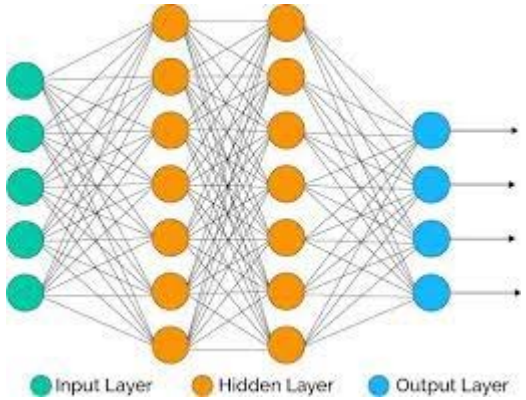
赛灵思技术日
XILINX TECHNOLOGY DAY

用賽靈思 FPGA 加速機器學習推斷

張帆
資深全球AI方案技術專家
2019年4月25日

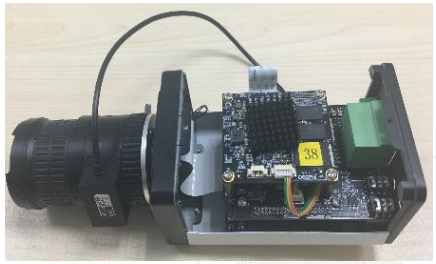
XILINX

AI在edge端應用中落地面臨的挑戰



= **43 TOPS**

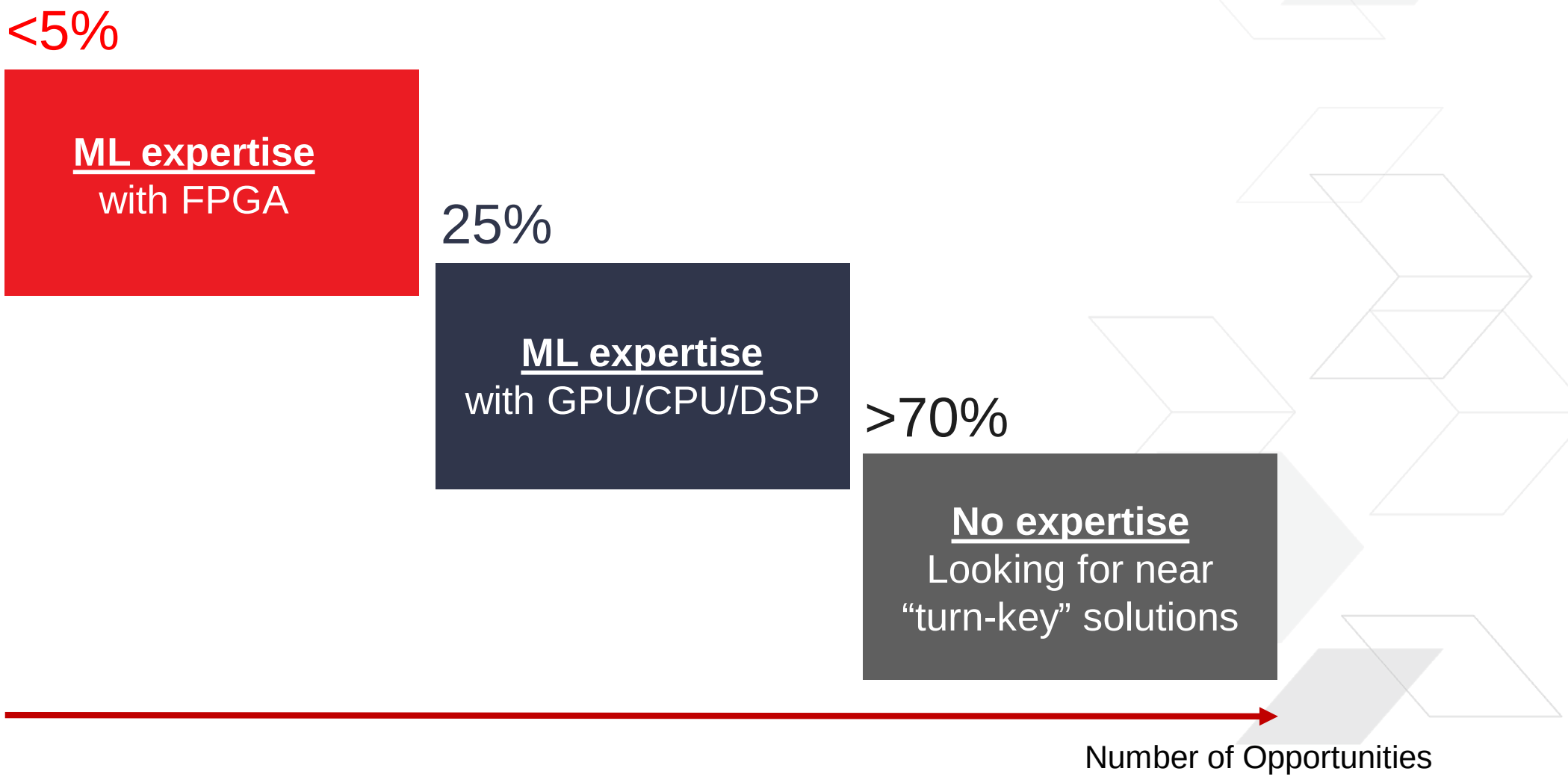
?



