arm AI

Virtual Tech Talk

Faster time-to-production for Computer Vision AI on Arm-powered Edge Devices

Deeplite Inc.

Davis Sawyer, CPO

Charles Marsh, CCO

Feb 8th, 2022

arm

Welcome!

Tweet us: @ArmSoftwareDev -> #AIVTT

Check out our Arm Software Developers YouTube channel

Signup now for our next AI Virtual Tech Talk: developer.arm.com/techtalks

Our upcoming Arm AI Tech Talks

Date	Title	Host
February 22nd , 2022	*** New Arm ML Quarterly Research Special *** Federated Learning Based on Dynamic Regularization to Debias Model Updates	Arm ML Research

Visit: developer.arm.com/techtalks

Presenters

Davis Sawyer

Davis is the Co-founder and CPO at Deeplite, where he leads product development and strategy. Recognized as a C2 Montreal Emerging Entrepreneur, he also serves as chairperson for tinyML Montreal and Toronto.

Charles Marsh

Charles is the CCO at Deeplite, leading sales and business development. He specializes in business growth for high-tech start-up companies, with two successful exits to-date.

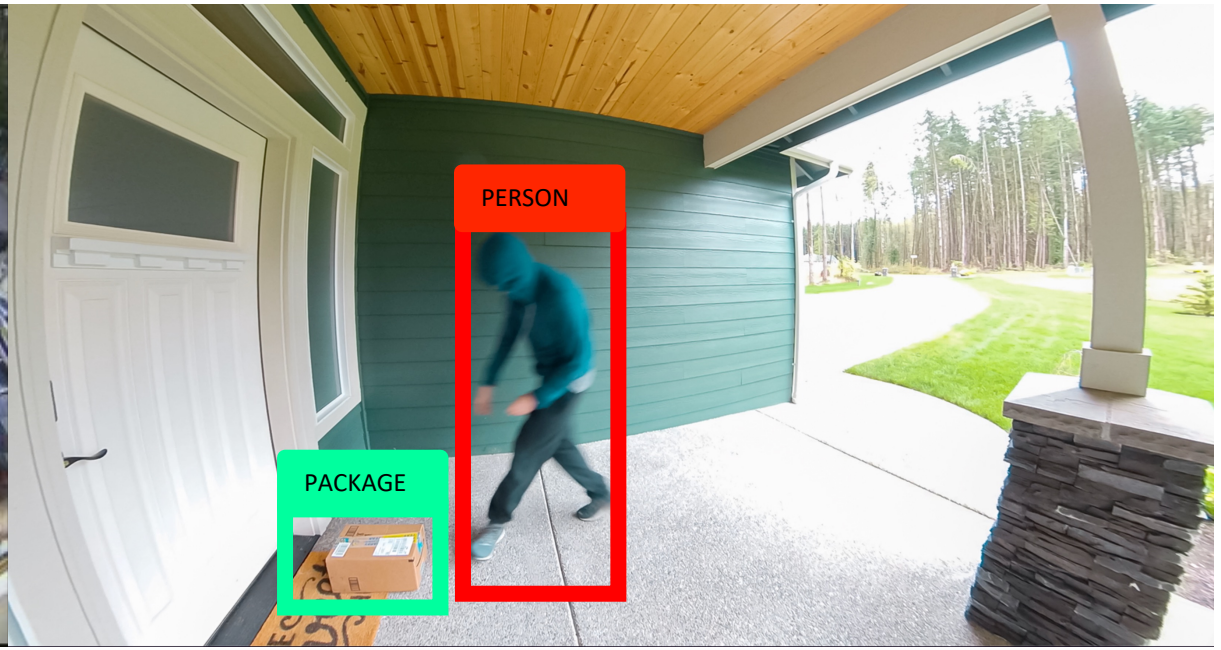
What we've learned as an industry:

Productizing computer vision AI at the edge is *hard*

Latest wave of accurate DNNs for Computer Vision

Making real world vision-powered products possible

Intelligent Video Analytics



Smart Doorbell Cameras

arm

Latest wave of accurate DNNs for Computer Vision

How do we go from research

NVIDIA Tesla V100 GPU

250 Watt

$2250-8000 USD

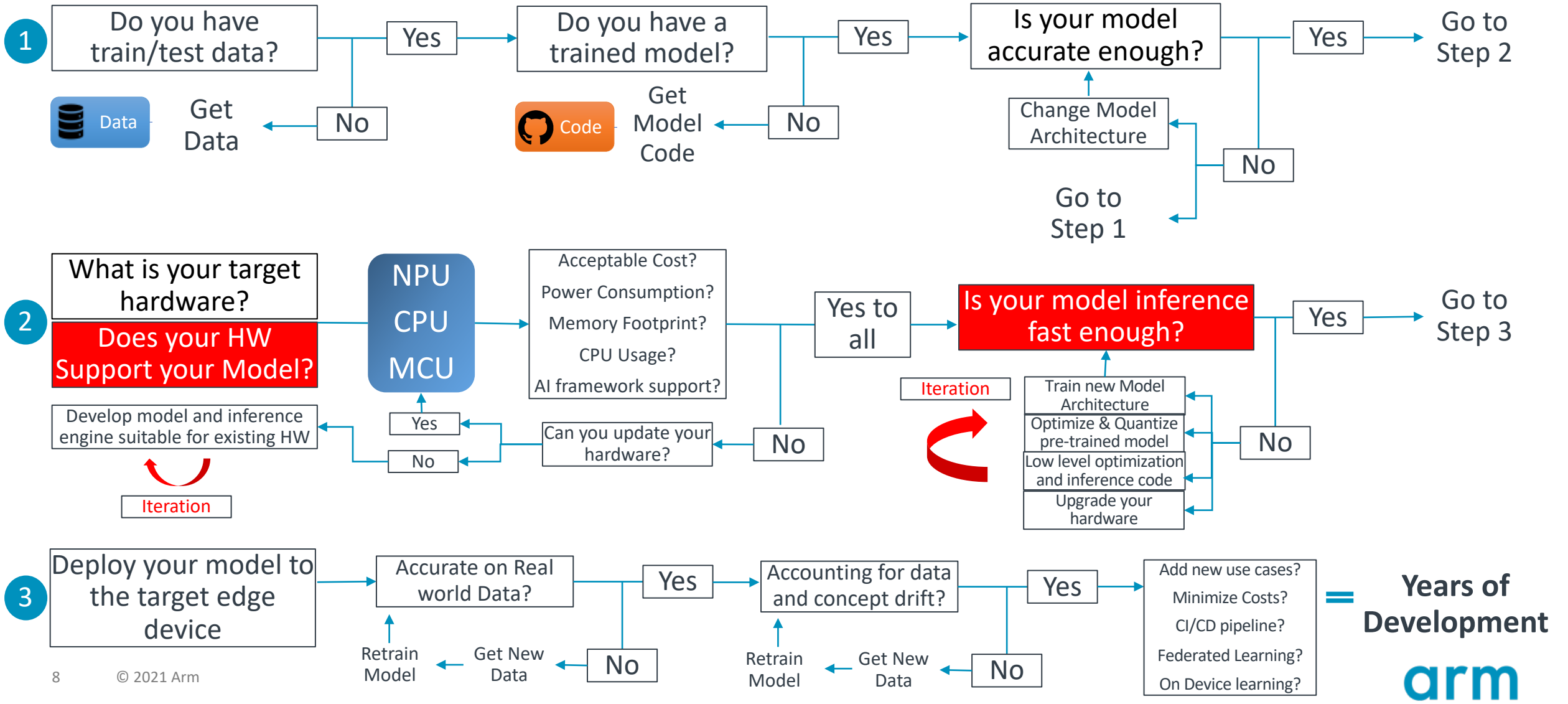
To the real-world:

