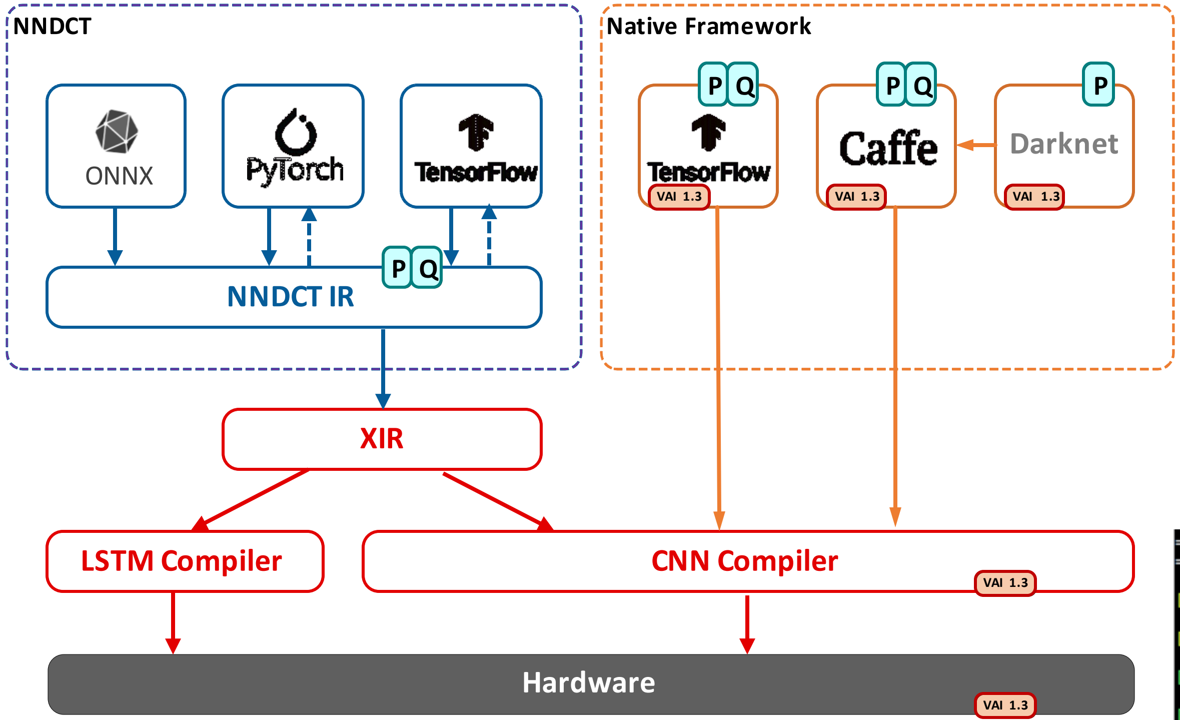

v1.1
v1.2
v1.3
50 models
64 models
92 models
ZYNQ
MPSOC
ALVEO
ZYNQ
MPSOC
ALVEO
ZYNQ
MPSOC
ALVEO
VERSAL

Model Zoo

```
description: inception-v1 classifier on ImageNet.
input size: 224*224
float ops: 3.16G
task: classification
framework: caffe
prune: 'no'
version: 1.3
files:
- name: cf_inceptionv1_imagenet_224_224_3.16G_1.3
  type: float & quantized
  board: GPU
  download link: download link
  checksum: md5sum value
- name: inception_v1
  type: xmodel
  board: zcu102 & zcu104
  download link: download link
  checksum: md5sum value
- name: inception_v1
  type: xmodel
  board: vck190
  download link: download link
  checksum: md5sum value
```

- Yaml file for each model
- Link for different overlays
- Readable from AI Library

