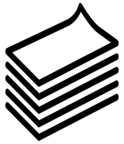
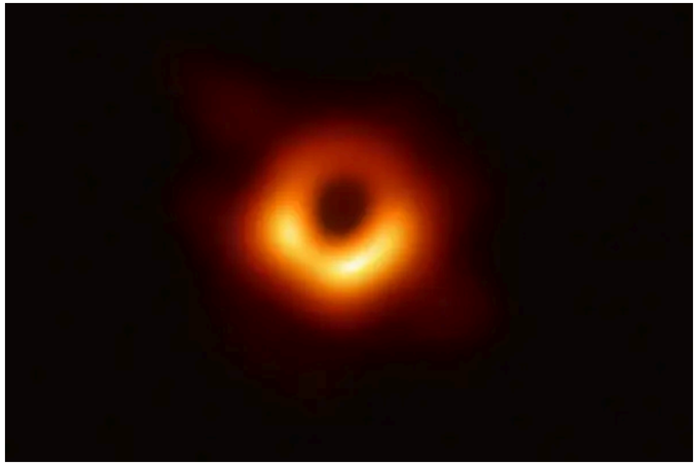
Characterizing the Deployment of Deep Neural Networks on Commercial Edge Devices

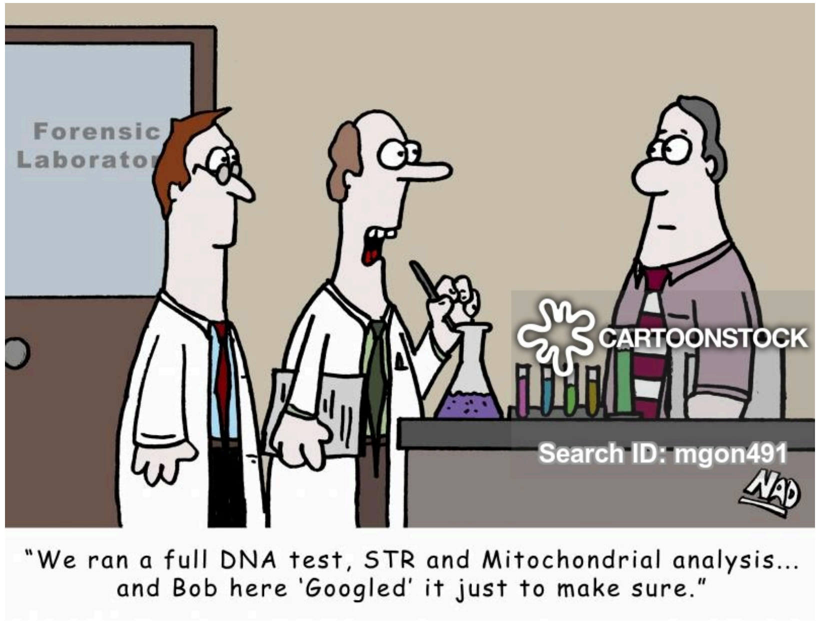
Ramyad Hadidi, Jiashen Cao, Yilun Xie, Bahar Asgari
Tushar Krishna, Hyesoon Kim

Georgia Tech
comparch
SYNERGY

a short story…





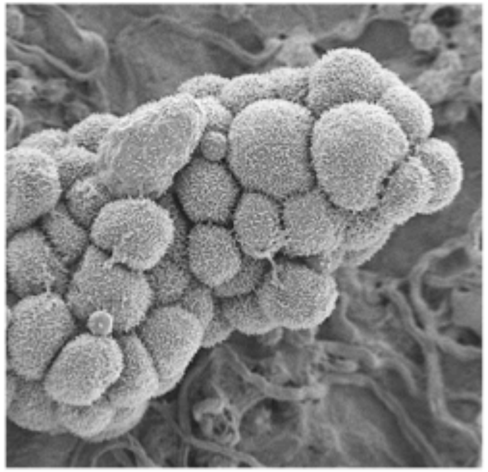


Our aim is to provide
an unbiased characterization of edge devices

Motivation: Deep Learning is Everywhere

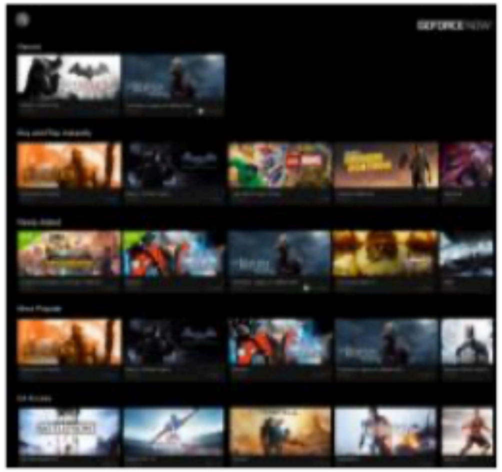
INTERNET & CLOUD

Image Classification
Speech Recognition
Language Translation
Language Processing
Sentiment Analysis
Recommendation