Quadcore Arm Cortex-A53 CPU

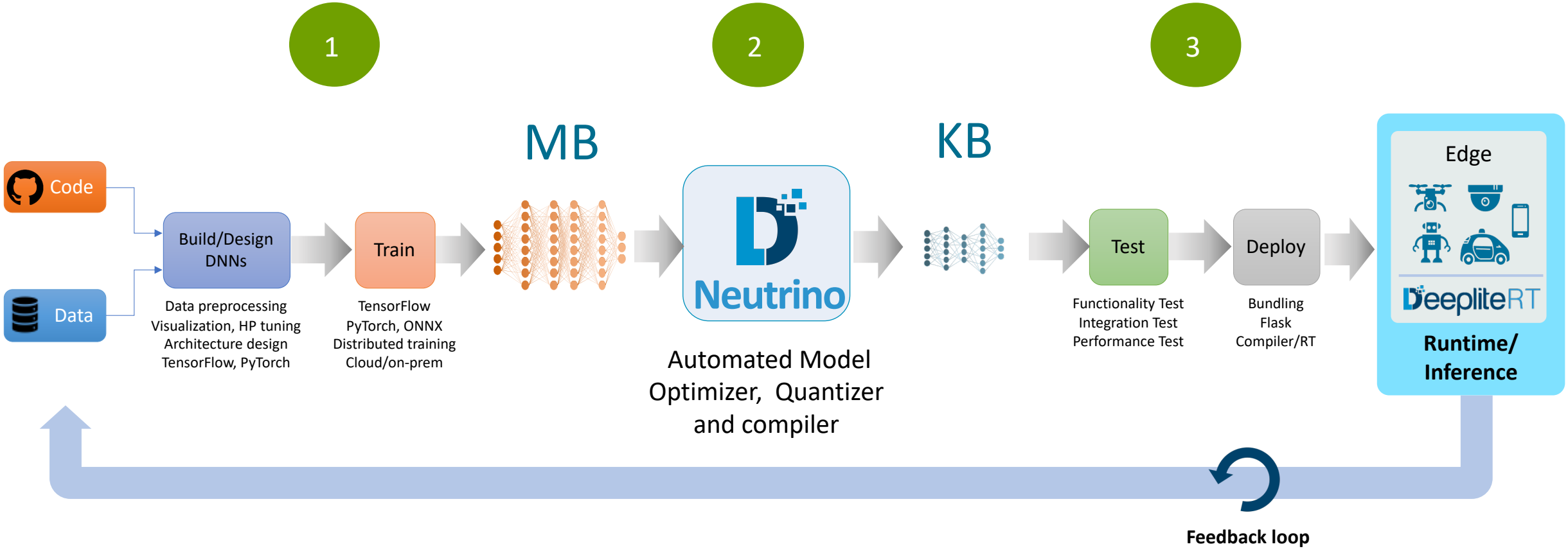
200mW

$30-35 USD (RPi3B)