AI Parser & Quantizer: Workflow

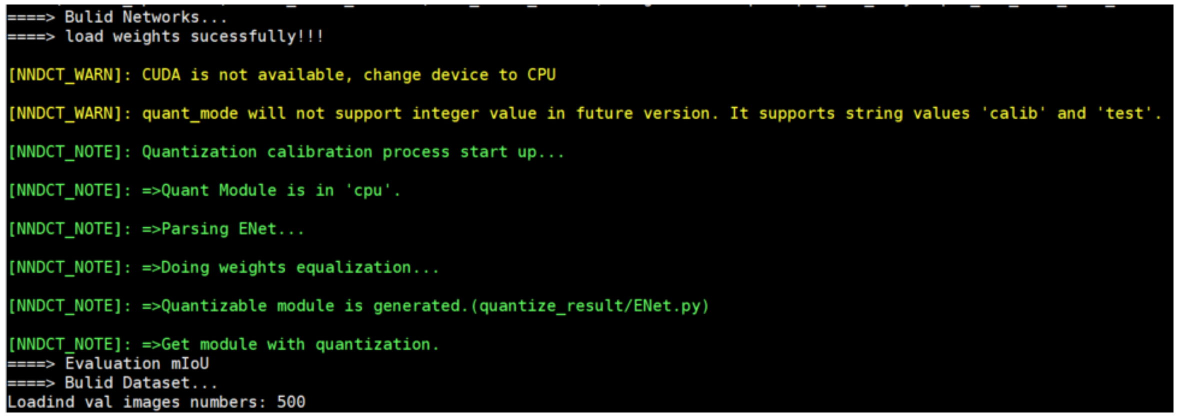


I. Gather batch data distribution

II. Configure bit width position base on hardware constraint

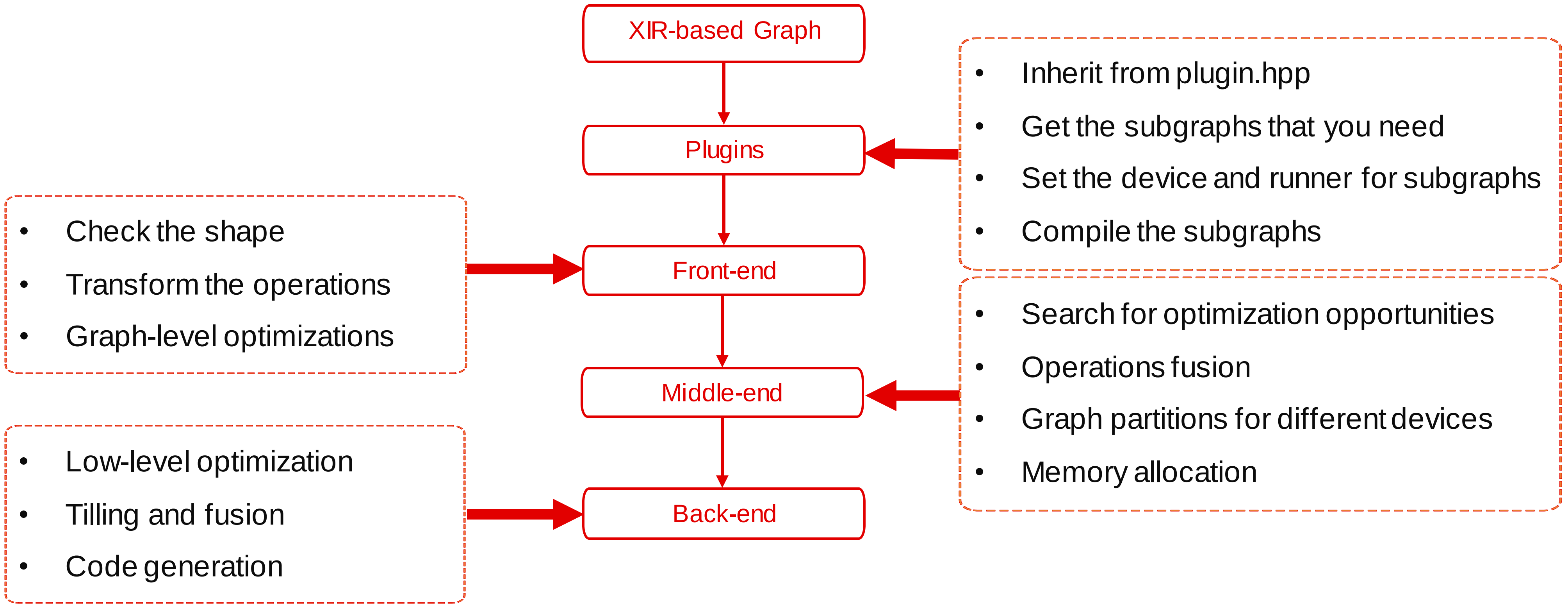
III. Reconstruct OP behavior

IV. Add fix-neuron OP at correct position



```
====> Bulid Networks...
====> load weights sucessfully!!!

[NNDCT_WARN]: CUDA is not available, change device to CPU

[NNDCT_WARN]: quant_mode will not support integer value in future version. It supports string values 'calib' and 'test'.