MEDICINE & BIOLOGY

Cancer Cell Detection
Diabetic Grading
Drug Discovery



MEDIA & ENTERTAINMENT

Video Captioning
Video Search
Real Time Translation

SECURITY & DEFENSE

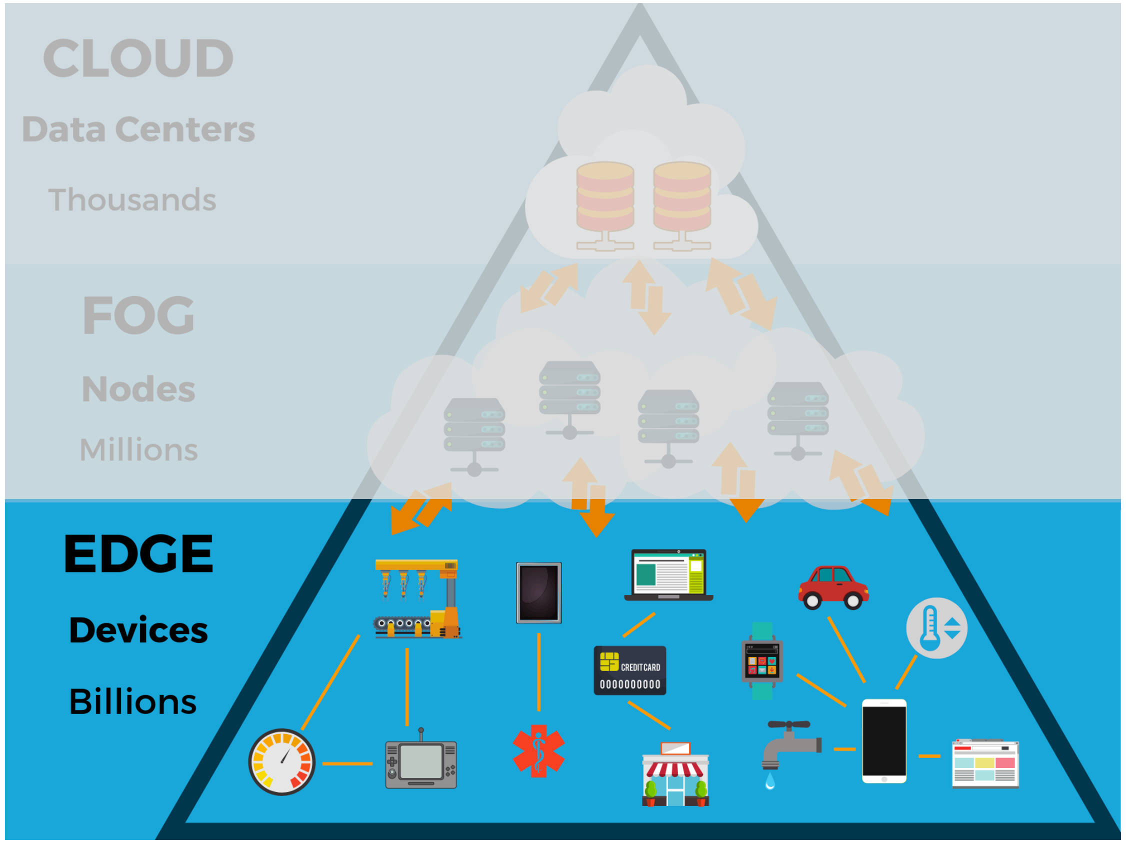
Face Detection
Video Surveillance
Satellite Imagery

AUTONOMOUS MACHINES

Pedestrian Detection
Lane Tracking
Recognize Traffic Sign

In-The-Edge Inferencing

- Some applications are in-the-edge
  - Self-driving cars, smart homes/cities
- Sometimes is the only option
  - No Internet connectivity
  - Intermittent connectivity
- Security and privacy
  - Most straightforward way to preserve privacy and ensure security
  - Personalization
- Cloud is not scalable forever
- Edge could be even faster
  - No cost associated with communication with the cloud
- Sometimes cost efficient