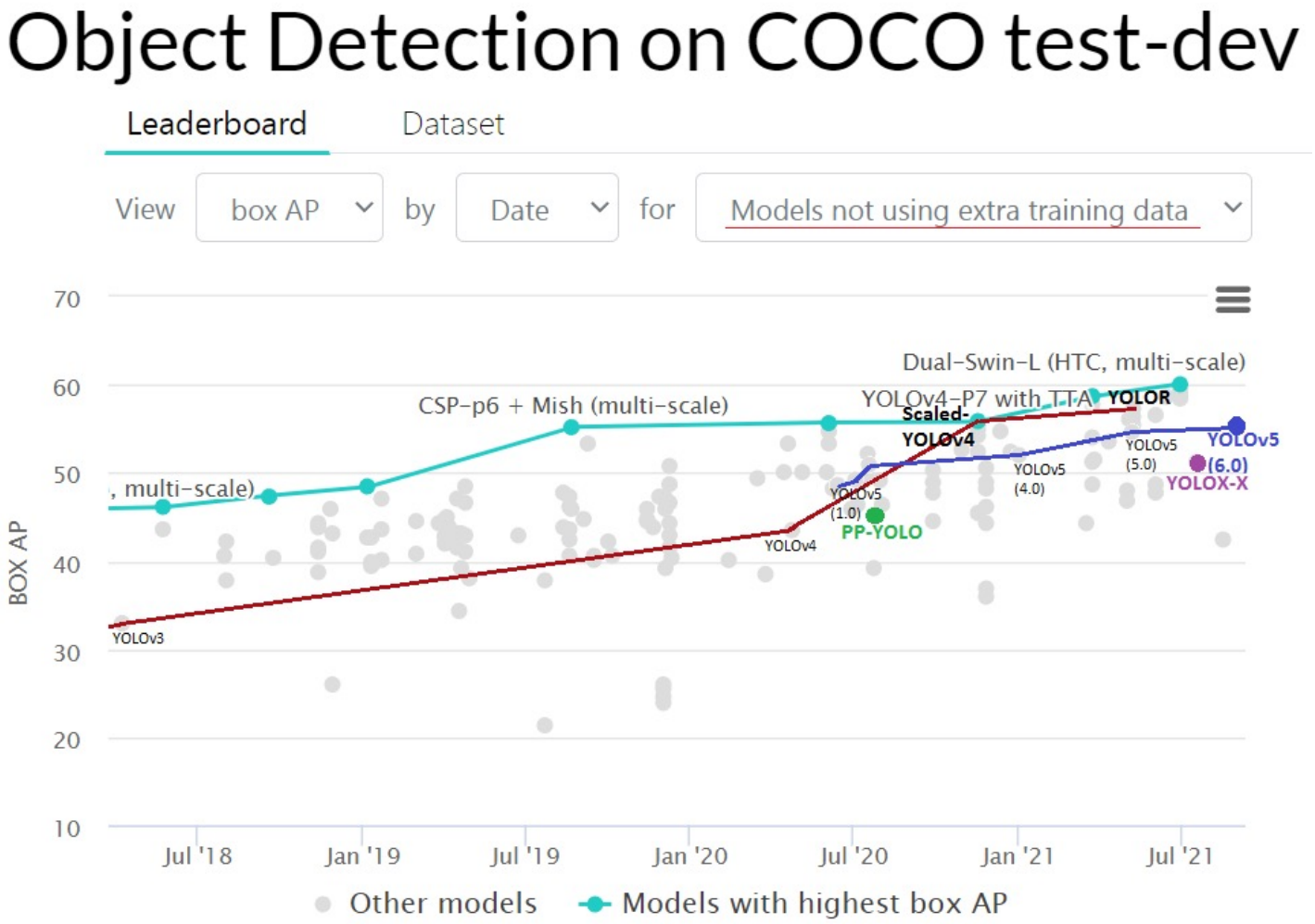
Simplified process for an edge computer vision AI use case



Really simplified process for an edge computer vision AI use case



Best Practices for Model Training / Development (Step 1)



- Surge in last few years in accurate, off-the-shelf deep neural networks for vision related tasks
- Model Size/Inference Speed tradeoffs to be made when moving from training to inference hardware

arm

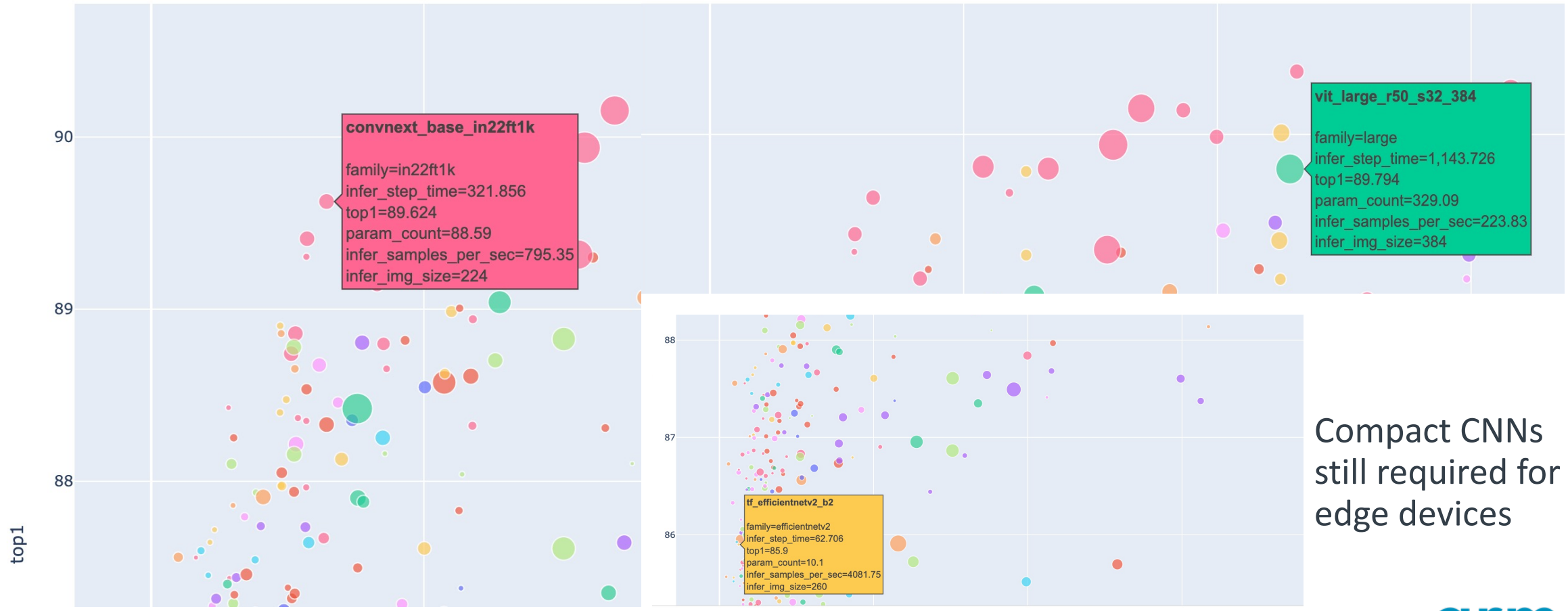
Best Practices for Model Training / Development (Step 1)

Pytorch Image Models Results Visualization.ipynb ☆

File Edit View Insert Runtime Tools Help Changes will not be saved

Code + Text Copy to Drive

Vision Transformers (ViTs) are gaining popularity but **massive compared to CNNs (329Mparams vs 10M)**