[NNDCT_NOTE]: Quantization calibration process start up...

[NNDCT_NOTE]: =>Quant Module is in 'cpu'.

[NNDCT_NOTE]: =>Parsing ENet...

[NNDCT_NOTE]: =>Doing weights equalization...

[NNDCT_NOTE]: =>Quantizable module is generated.(quantize_result/ENet.py)

[NNDCT_NOTE]: =>Get module with quantization.
====> Evaluation mIoU
====> Bulid Dataset...
Loadind val images numbers: 500
```

AI Compiler



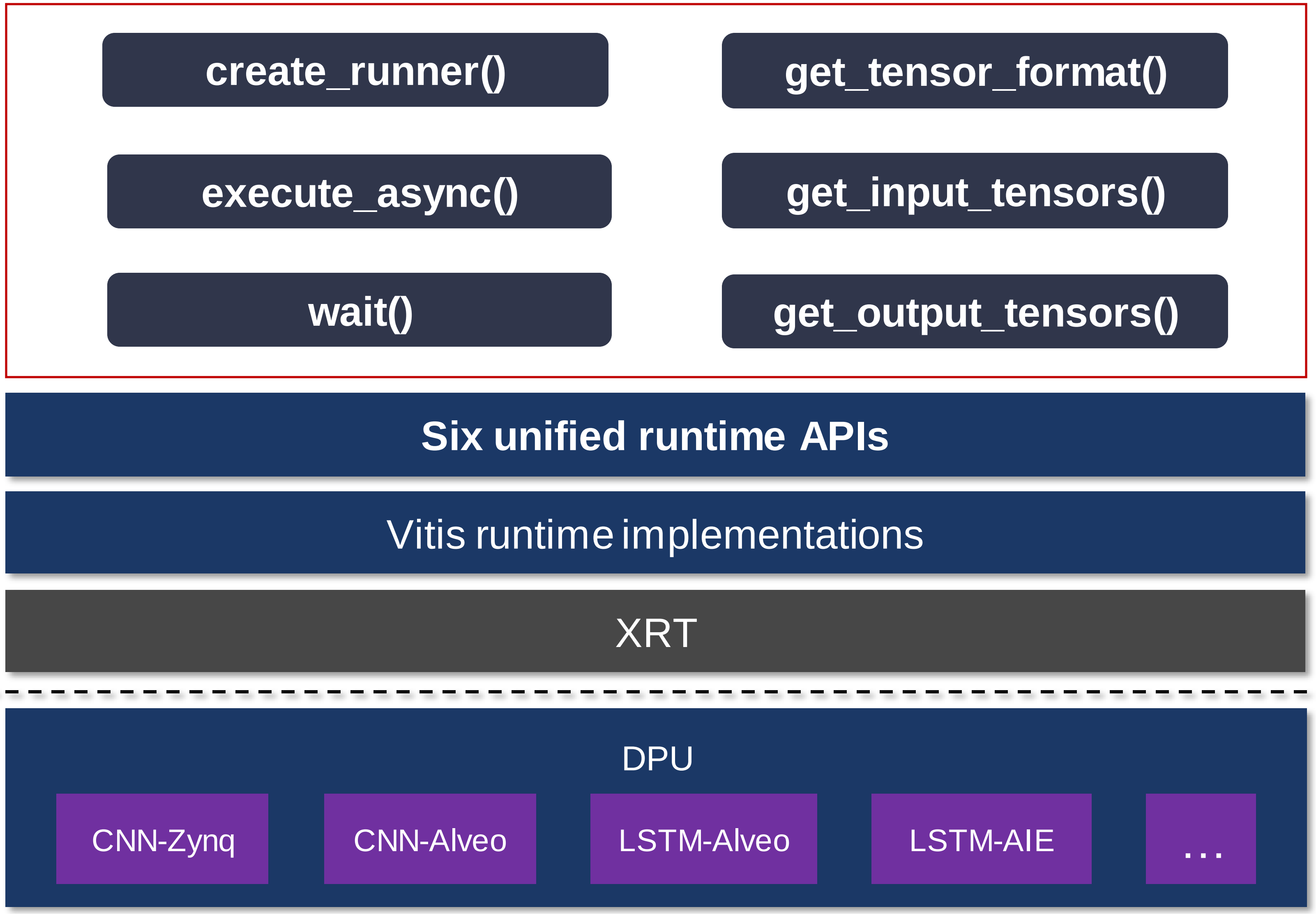
- Inherit from plugin.hpp
- Get the subgraphs that you need
- Set the device and runner for subgraphs
- Compile the subgraphs