Challenges of In-The-Edge Inferencing

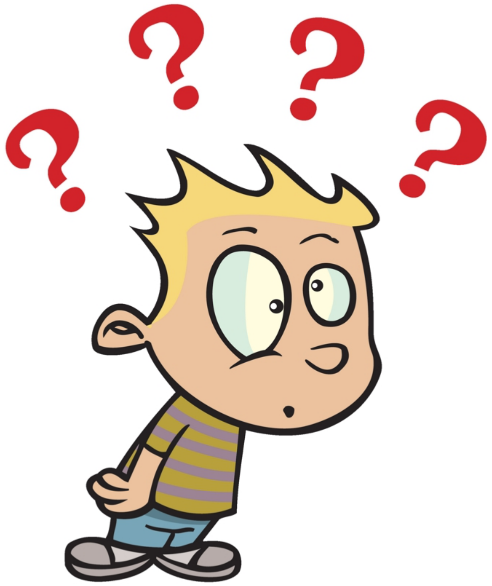
- When to use the cloud?
- Load balancing between edge devices
- API and service management
- Programming model and architectures
- Security, reliability, and fault tolerance

Our Focused Challenge:

Resources of Edge Devices	$\neq$	Intensive Resource Requirements of Real-Time Deep Learning

To Measure is to Know!

- Several companies have released edge-specific devices
- Several frameworks for deep learning
- Several optimizations across HW/SW stack, several papers…
- How to choose one?
  - No unified study
  - Specially for **single-batch** inferencing, the common case for edge
  - Similar endeavors, such as MLPerf. Our focus is more **on the edge**.



Outline

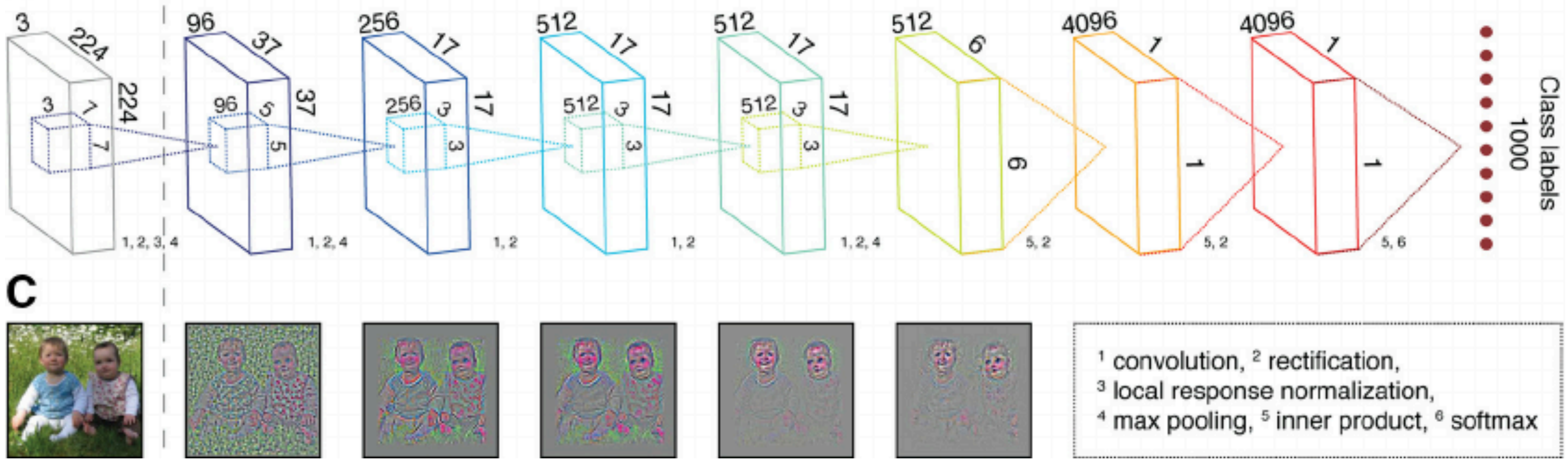
- Introduction & Motivation
- **Deep Learning Models**
- Frameworks & Optimizations
- Hardware Platforms
- Experiments
    - Execution Time Analysis
    - Edge Versus HPC Platforms
    - Virtualization Overhead Study
    - Energy Measurements
    - Power & Time Correlation
    - Framework Analysis
        - Framework Comparisons
        - Edge-Specific Frameworks
        - Software Stack Analysis
    - Temperature Measurements
- Conclusions

Really Short Introduction on DNN

Computation Layers:

- Fully connected (**FC**): Weighted sum
- Convolution (**Conv**): Basically a shared version of fully connected
- Others: Activation, Batch Normalization, Pooling layers

Deep neural network (**DNN**) is basically a stacking of these layers:



Georgia Tech comparch

Our Models

Models: Famous hand-crafted stacking of those layers

We focusing on computer vision, or convolution neural networks (**CNNs**)