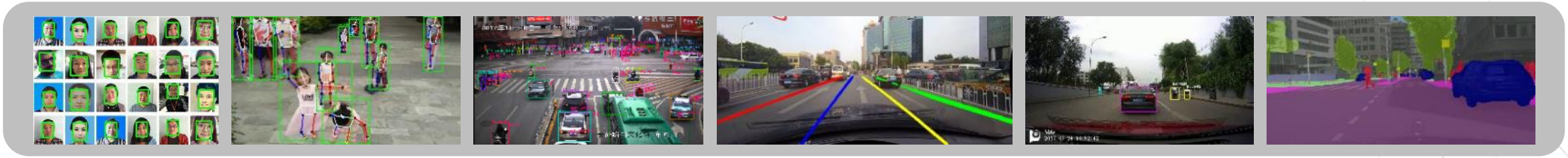
1080p Object Detection (SSD) @ 30 FPS

< 10W, < 50 ms latency, <$50

Challenges Are Opportunities



Xilinx AI Solution

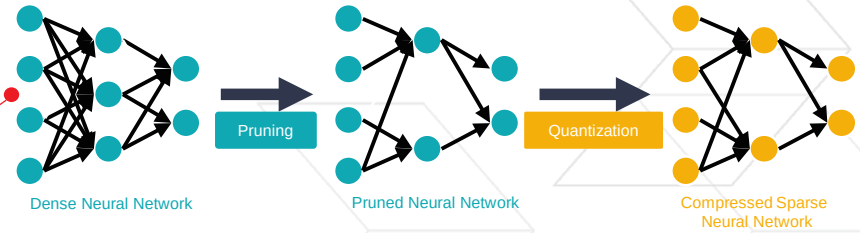


Algorithm Model Zoo

DNNDK

Xilinx AI SDK

DPU and Accelerating IPs



Surveillance | **Data center** | **ADAS/AD**

Reference solutions

Deep learning libraries

Cross-compiling tools

BSP for boards

Compilation

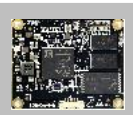
Compiler | Assembler

Runtime

Core API | Loader

Driver | Tracer


Zynq-7020


Zynq-7020


ZU2


ZU2

ZU9

Algorithm Model Zoo

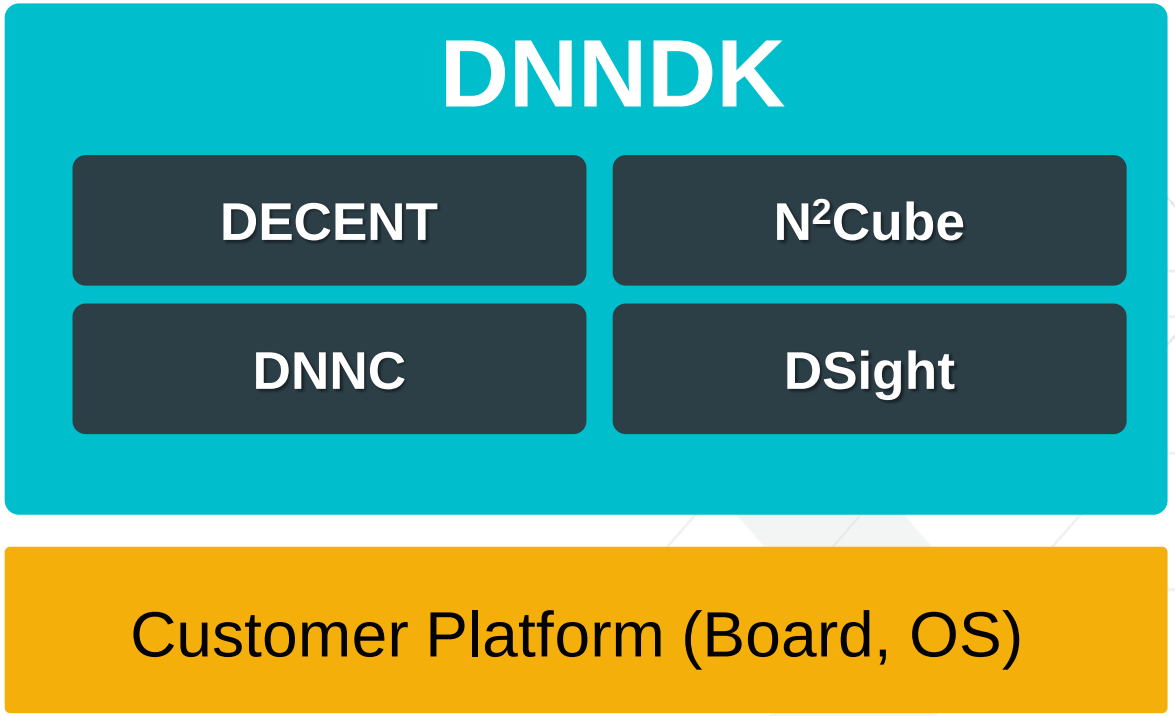
69 Models in the zoo, in which 34 models are already deployed.

