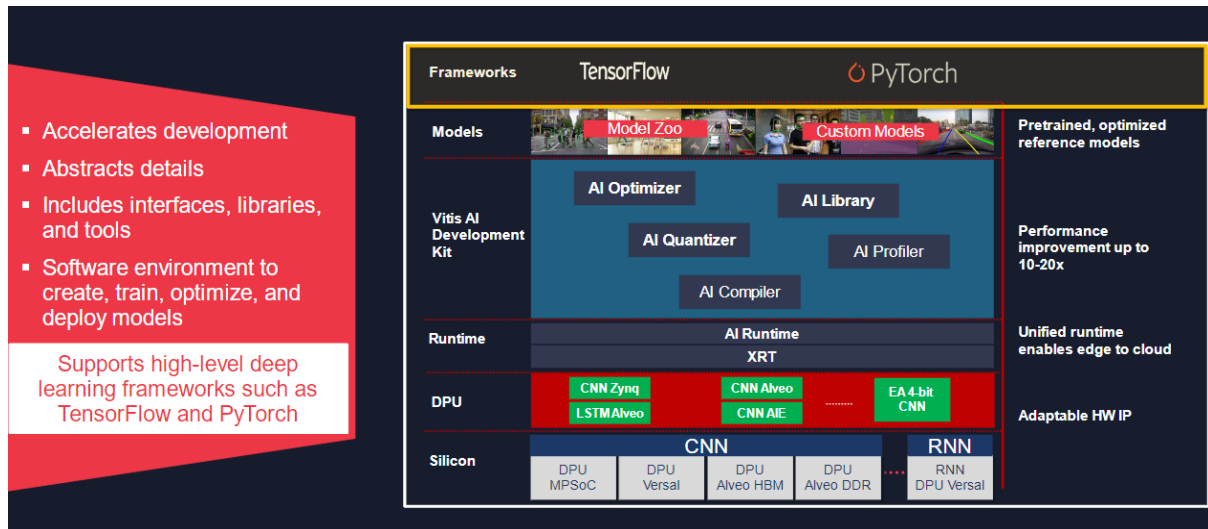
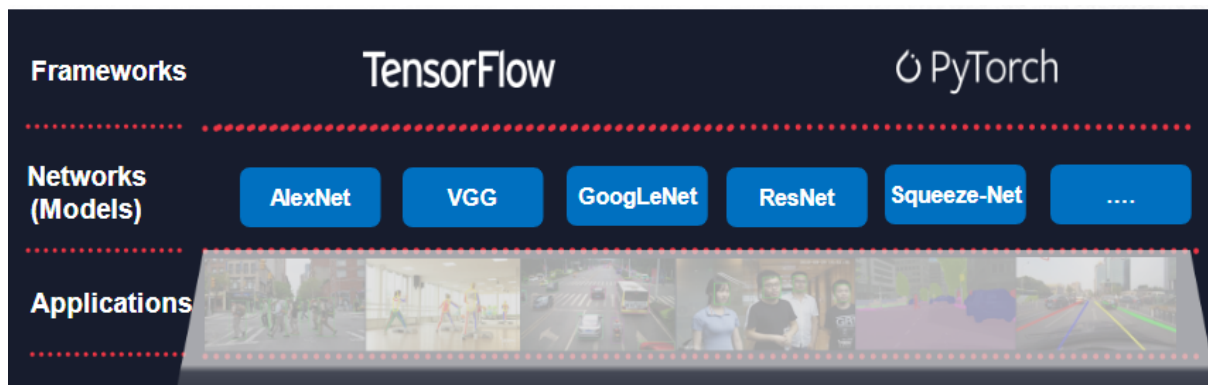