Image Recognition

Object recognition, Video recognition

Model Name	Input Size	FLOP (giga)	Number of Parameters	FLOP/Param.
ResNet-18 [44]	224x224	1.83	11.69 m	156.54
ResNet-50 [44]	224x224	4.14	25.56 m	161.97
ResNet-101 [44]	224x224	7.87	44.55 m	176.66
Xception [45]	224x224	4.65	22.91 m	202.97
MobileNet-v2 [46]	224x224	0.32	3.53 m	90.65
Inception-v4 [47]	224x224	12.27	42.71 m	287.29
AlexNet [48]	224x224	0.72	102.14 m	7.05
VGG16 [5]	224x224	15.47	138.36 m	111.81
VGG19 [5]	224x224	19.63	143.66 m	136.64
VGG-S [5]	32x32	0.11	32.11 m	3.42
VGG-S [5]	224x224	3.27	102.91 m	31.77
CifarNet [49]	32x32	0.01	0.79 m	12.65
SSD [39] with MobileNet-v1 [40]	300x300	0.98	4.23 m	236.07
YOLOv3 [41], [42]	224x224	38.97	62.00 m	628.54
TinyYolo [42]	224x224	5.56	15.87 m	350.35
C3D [43]	12x112x112	57.99	89.00 m	734.05

FLOP and #**Parameters**:

Reported for every DNN

Proxy for compute/memory

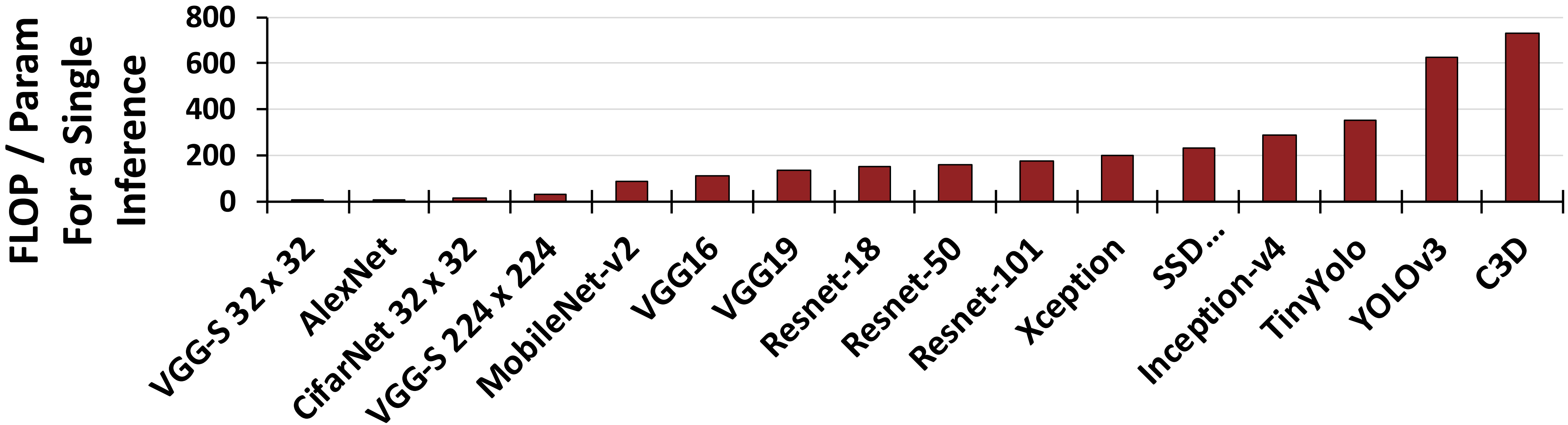
FLOP/Parameter:

Represents reuse possibility

Characterized Models FLOP/Param

We study a wide range of models

- Models sorted by their FLOP/Param
  - Compute-intensive (right side) vs. Memory-intensive (left side)
  - Efficient model design? e.g., Accuracy%/Param



Georgia Tech comparch

Outline

- Introduction & Motivation
- Deep Learning Models
- **Frameworks & Optimizations**
- Hardware Platforms
- Experiments
  - Execution Time Analysis
  - Edge Versus HPC Platforms
  - Virtualization Overhead Study
  - Energy Measurements
  - Power & Time Correlation
  - Framework Analysis
    - Framework Comparisons
    - Edge-Specific Frameworks
    - Software Stack Analysis
  - Temperature Measurements
- Conclusions

Frameworks

Popular off-the-shelf DNN frameworks provide tools to design, train, and deploy DNN models

- We study widely-used frameworks:
  - **Common**: TensorFlow (+Keras), Pytorch, DarkNet, Caffe1/2
  - **Specific/Mobile Platforms**: TFLite, Movidius, TensorRT

	TensorFlow	TFLite	Caffe1/2	Movidius	PyTorch	TensorRT	DarkNet
Language†	Python						C
Industry Backed	✓						✗
Training Framework	✓	✗	✓				
Usability	***	*	**	*	***	**	**
Adding New Models	**	*	***	*	***	**	***
Pre-Defined Models	***	*	**	*	***	**	**
Documentation	**	*	*	*	***	*	*
No Extra Steps	✓	✗	✓	✗	✓	✓	✓
Mobile Device Deployment	✗	✓		✗			
Low-Level Modifications	**	*	**	*	*	*	***
Compatibility with Others	*	*	*	*	*	**	*