Network	Backbone	DPU Deployment	Input Size	OPs	Paras	Training Set	Val Set	Eva Norm Float	Eva Norm Fixed
Resnet50_v1	Resnet	Yes	224*224	7.7G	25.6M	ImageNet	ImageNet	0.7483	0.7338
Inception_v3	Inception	No	299*299	11.43G	23.8M	ImageNet	ImageNet	0.7401	0.7347
SSD	VGG16	No	300*300	62.77G	26.3M	Voc07+12	Voc07	77.19%	
RefineDet	VGG-16	Support	480*360	123.9G	29.6M	Coco2014	Coco2014	70.14%	
Densebox		Yes	320*320	492M		Private	Private	97.92%	97.50%
Yolo_v3	Yolo	Yes	512*288	53.7G	61.8M	Cityscape	Cityscape	53.7%	53.1%

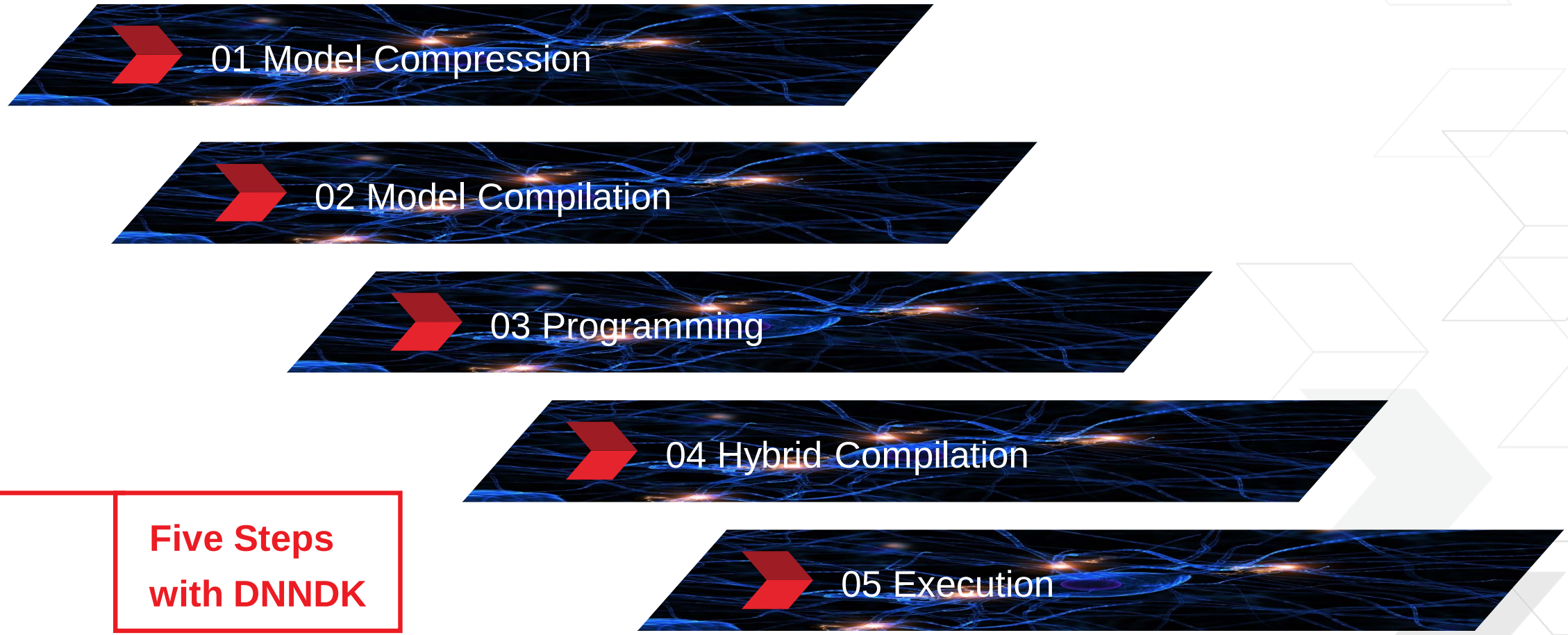
- DPU deployment
  - **Yes:** the model is successfully deployed on DPU.
  - **Support:** the model is supported but not deployed. Similar model structure is deployed and test successfully.
  - **No:** the model is not supported by DPU right now mainly due to some special operations or layers.

DNNDK – Deep Neural Network Development Kit

- DECENT
  - Flt32 to Int8 quantization with one line command
- DNNC
  - Automatic layer fusion to avoid frequently data read and write
- Runtime N^2Cube
  - Various APIs to facilitate specific application
- Profiler Dsight
  - Powerful tool as failure analysis and optimization