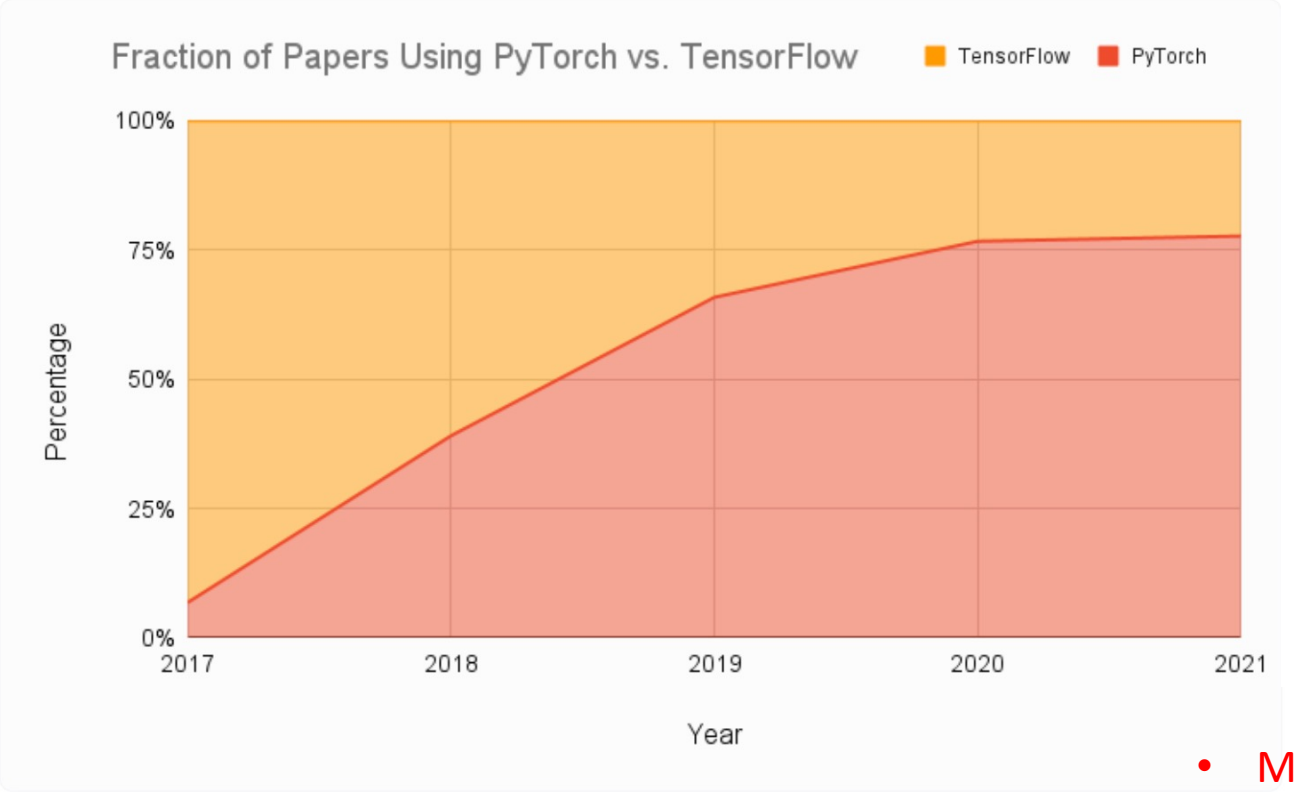


Compact CNNs still required for edge devices

data source

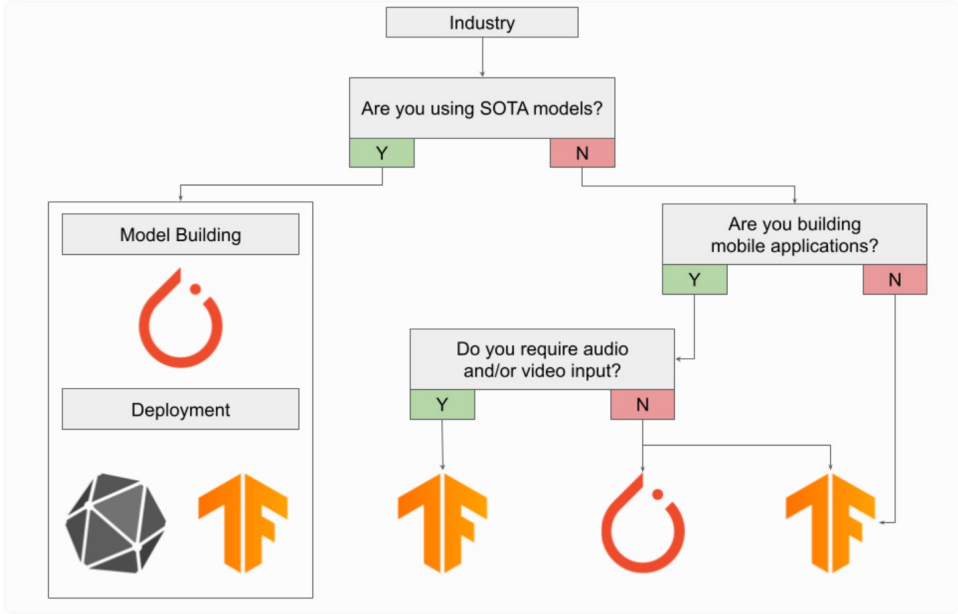
Best Practices for Model Training / Development (Step 1)

AI Frameworks



[data source]

What if I'm in Industry?