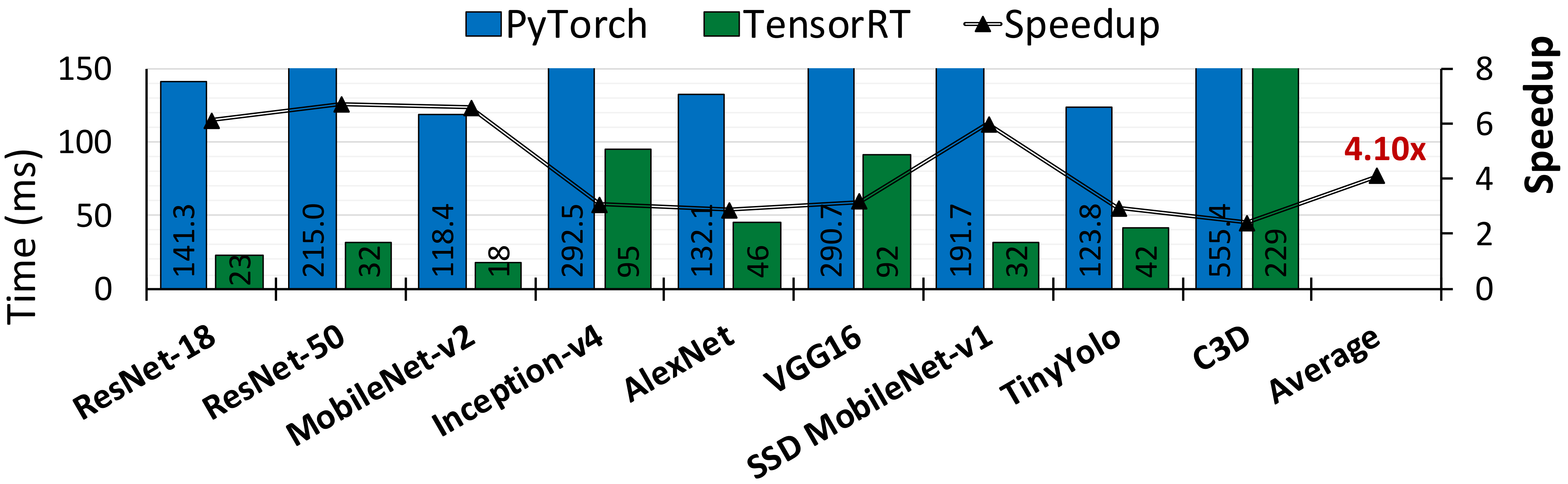
PyTorch Movidius TensorFlowLite TensorFlow + Keras TensorRT Caffe2

Generality vs. Specialization

Several design decisions that tradeoff:

Generality to Platforms ≠ Specialization & Performance

For instance, TensorRT over PyTorch on Nvidia Jetson Nano: **4.10x Speedup**



Why? Optimizations!

Each Framework has its own set of optimizations:

- Generality contradicts with most of the optimizations
- Optimizations limits hardware platforms
- We study officially supported optimizations for inference

		TensorFlow	TFLite	Caffe1/2	Movidius	PyTorch	TensorRT	DarkNet
Optimizations	Quantization	✓	✓	✓	✓	✓	✓	✗
	Mixed-Precision‡	✗	✗	✗	✗	✗	✓	✗
	Dynamic Graph	✗§	✗§	✗	✗	✓	✓	✗
	Pruning‡‡	✓††	✓	✗	✗	✗	✓	✗
	Fusion	✓††	✓	✗	✓	✗	✓	✗
	Auto Tuning	✗	✗	✗	✗	✗	✓	✗
	Half-Precision	✓	✓	✓	✓	✓	✓	✗