DNNDK Dev Flow



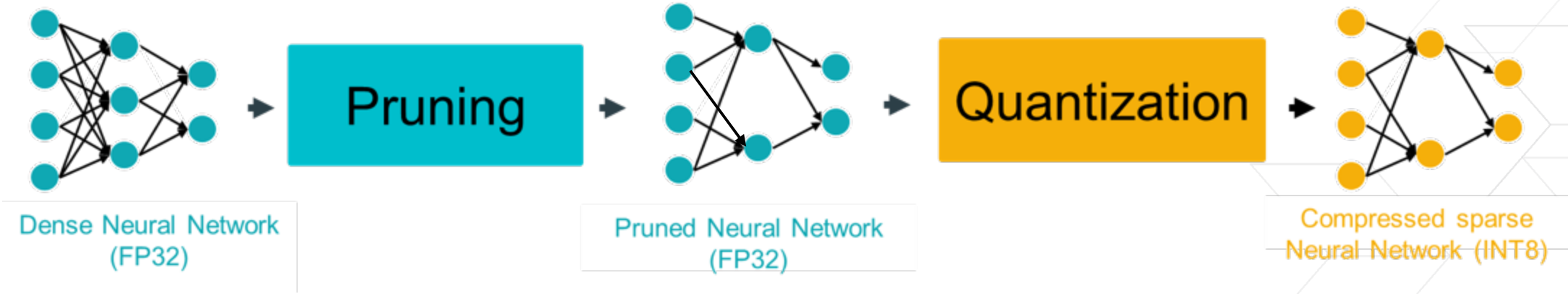
Five Steps with DNNDK

Step 1: Model Compression With DECENT

decent – Deep compression Tool

decent_q – Quantization Tool

decent_p – Pruning Tool

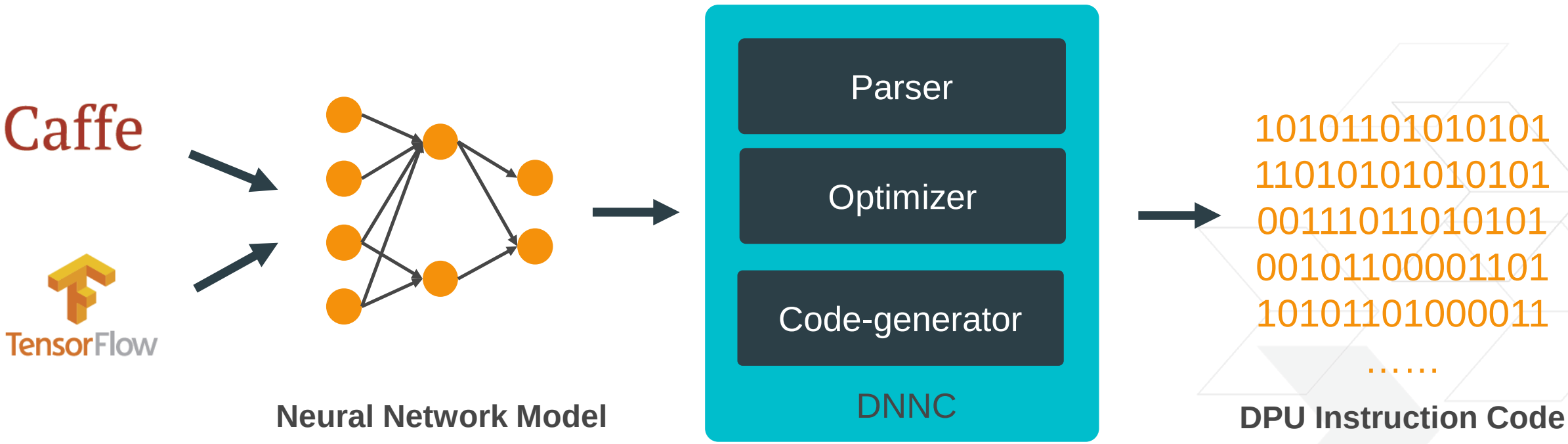


- Consists of two separate tools
  - Quantization Tool
  - Pruning Tool

- Effects
  - Compress model size 5x – 100x
  - Compress running time 1.5x – 10x

- Platform
  - Caffe, Darknet
  - TensorFlow
    - Quantization Tool Beta version
    - Pruning Tool Internal version