Frameworks - Vitis AI Development Environment

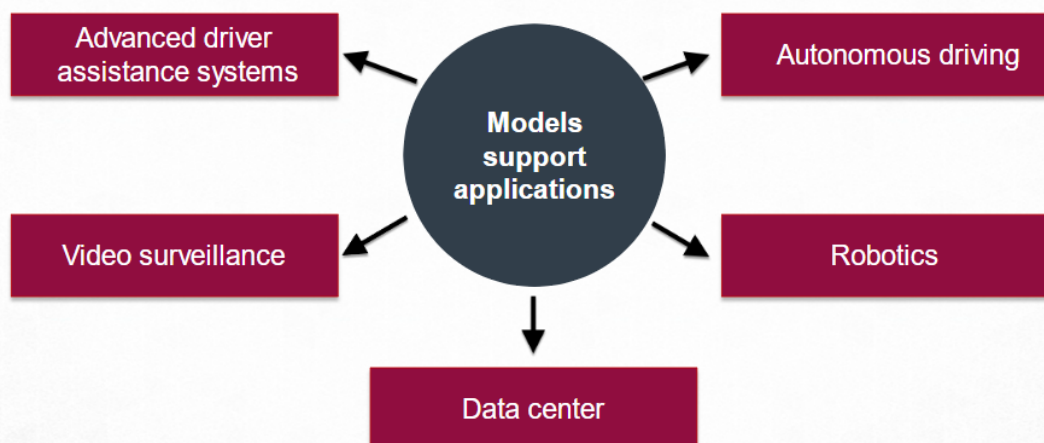


Frameworks and Models






Vitis AI Model Zoo

Vitis AI model zoo repository includes optimized deep learning models



- Pre-trained models for quick start
- Rich models from Tensorflow and PyTorch
- Advanced optimization, including PRUNING
- Retraining with the CUSTOM dataset

AI Model Zoo At a Glance

<div> <ul style="list-style-type: none"> ▪ Supports four mainstream ML frameworks ▪ Supports more applications <div>  TensorFlow  </div> <div>  TensorFlow 2.0 </div> </div>					
	Automotive	Smart Medical	Smart City	General Detection	General Classification
TensorFlow	Mobilenet_v2 – Segmentation	RefineDet – Endoscope Detection		ssdlite_mobilenet_v2, ssd_inception_v2, RefineDet	Inception v2, ResNet v2_50, ResNet v2_101, ResNet v2_152, MLPerf_ResNet50_v1.5, Mobilenet_edge_1.0, Mobilenet_edge_0.75
TensorFlow 2		2D Unet – Cell Segmentation	Semantic segmentation	ResNet 50, MobilenetV1, Inception v3	-
PyTorch	PointPillars – Point Cloud Detection SalsaNext – Point Cloud Segmentation	2D Unet – Chaos CT Liver Segmentation FPN R18 – Covid19 Segmentation	Person ReID ResNet 18		-

Vitis AI Zoo Model - Basic Information

Example

ResNet50: TensorFlow image classification

tf_resnetv1_50_imagenet_224_224_6.97G_<version>

Format for naming models:

F_M_(D)_H_W_(P)_C_V

F – Training Framework

- Tf – TensorFlow
- Dk – Darknet
- Pt – PyTorch

W – Width of the Input Data

M – Model Feature

P – Pruning Ratio

D – Dataset

C – Computation of the Model

H – Height of the Input Data

V – Version of the Vitis AI Environment

Vitis AI Model Zoo Benefits

- Quickly deploy with pre-validated models
- Measure the quantization accuracy
- Find the latest AI model resources

TensorFlow Model Directory Structure

code	# Contains test code which can run demo and evaluate model performance.
readme.md	# Contains the environment requirements, data preprocess and model information. Refer this to know that how to test the model with scripts.
data	# Contains the dataset that used for model test and training. When test or training scripts run successfully, dataset will be automatically placed in it.
quantized	
└ quantize_eval_model.pb	# Quantized model for evaluation.
float	
└ frozen.pb	# Float-point frozen model, the input to the `vai_q_tensorflow`. The pb name of different models may be different.

PyTorch Model Directory Structure

code	# Contains test and training code.
readme.md	# Contains the environment requirements, data preprocess and model information. Refer this to know that how to test and train the model with scripts.
data	# Contains the dataset that used for model test and training. When test or training scripts run successfully, dataset will be automatically placed in it.
qat	# Contains the QAT(Quantization Aware Training) results. The accuracy of QAT result is better than direct quantization. Some models but not all supported QAT, and only these models have qat folder.
quantized	
└ _int.pth	# Quantized model.
└ quant_info.json	# Quantization steps of tensors got. Please keep it for evaluation of quantized model.
└ _int.py	# Converted vai_q_pytorch format model.
└ _int.xmodel	# Deployed model. The name of different models may be different. For some models that support QAT you could find better quantization results in 'qat' folder.
float	
└ _int.pth	# Trained float-point model. The pth name of different models may be different. Path and model name in test scripts could be modified according to actual situation.

What is TensorFlow?

Features of TensorFlow

Easy model building to build and train models

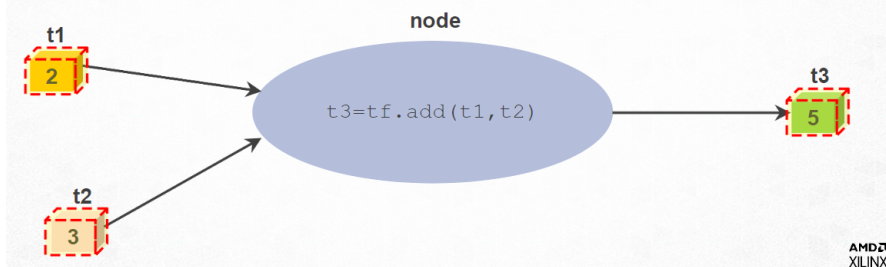
Widely used, open-source framework

Support for deployment to CPUs/GPUs

TensorBoard visualizes TensorFlow graphs and plots quantitative metrics

Everything in TensorFlow is based on the computation graphs

Graph is the actual model



The graph nodes represent the mathematical operations, while the graph edges (which connect the nodes) represent the multi-dimensional data arrays that flow between them.