Optimizations

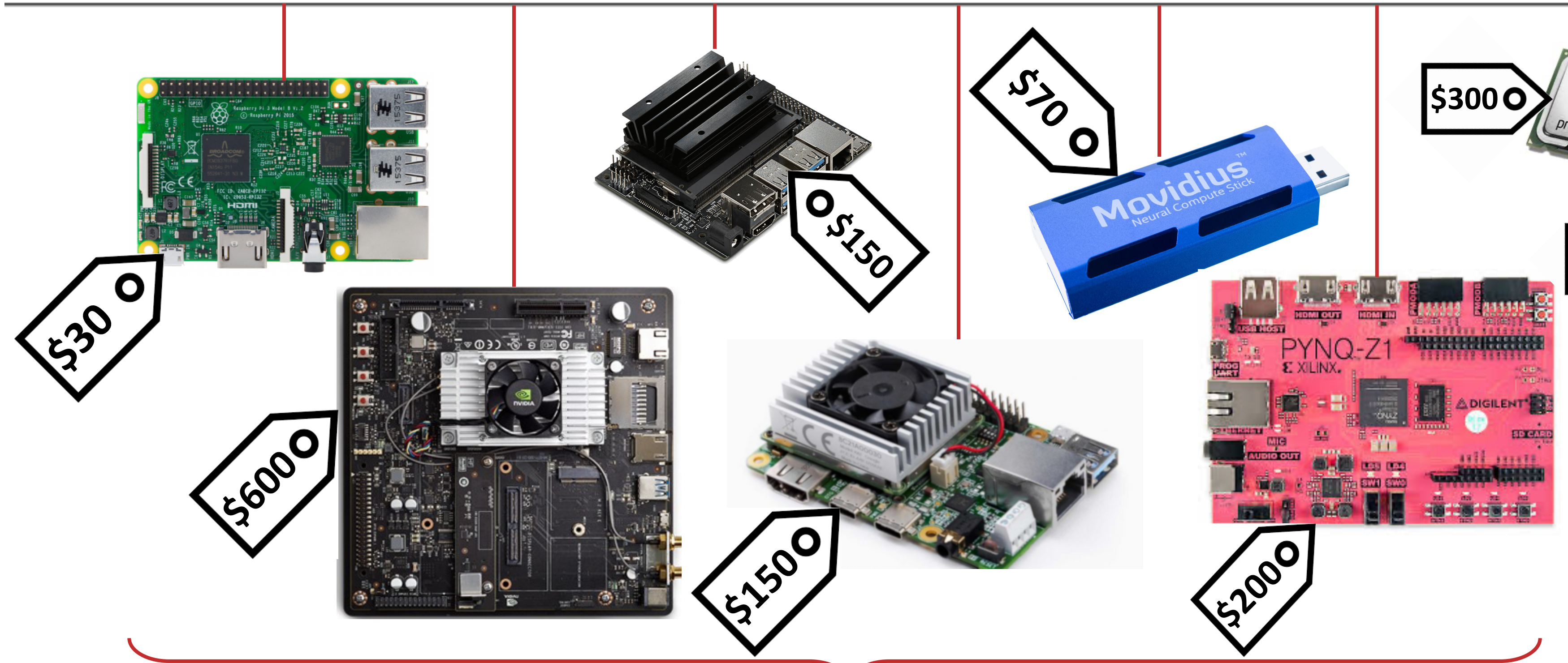
Please check the paper for discussions about each optimization

Outline

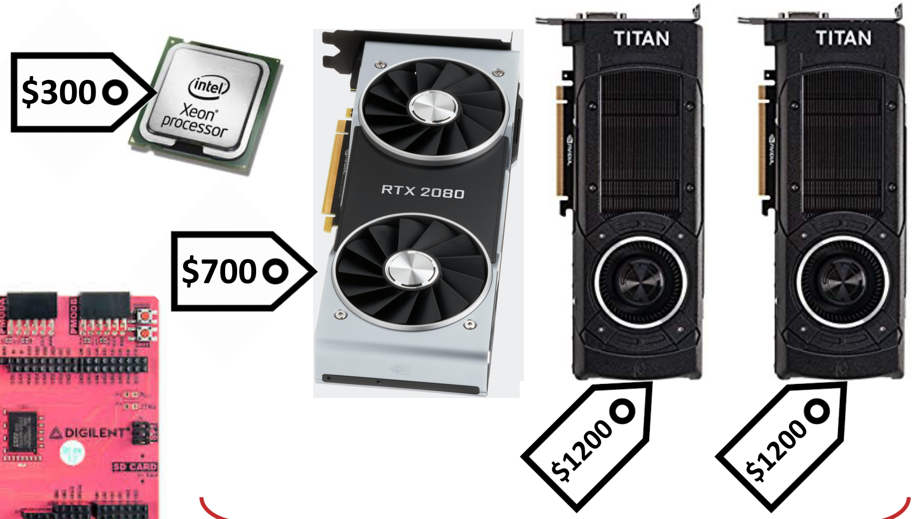
- Introduction & Motivation
- Deep Learning Models
- Frameworks & Optimizations
- **Hardware Platforms**
- Experiments
  - Execution Time Analysis
  - Edge Versus HPC Platforms
  - Virtualization Overhead Study
  - Energy Measurements
  - Power & Time Correlation
  - Framework Analysis
    - Framework Comparisons
    - Edge-Specific Frameworks
    - Software Stack Analysis
  - Temperature Measurements
- Conclusions