Step 2: Model Compilation With DNNC

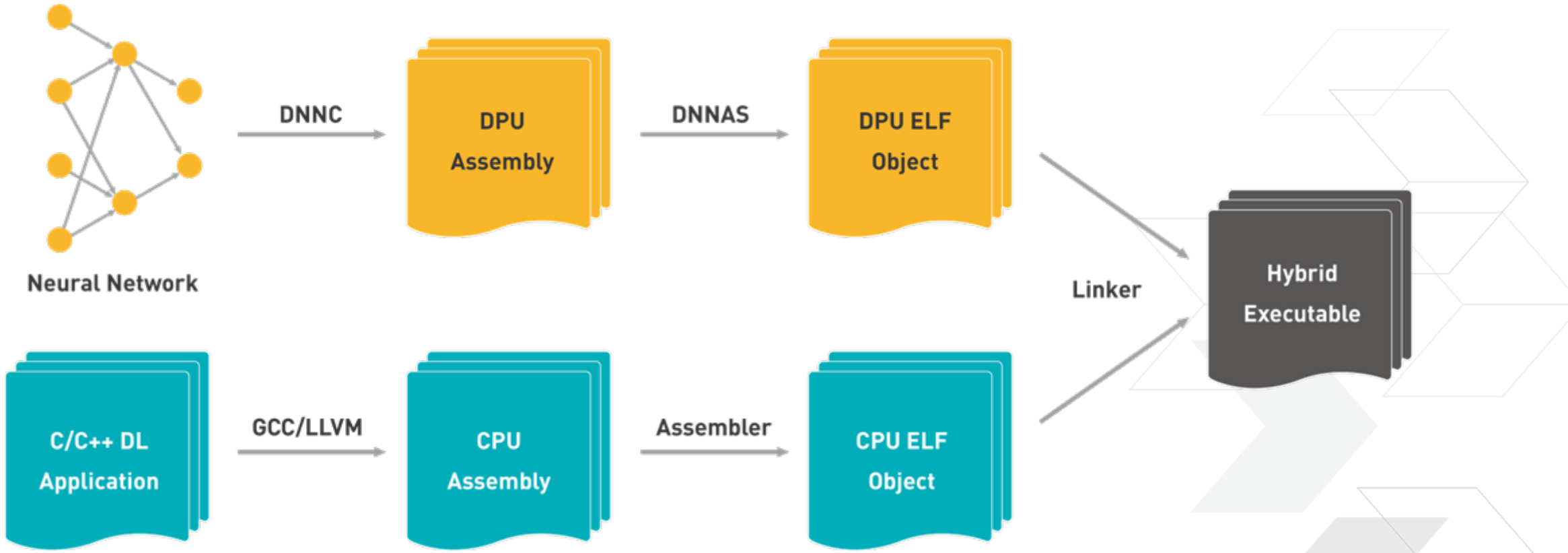


Programmable tensor-level DPU instruction set
- Compatible for Caffe/TensorFlow frameworks
- Flexible & scalable for various CNN layers

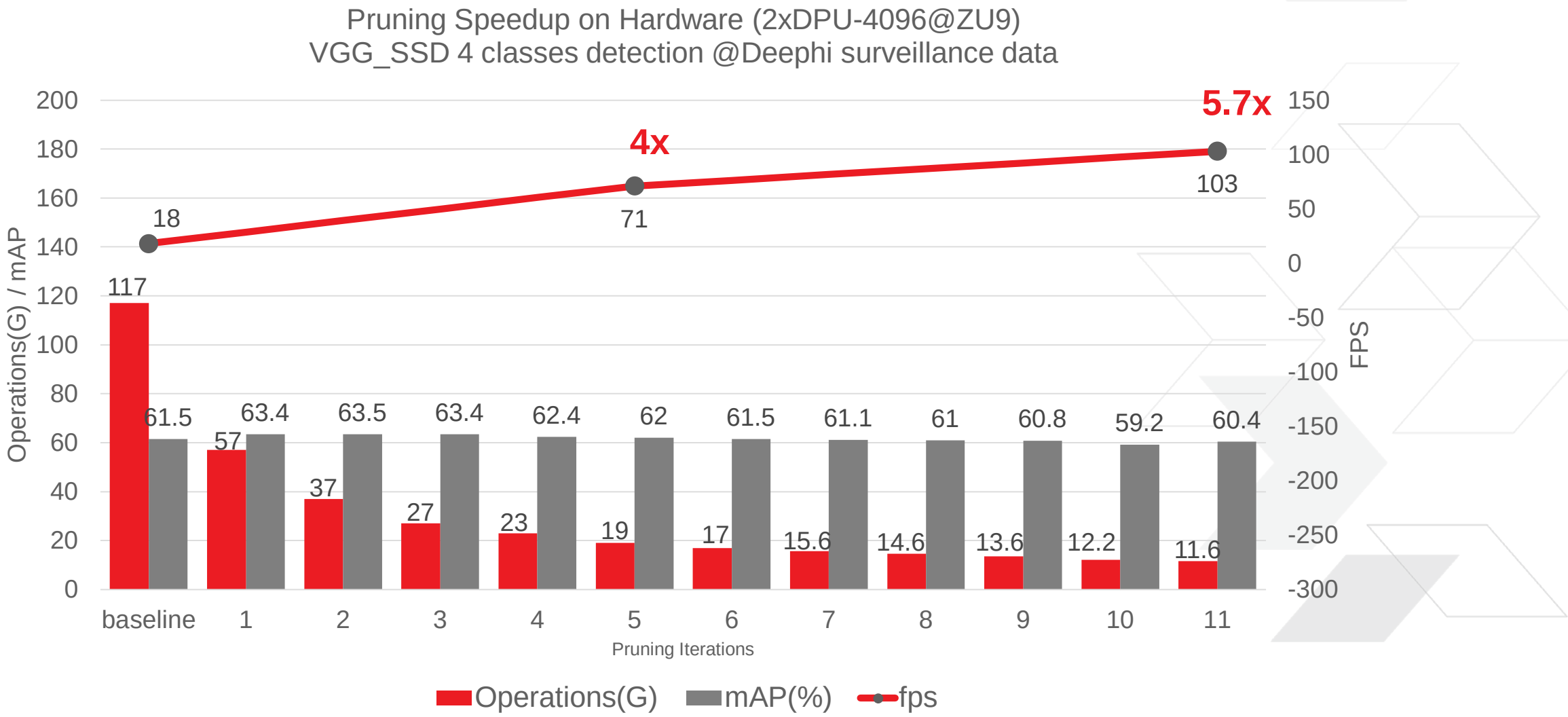
Step 3: Programming with DNNDK API

```c
int main(int argc, char *argv[])
{
    DPUKernel *kernel_conv;
    DPUKernel *kernel_fc;
    DPUTask   *task_conv;
    DPUTask   *task_fc;
    char      *input_addr;
    char      *output_addr;

    /* DNNDK API to attach to DPU driver */
    dpuInit();

    /* DNNDK API to create DPU kernels for CONV & FC networks */
    kernel_conv = dpuLoadKernel("resnet50_conv", 224, 224);
    kernel_fc = dpuLoadKernel("resnet50_fc", 1, 1);

    /* Create tasks from CONV & FC kernels */
    task_conv = dpuCreateTask(kernel_conv);
    task_fc = dpuCreateTask(kernel_fc);

    /* Set input tensor for CONV task and run */
    input_addr = dpuGetTensorAddress(dpuGetTaskInputTensor(task_conv));
    setInputImage(Mat &image, input_addr);
    dpuRunTask(task_conv);
    output_addr = dpuGetTensorAddress(dpuGetTaskOutputTensor(task_conv));

    /* Run average pooling layer on CPU */
    run_average_pooling(output_addr);

    /* Set input tensor for FC task and run */
    input_addr = dpuGetTensorAddress(dpuGetTaskInputTensor(task_fc));
    setFCInputData(task_fc, input_addr);
    dpuRunTask(task_fc);
    output_addr = dpuGetTensorAddress(dpuGetTaskOutputTensor(task_fc));

    /* Diaplay the Classification result from FC task */
    displayClassificationResult(output_addr);

    /* DNNDK API to destroy DPU tasks/kernels */
    dpuDestroyTask(task_conv);
    dpuDestroyTask(task_fc);

    dpuDestroyKernel(kernel_conv);
    dpuDestroyKernel(kernel_fc);

    /* DNNDK API to dettach from DPU driver and free DPU resources */
    dpuFini();

    return 0;
}
```