- Many CPUs and MCUs don't support PyTorch/ONNX for inference
- SOTA Vision models are increasingly available in PyTorch

Best Practices for Inference Optimization (Step 2)

CV Model Architecture Optimization

TensorFlow

PyTorch

Built-in Capabilities

- Weight Pruning
- Pruning+XNN Pack
- NAS

3rd Party Tools

- Open-Source Distillation & Compression
- SW Accelerators

Application	Initial Architecture	Deeplite Neutrino Compression (FP32)			Accuracy Change (%)	Dataset
		Original Size	Optimized Size	Improvement		
Image classification	VGG19	80MB	2.16MB	x37	<1.00% (Top1)	CIFAR100
	Mobilenet-v1.0	12.8MB	530KB	x22	~1.50% (Top1)	Visual Wake Words
	ResNet18	45MB	1.8MB	x27	<1.00% (Top1)	Subset of ImageNet
	ResNet50	97MB	26MB	x3.72	~1.50% (Top1)	ImageNet
Object Detection	ResNet50-SSD300	54MB	18MB	x3	~0.00 (mAP)	Subset of COCO2017
	ResNet34-SSD300	32MB	11MB	x3.75	~0.00 (mAP)	VOC 2012
	Yolov3	235MB	28MB	x8	<0.02 (mAP)	
Segmentation	Unet-ResNet18	83.34MB	34.7MB	x2.4	~0.00 (mIoU)	
	Deeplab-Mobilenetv2	29.1MB	13.3MB	x2.2	<0.02 (mIoU)	
	U-Net	65.9MB	1.04MB	x63.5	<0.01 (mIoU)	Carvana

arm

Best Practices for Inference Optimization (Step 2)

Low precision quantization for efficient inference

TensorFlow

Post-Training Quantization (PTQ)

FP16, INT8

Quantization-Aware Training (QAT)

INT8

PyTorch

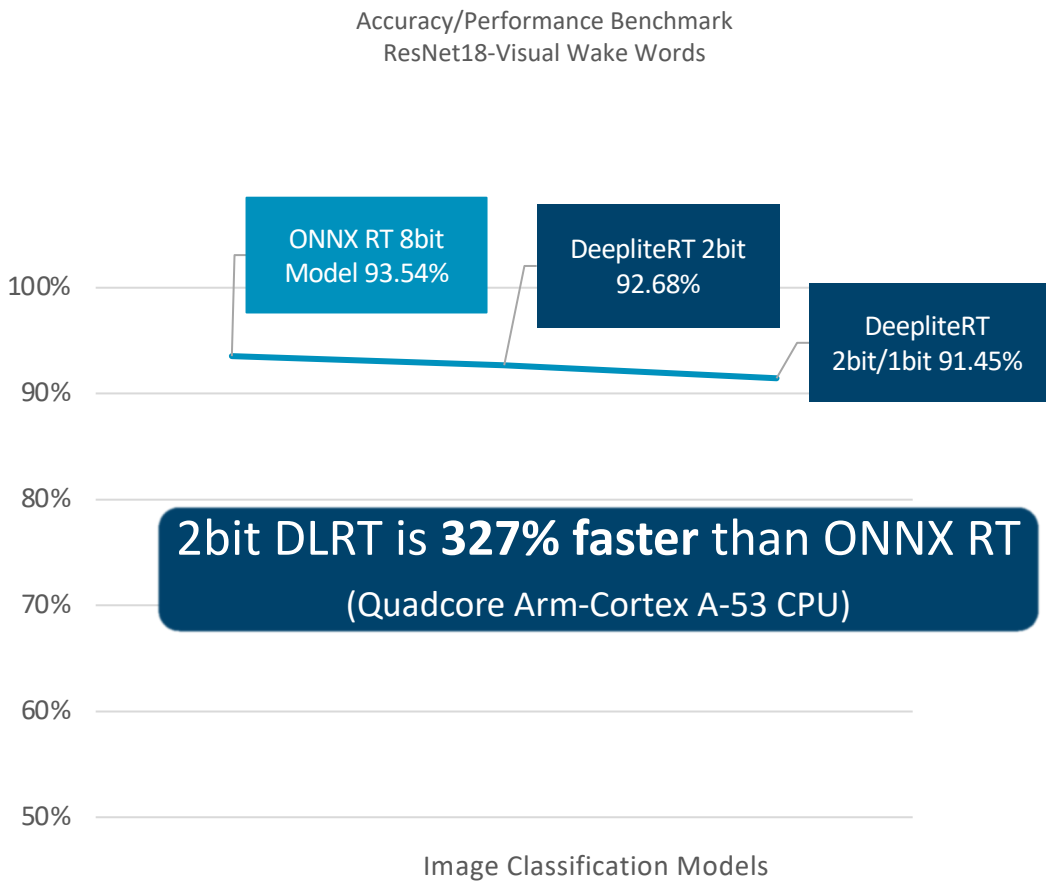
Post-Training Quantization (PTQ)

FP16, INT8

Quantization-Aware Training (QAT)