- Check the shape
- Transform the operations
- Graph-level optimizations

- Search for optimization opportunities
- Operations fusion
- Graph partitions for different devices
- Memory allocation

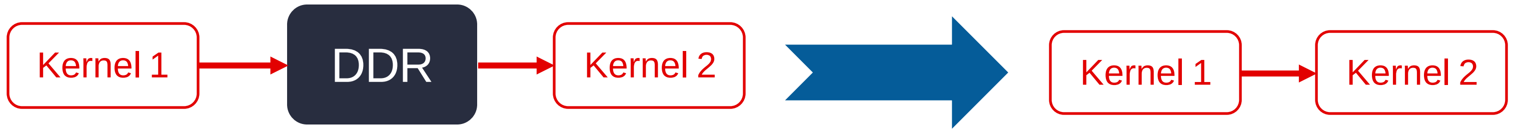
- Low-level optimization
- Tilling and fusion
- Code generation

VART：Unified runtime APIs



Unified Vitis AI runtime with same six APIs across edge and cloud

VART：Zero Copy

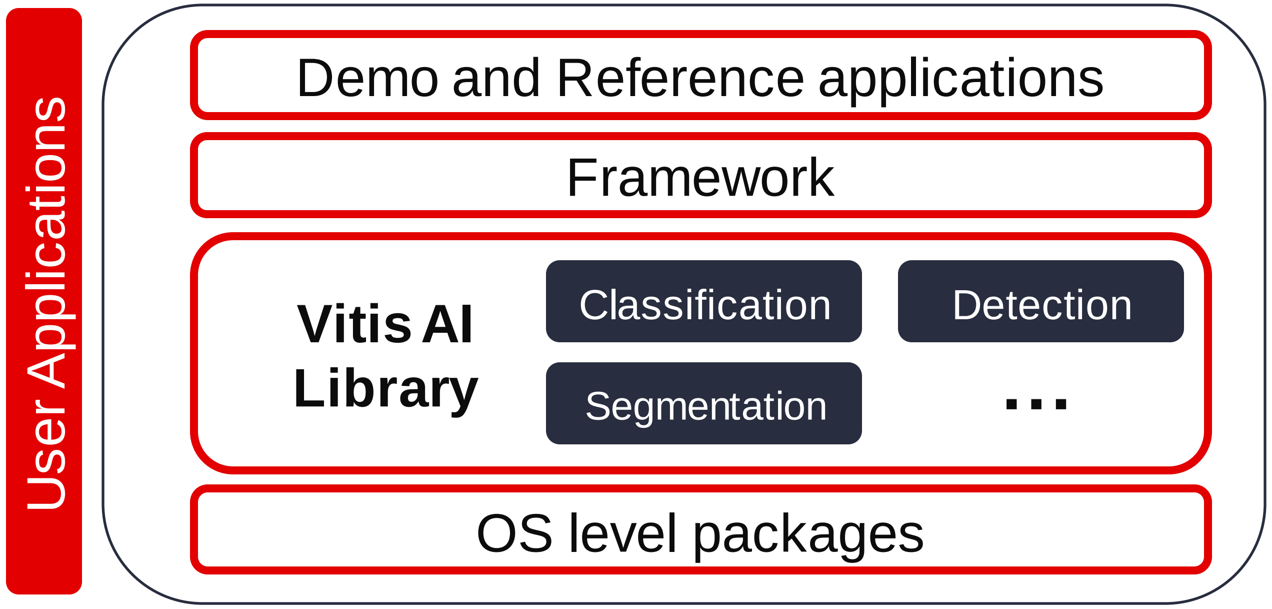


get_location()	get_tensor()
data()	data_phy()
copy_from_host()	copy_to_host()
sync_for_read()	sync_for_write()

New added APIs to achieve zero copy

Vitis AI Library: the What?