Hardware Platforms

Category	IoT/Edge Devices	GPU-Based Edge Devices		Custom-ASIC Edge Accelerators		FPGA Based	HPC Platforms: CPU	HPC Platforms: GPU		
Platform	Raspberry Pi 3B [34]*	Jetson TX2 [69]	Jetson Nano [36]	EdgeTPU [35]	Movidius NCS [37]♦	PYNQ-Z1 [64]	Xeon	RTX 2080	GTX Titan X	Titan Xp



Edge Platforms



HPC Platforms

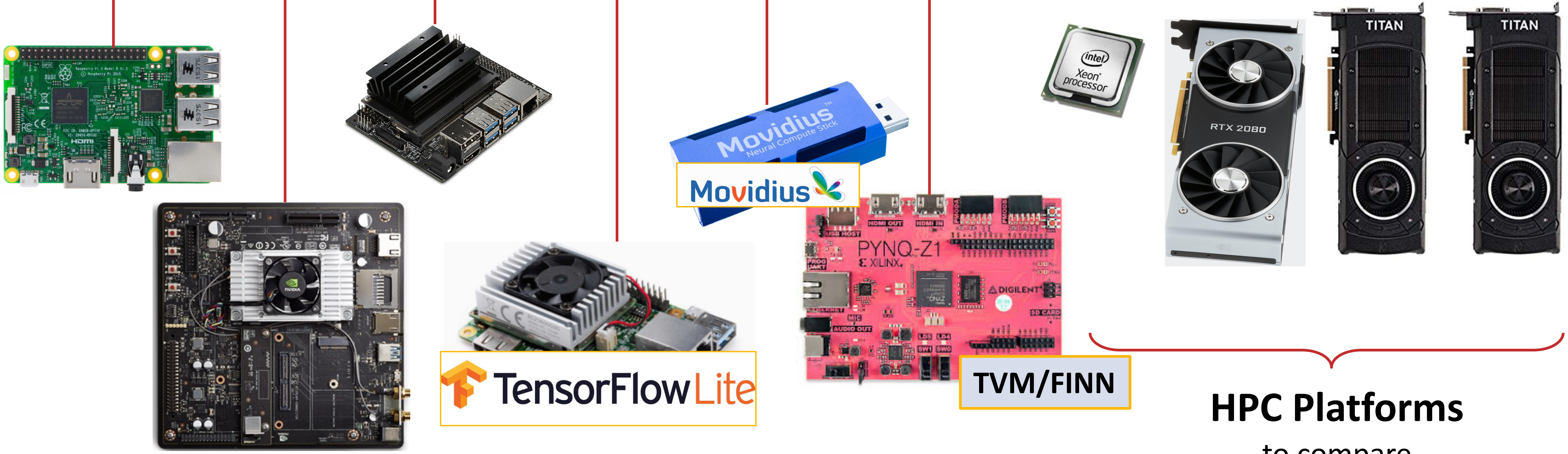
to compare performance of single-batch inferencing

* Detailed HW description in the paper

Georgia Tech comparch

Hardware Platforms

Category	IoT/Edge Devices	GPU-Based Edge Devices		Custom-ASIC Edge Accelerators		FPGA Based	HPC Platforms CPU	HPC Platforms GPU		
Platform	Raspberry Pi 3B [34]*	Jetson TX2 [69]	Jetson Nano [36]	EdgeTPU [35]	Movidius NCS [37]♦	PYNQ-Z1 [64]	Xeon	RTX 2080	GTX Titan X	Titan Xp



Edge Platforms

HPC Platforms

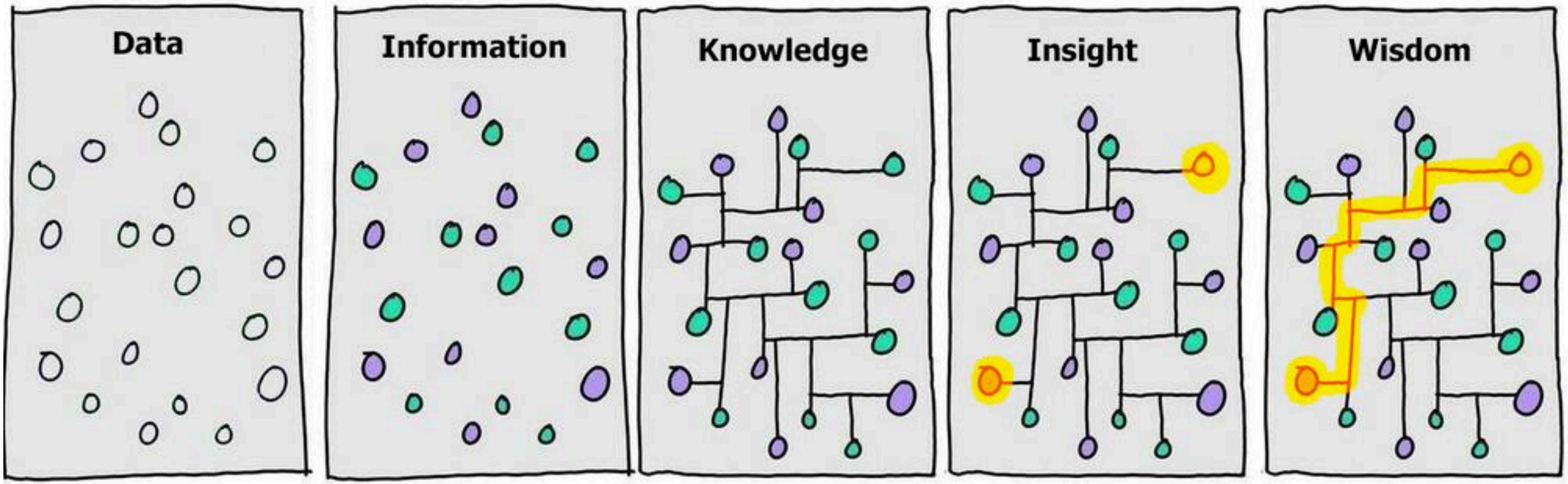
to compare performance of single-batch inferencing