INT8, **2bit, 1bit**

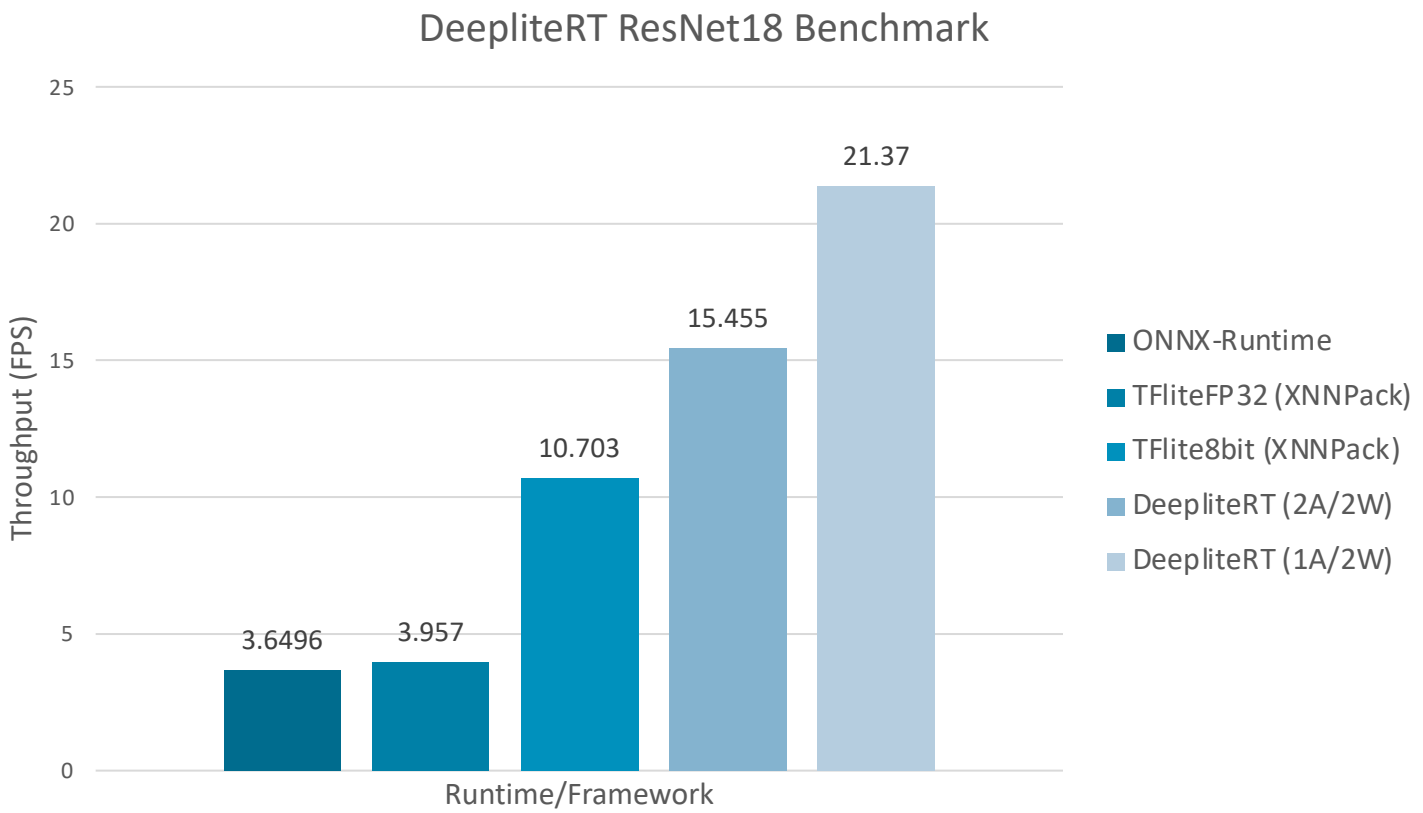
DeepliteRT



arm

Best Practices for Streamlined Edge Deployment (Step 3)