- Vitis AI Library provides high-level API based libraries across different vision tasks: classification, detection, segmentation and etc.
  - Reference applications to help customers' fast prototyping
  - Optimized codes used in AI applications and products



Ease-of-Use

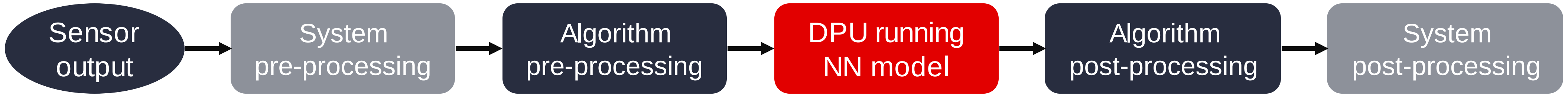
Optimized

Open

XILINX.

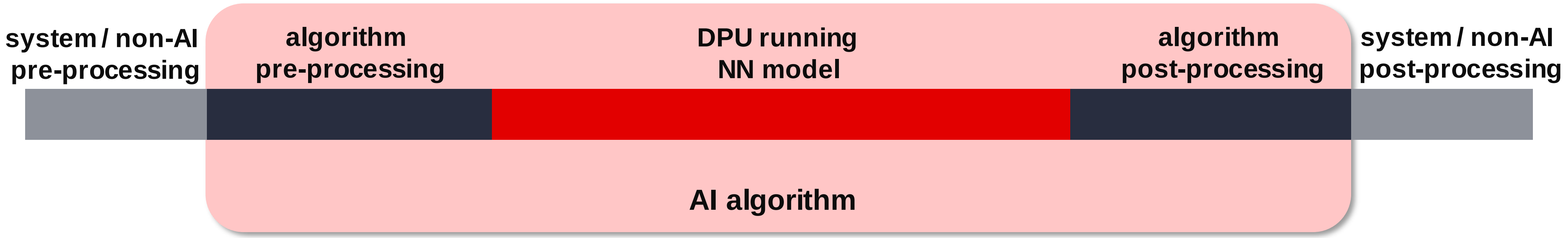
AI Application General Processing Flow

- **A typical abstraction of processing flow:**



- **Algorithm-level processing**
  - Data normalization before sending to DPU
  - Post processing (e.g. bounding boxes decoding in detection)
- **Additional system-level workloads for AI inference**
  - Color conversion / resizing
  - Path planning / control / status update