Step 4: DNNDK Hybrid Compilation Model

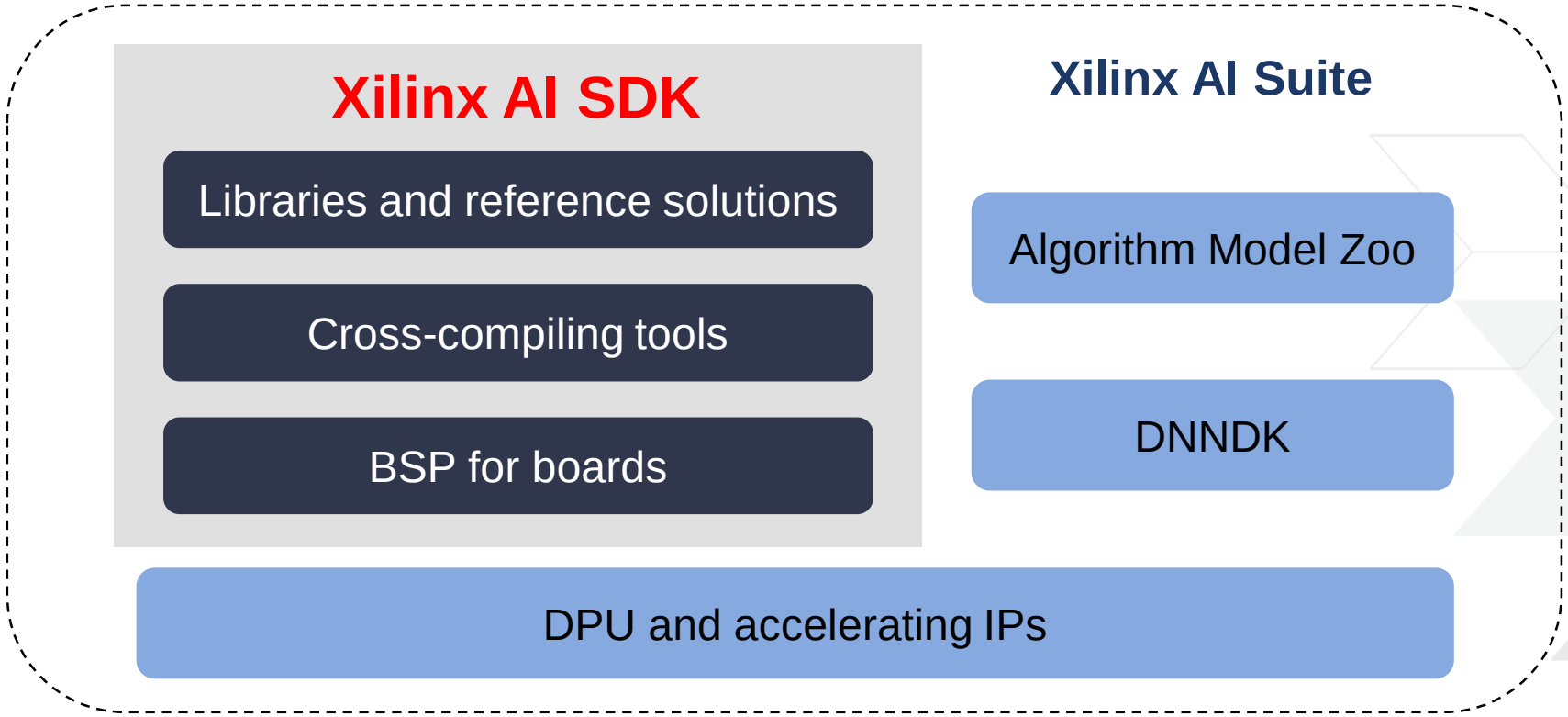


Acceleration on Hardware - SSD

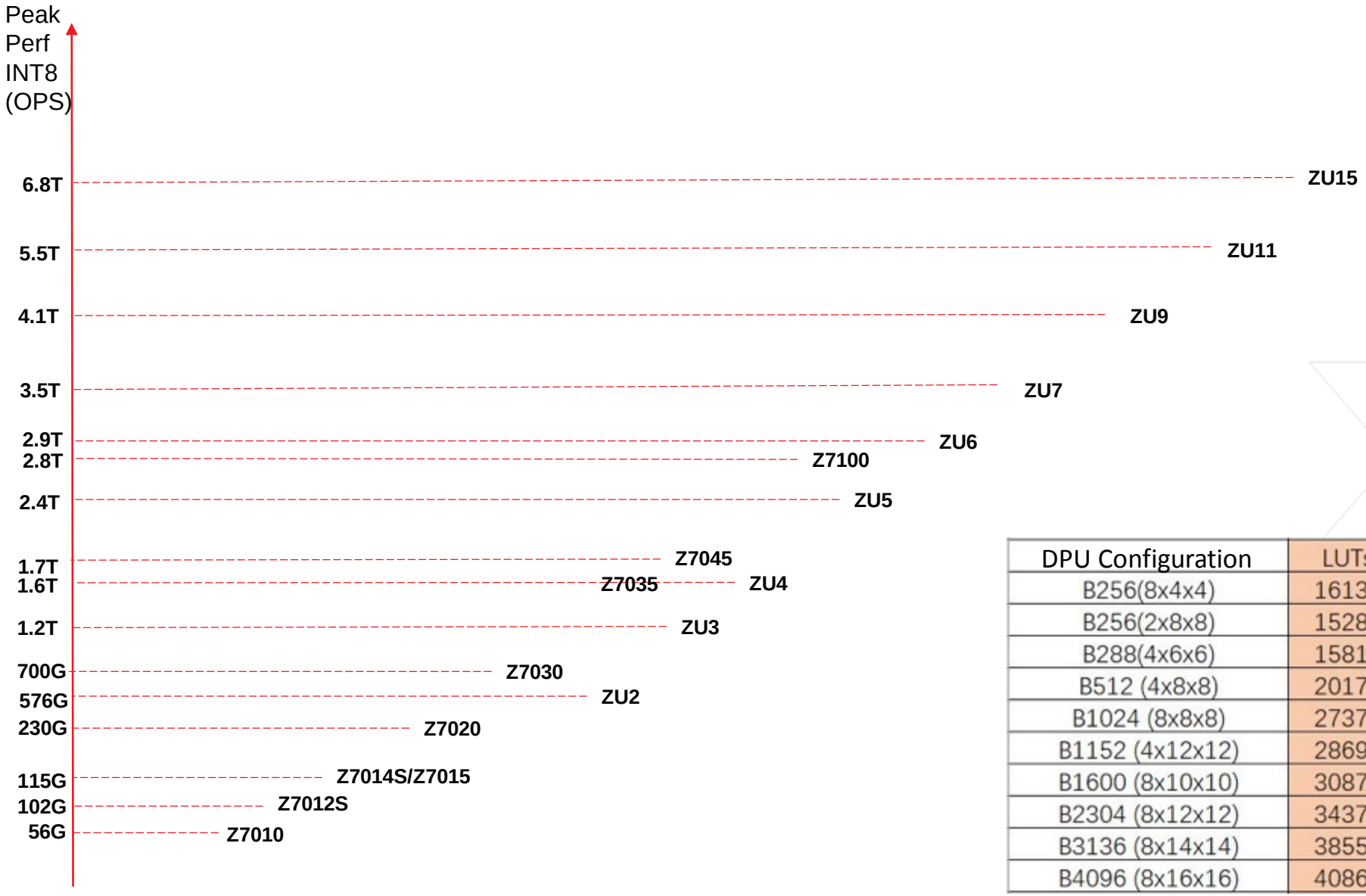


Xilinx AI SDK

- **Xilinx AI SDK to enable low touch engagements for customers**
  - Encourage top customers to use SDK to build applications and solutions
  - Only use generic SDK release to support low priority customers