There are four properties of the **tensors**:

- The name of the tensor
- The number of dimensions, or rank
- The size of the dimensions, which define the shape
- The data type of the element, which is the same for all elements within the tensor

Example:

```
# make some tensors
t1 = tf.constant([[1,2],
                  [3,4],
                  [5,6]])
```

t1 shape: [3, 2]

t1 size: 6

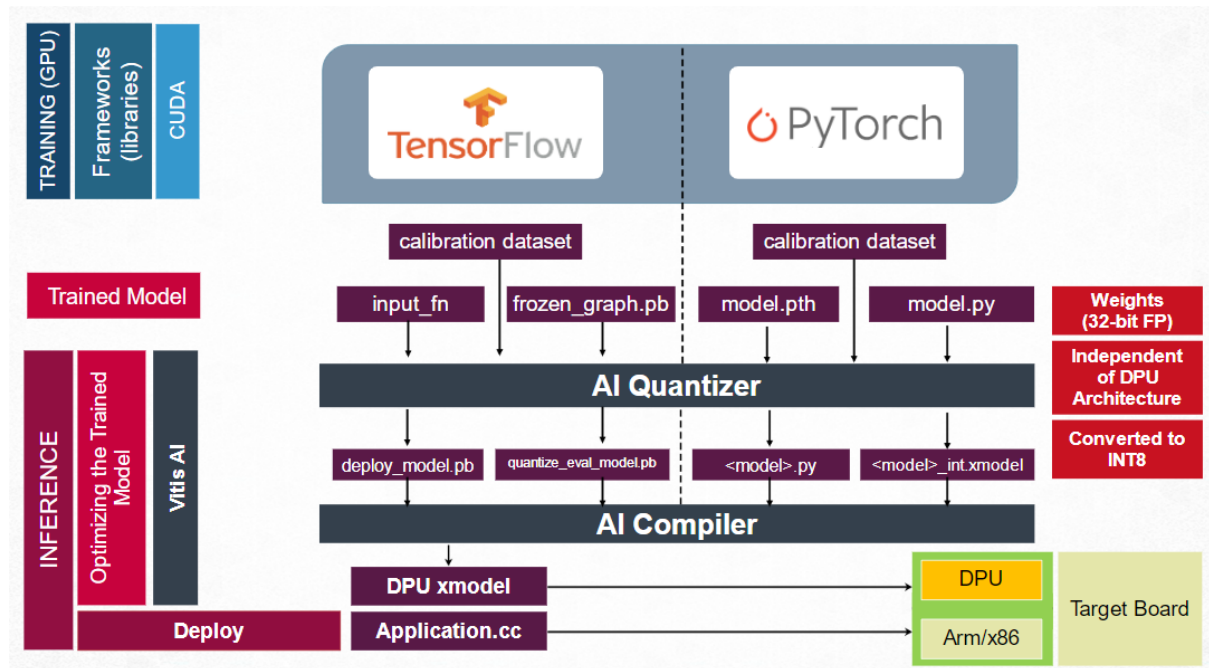
t1 rank: 2

What is PyTorch?

PyTorch

- Based on Torch library
- Developed by Facebook's AI Research lab
- Python-based scientific computing package
- NumPy replacement uses GPU
- Provides maximum flexibility and speed

Application Development Flow Using Supported Frameworks



For TensorFlow framework, Vitis AI quantizer takes a floating point frozen graph, a calibration data set and the Python function `input_fn` to read images in the calibration data set and performs a preprocess on it. As an output, the tool generates a quantized model for the Vitis AI compiler and a quantized model for evaluation, which are in an extended Tensorflow format; that is PB format as an output.

For a pytorch framework, the Vitis AI quantizer, it takes a pre-trained PyTorch model, generally a PTH file, a Python script (including a float model definition), and a subset of the training dataset containing 100 to 1000 images. Two important files will be generated: `<model>.py` which is the converted PyTorch format model, and `<model>_int.xmodel`, which is the deployed model.

The Vitis AI compiler compiles the models into one or more DPU kernels, which are xmodel files containing DPU instructions.

The Vitis AI development environment offers a unified set of high levels C++/Python programming API's to smooth machine learning application development across our cloud to edge devices. These enable you to easily port deployed DPU applications from Cloud to Edge or vice versa.