What Vitis AI Library Provides

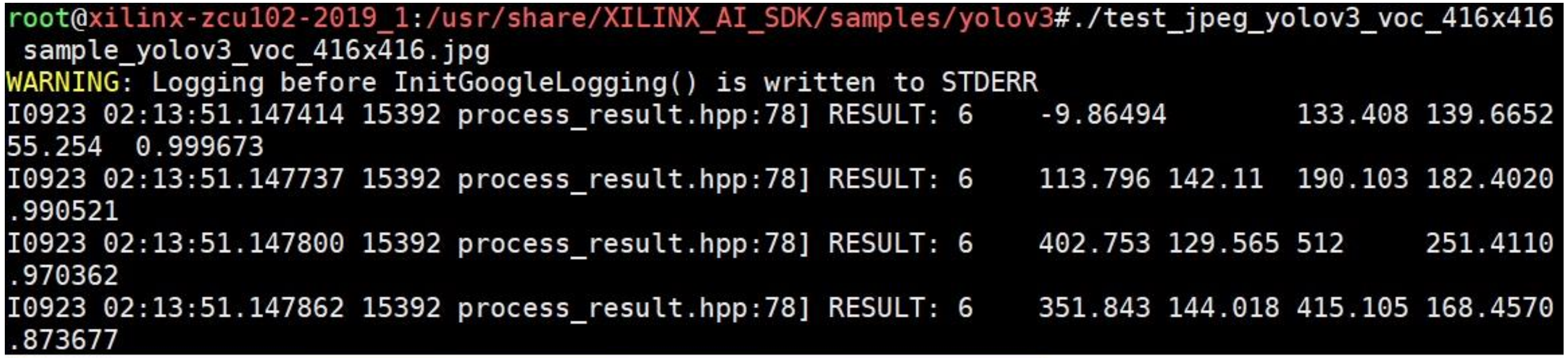

system / non-AI pre-processing
algorithm pre-processing
DPU running NN model
algorithm post-processing
system / non-AI post-processing
AI algorithm

- **AI Library offers libraries for**
  - Algorithm-level optimization
  - Open and easy to extend
  - Directly support models in AI Model Zoo