- Approaching GPU inference speed on Cortex-A CPU

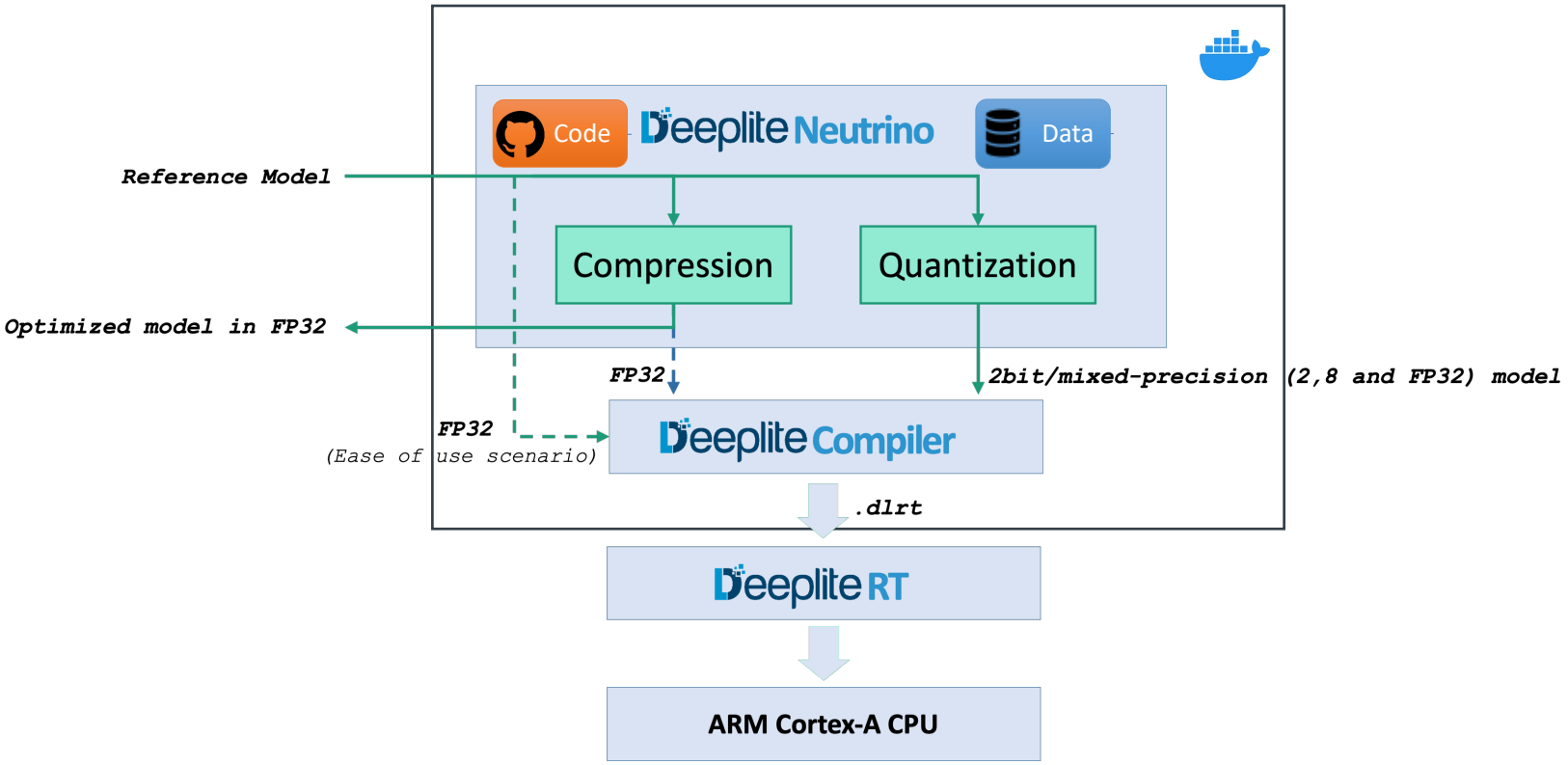


Visual Wake Up + Detection on Arm Cortex-A CPU

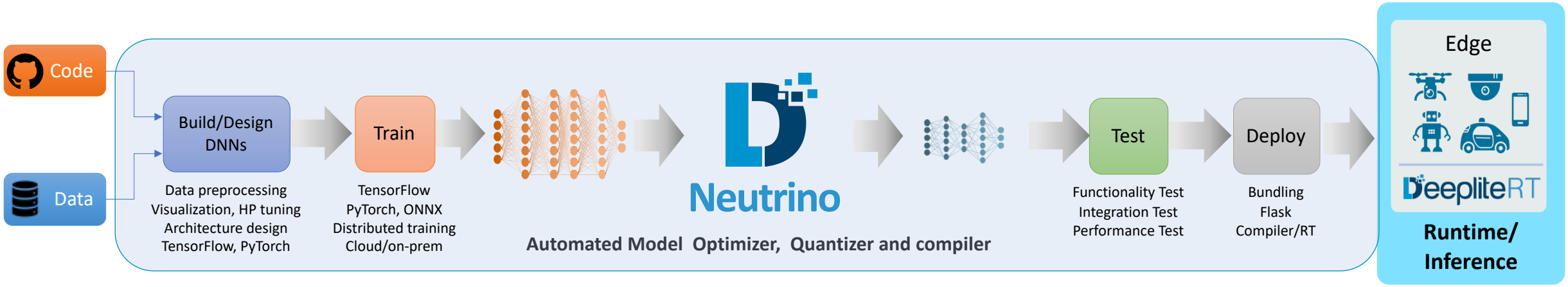
2bit YOLOv5 Running on Raspberry Pi4 (Arm Cortex-A72 @ 1.5GHz Quad-core CPU)

Best Practices for Streamlined Edge Deployment (Step 3)

- Save on memory, cost and power for inference
- Save on development effort with docker & pip install for deployment



Putting it all Together in the Edge AI Workflow



arm

Person Detection on low-power Arm CPU

Problem

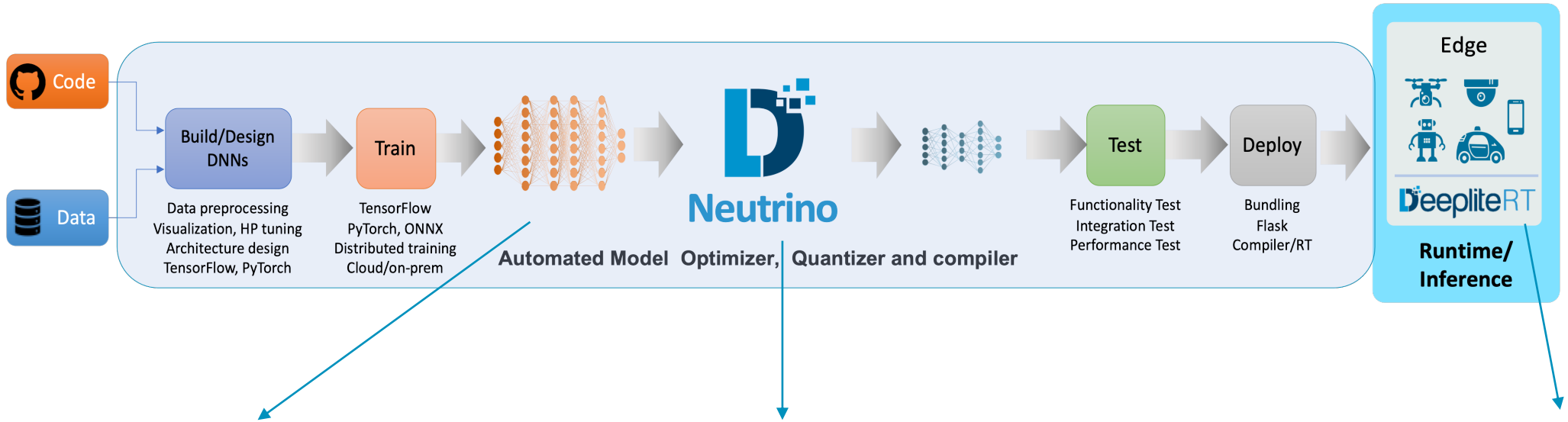
- Increasing cloud costs to support application
- 1000s of cameras in field (Cortex-A53)
- Train a multi class object detection model
- Small, fast & accurate enough to run on device
- Product differentiation

Solution & Result

- Developed unique version of Yolov5 family (code name **Deeplite Piccolo**)
- 2-bit quantization via Neutrino engine
- Running via DeepliteRT on camera
- Ground-breaking performance
  - < 200 ms
  - < 4% impact on accuracy

Person Detection on low-power Arm CPU

Approach



- Tested Yolo5 family on Cortex-A53
- Developed unique version of Yolo5 family (code name Deeplite Piccolo)
- Upload to Deeplite Model Zoo

2 months on multi-GPU

- Piccolo-n 2bit-QAT and Compiler with Deeplite Neutrino
- **< 4%** impact on accuracy

1 month on multi-GPU

- **< 200 ms** running on rasp-pi4b
- Deployment on camera (quadcore Arm Cortex-A53 CPU) as next step

1 week to test

Arm CPU-Powered Person Detection in Action!