DPU Scalability

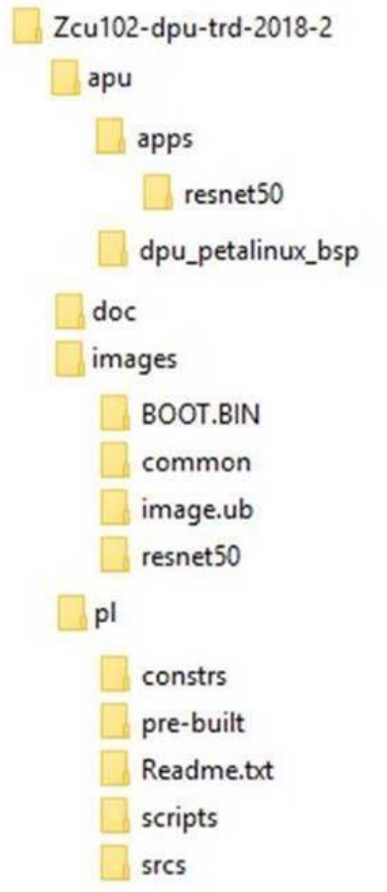


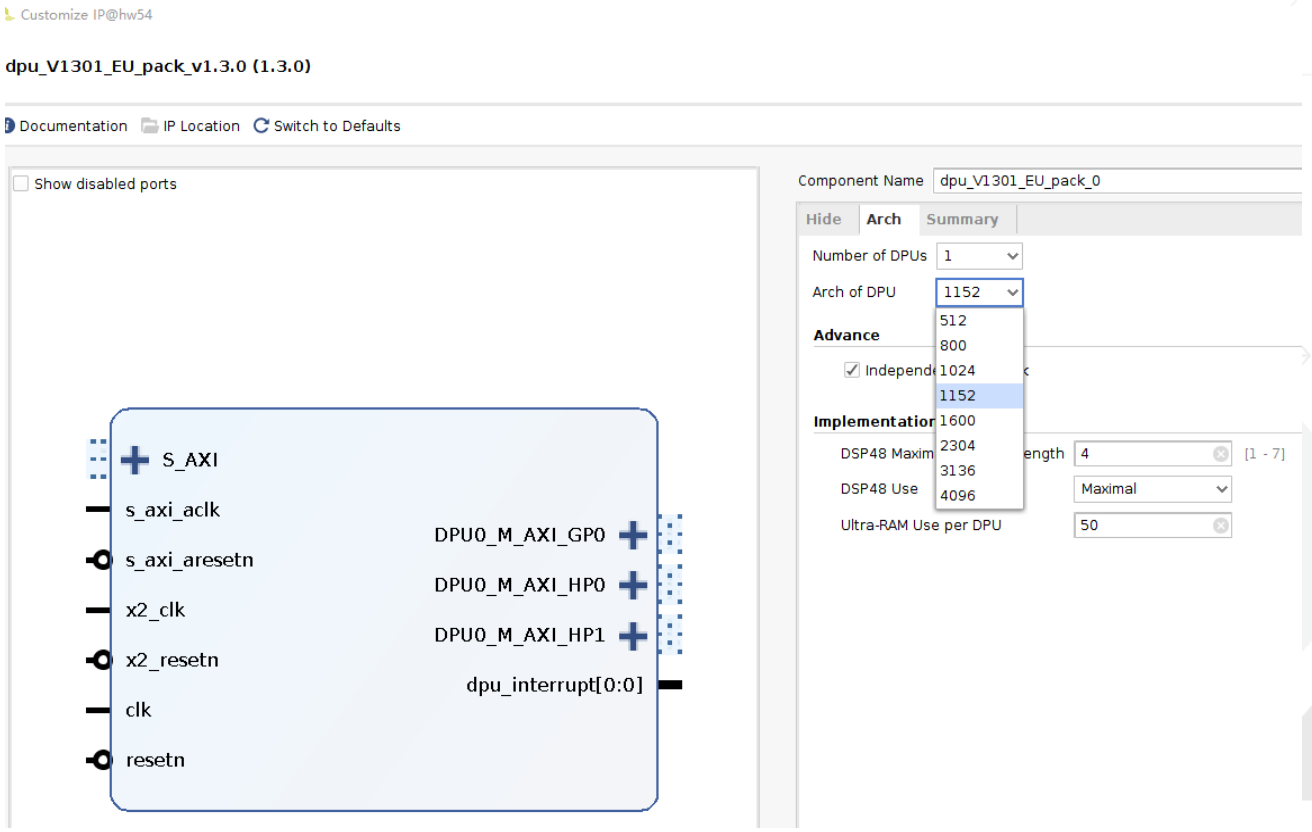
DPU Configuration	LUTs	Registers	BRAM	DSP
B256(8x4x4)	16132	25064	43	66
B256(2x8x8)	15286	22624	53.5	50
B288(4x6x6)	15812	23689	46	62
B512 (4x8x8)	20177	31782	69.5	98
B1024 (8x8x8)	27377	46241	101.5	194
B1152 (4x12x12)	28698	46906	117.5	194
B1600 (8x10x10)	30877	56267	123	282
B2304 (8x12x12)	34379	67481	161.5	386
B3136 (8x14x14)	38555	79867	203.5	506
B4096 (8x16x16)	40865	92630	249.5	642

* B256/288/512/3136 work in progress

DPU IP Integration

The DPU TRD (Targeted Reference Design) has been published in Xilinx.com.
https://www.xilinx.com/products/design-tools/ai-inference/ai-developer-hub.html#edge





Key Takeaway

1. Xilinx offers advanced Edge ML solution
2. DNNDK makes neural network deployment easily on FPGA
3. DPU IP is ready to download from Xilinx website for free

Adaptable.
Intelligent.

赛灵思技术日
XILINX TECHNOLOGY DAY

XILINX