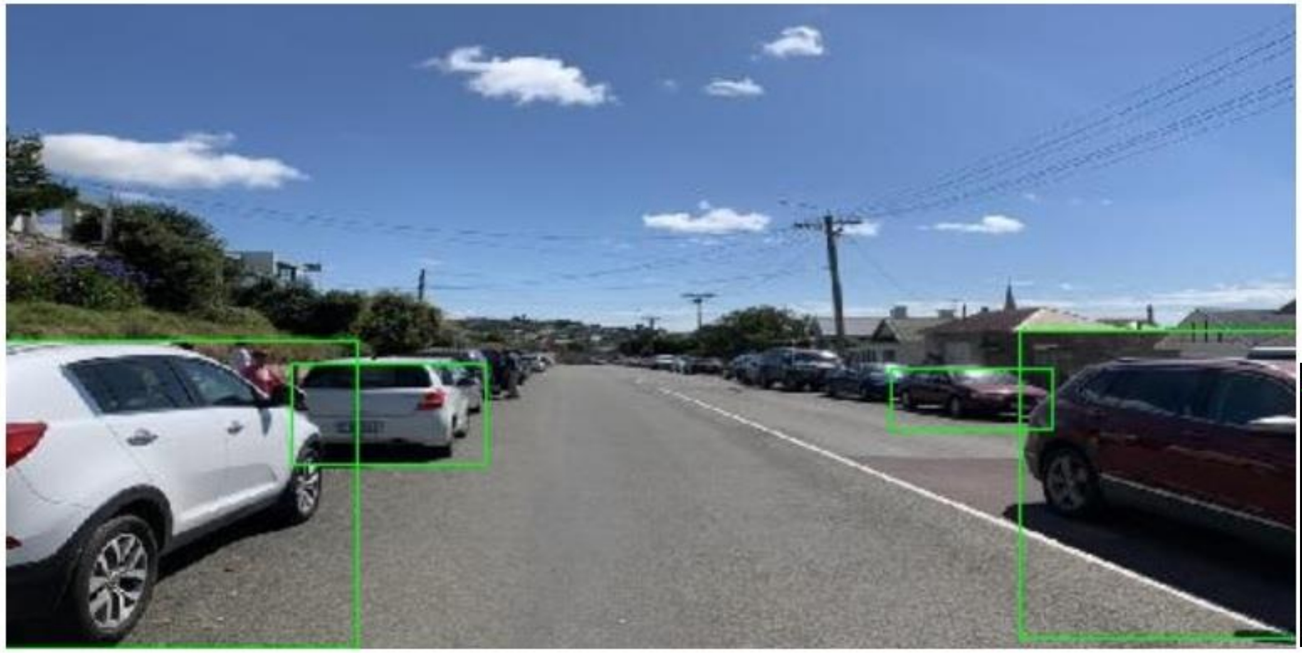
AI Library Samples

- The Vitis AI Library provides image test samples ,video test samples, performance test samples for all the above networks. Each sample has the following four kinds of test sample.
  - test_jpeg_[model type]
  - test_video_[model type]
  - test_performance_[model type]
  - test_accuracy_[model type]
- In addition, the kit provides the corresponding performance test program. For video based testing, we recommend to use raw video for evaluation. Because decoding by software libraries on Arm® CPU may have inconsistent decoding time, which may affect the accuracy of evaluation.

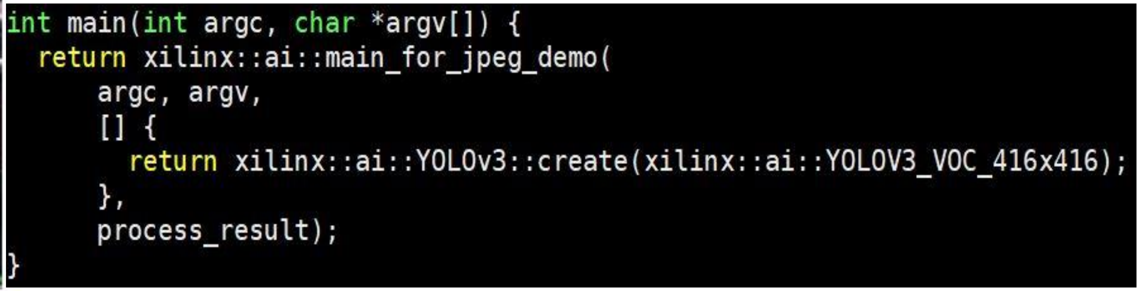
AI Library Samples: test_jpeg_yolov3



```
root@xilinx-zcu102-2019_1:/usr/share/XILINX_AI_SDK/samples/yolov3#./test_jpeg_yolov3_voc_416x416
 sample_yolov3_voc_416x416.jpg
WARNING: Logging before InitGoogleLogging() is written to STDERR
I0923 02:13:51.147414 15392 process_result.hpp:78] RESULT: 6    -9.86494        133.408 139.6652
55.254  0.999673
I0923 02:13:51.147737 15392 process_result.hpp:78] RESULT: 6    113.796 142.11  190.103 182.4020
.990521
I0923 02:13:51.147800 15392 process_result.hpp:78] RESULT: 6    402.753 129.565 512     251.4110
.970362
I0923 02:13:51.147862 15392 process_result.hpp:78] RESULT: 6    351.843 144.018 415.105 168.4570
.873677
```