Intelligent Speed Adaptation on low-power Arm CPU

Problem

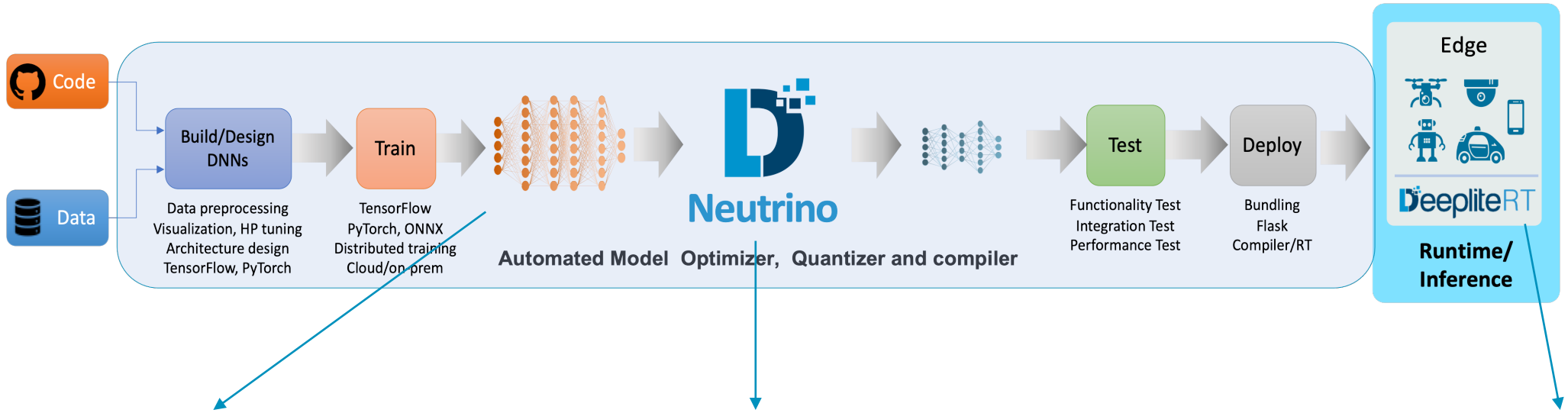
- GPS speed control not efficient & accurate
- Speeding in construction zones = huge expense
- 1000s of units in field (must use existing CPU)
- Train an efficient object detection model
- Small and fast enough to run on Cortex-A53

Solution & Result

- Model training pipeline & testing for Yolo and Resnet-SSD
- Neutrino optimized model (17x size reduction)
- Deeplite platform and run-time produced Yolo5s (2bit quantized model) on rasp-pi4b
  - 256.78 ms
  - < 4% impact on accuracy

Intelligent Speed Adaptation on low-power Arm CPU

Approach



- Train object detection models for construction zones & speed limits >0.80mAP
- Based on Yolo and Resnet-SSD series of models (PyTorch)

2 months on multi-GPU

- **17x compression** over 32bit models with <0.02 mAP drop
- Yolo5s 2bit-QAT and Compiler with Deeplite Neutrino

1 month on multi-GPU

- **256.78 ms** running on rasp-pi4b
- Successfully deployed & running on camera (quadcore Arm Cortex-A53 CPU)

1 month to test/deploy

Always-on, Optimized Computer Vision with Arm NPU

Deeplite Neutrino

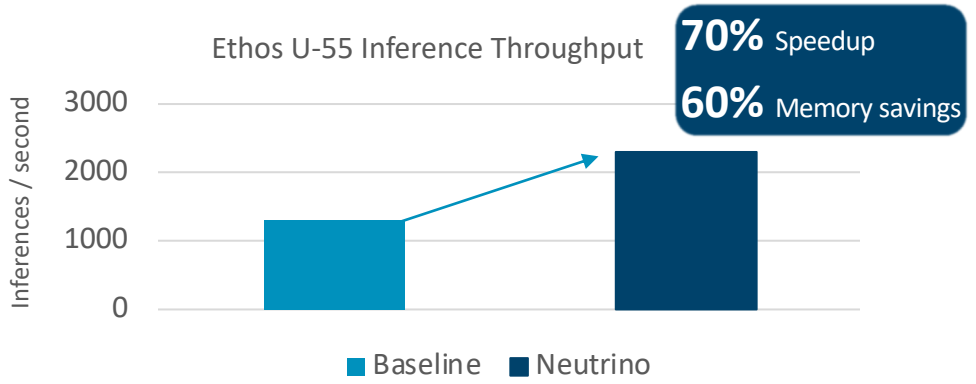
Person Detection, High Efficiency Use Case:

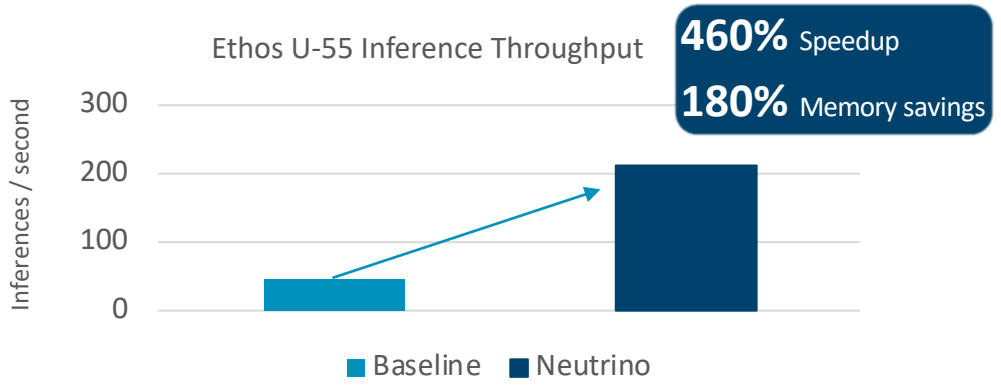
- Baseline MobileNetv1-0.25x model
- Neutrino 2x compression with > 85% final accuracy

Object Classification, High Performance Use Case

- Baseline ResNet-18 model
- Neutrino 22x compression with < 1% drop in accuracy

arm





arm

Go from AI research to the real-world with **DeepliteRT**

Try it Today: dlrt.deeplite.ai

Username: armdlrt

Password: armdlrt

arm AI

AI Virtual Tech Talks Series

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה

arm

Thank you!

Tweet us: @ArmSoftwareDev -> #AIVTT

Check out our Arm Software Developers YouTube channel

Signup now for our next AI Virtual Tech Talk: developer.arm.com/techtalks