Fast implementation of YOLOv3 demo by very simple code



```
int main(int argc, char *argv[]) {
  return xilinx::ai::main_for_jpeg_demo(
      argc, argv,
      [] {
        return xilinx::ai::YOLOv3::create(xilinx::ai::YOLOV3_VOC_416x416);
      },
      process_result);
}
```

Easy-to-Use APIs to Deploy Full Algorithm

1

Seamlessly compatible with AI Model Zoo

- Classification, detection, segmentation and others

2

Samples for fast prototyping

- Every algorithm has several samples, image, video and performance benchmarking
- Complicated samples can be refer to AI Demo Zoo which is also built on AI Library

3

High-level APIs to deploy algorithm

- No need to consider algorithm-level processing and DPU running codes

4

Support multiple deploying approaches

- Besides suggested high-level APIs, DPU running can be also controlled by users

XILINX

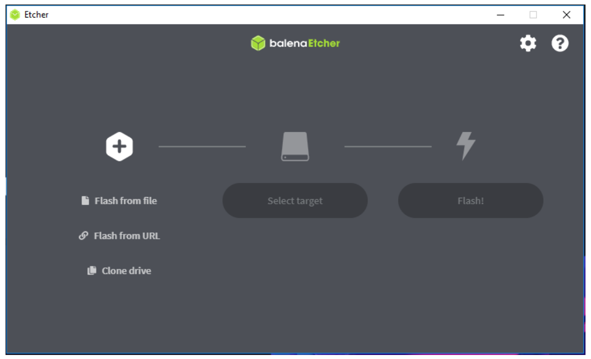
DPU Overlays

Example	DPU	Application (C, R, B, F) C – CNN R – RNN B – Bert F – Random Forest	Hardware platform (AD, AH, VD, VH, ZD) AD – Alveo DDR AH – Alveo HBM VD – Versal DDR with AIE & PL VH – Versal HBM VP – Versal DDR with PL only ZD – Zynq DDR	Quantization Method (X, F, I) X – DECENT F – Float threshold I – Integer threshold M – Metropolis R – RNN	Quantization Bitwidth (4, 8, 16, M) 4 – 4 bit 8 – 8 bit 16 – 16 bit M – Mixed Precision	Design Target (G, H, L, P, C) G – General purpose H – High throughput L – Low latency C – Cost optimized	Major	Minor	Patch	DPU name
DPUv1	DPU	C	AD	X	8	G	3	0	0	**DPU-CADX8G-3.0.0**
DPUv2	DPU	C	ZD	X	8	G	1	4	1	**DPU-CZDX8G-1.4.1**
DPUv3e	DPU	C	AH	X	8	H	1	0	0	**DPU-CAHX8H-1.0.0**
DPUv3me	DPU	C	AH	X	8	L	1	0	0	**DPU-CAHX8L-1.0.0**
DPUv3int8	DPU	C	AD	F	8	H	1	0	0	**DPU-CADF8H-1.0.0**
XRNN	DPU	R	AH	R	16	L	1	0	0	**DPU-RAHR16L-1.0.0**
XVDPU	DPU	C	VD	X	8	G	1	0	0	**DPU-CVDX8G-1.0.0**
DPUv4e	DPU	C	VD	X	8	H	1	0	0	**DPU-CVDX8H-1.0.0**

To build up the Demo

① Please get the boards ready.

② Install image flashing tool *eg. etcher*
https://etcher.io/



Vitis AI v1.3 will be available on Dec 18th

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

XILINX®

Thank You

Xilinx Core Values

- Excellence
  - Question, learn, and innovate for exceptional results
- Teamwork
  - Work together in the best interest of Xilinx
  - Embrace diversity of thought and experience
  - Collaborate effectively and respectfully
- Accountability
  - Own commitments to their full conclusion
  - Deal with the unexpected quickly and professionally
  - Be transparent about issues, see them as opportunities, and learn from them