* Detailed HW description in the paper

Outline

- Introduction & Motivation
- Deep Learning Models
- Frameworks & Optimizations
- Hardware Platforms
- **Experiments**
  - Execution Time Analysis
  - Edge Versus HPC Platforms
  - Virtualization Overhead Study
  - Energy Measurements
  - Power & Time Correlation
  - Framework Analysis
    - Framework Comparisons
    - Edge-Specific Frameworks
    - Software Stack Analysis
  - Temperature Measurements
- Conclusions

Experiments

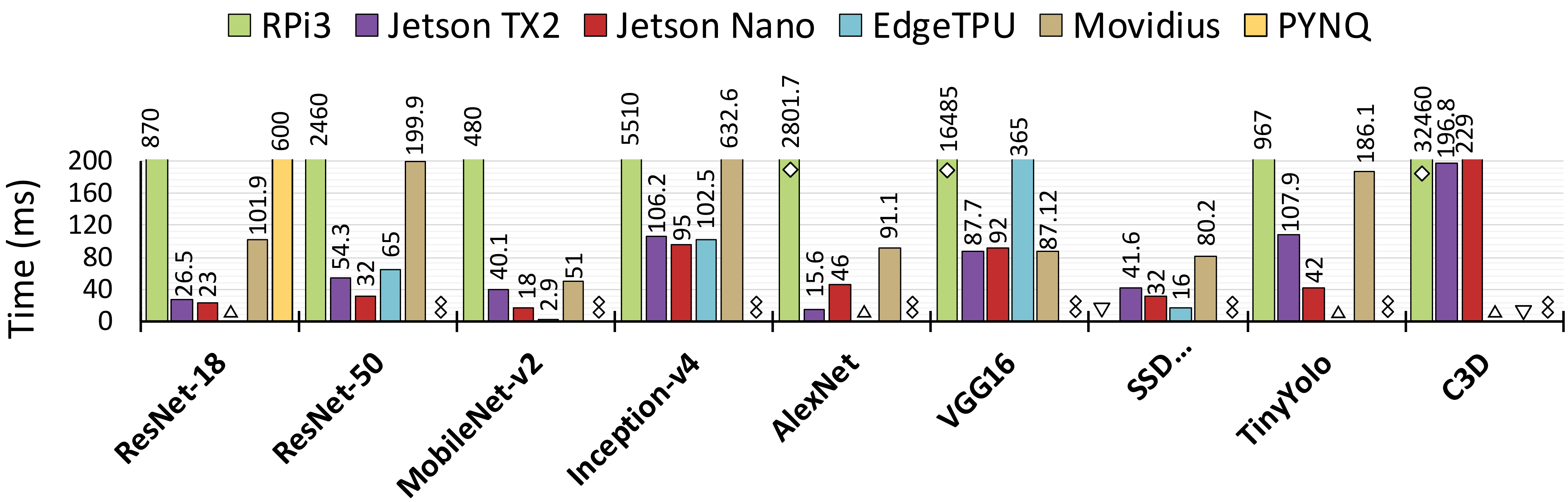


Question

Which device, regardless of frameworks, performs the best?

Execution Time Analysis

Time per inference on all edge devices with best performing framework



△ ⩫ ◇ ▽ Implementation Details, See Table III

Georgia Tech comparch

Takeaways

- Raspberry Pi executes all models (generality)
- GPU-based platforms achieve a good balance between performance and generality
- EdgeTPU performs the best on MobileNet
  - But has several compilation, quantization, retraining issues for extending to other models
- Movidius results are all close to others, but not the best
- **No overall best device**

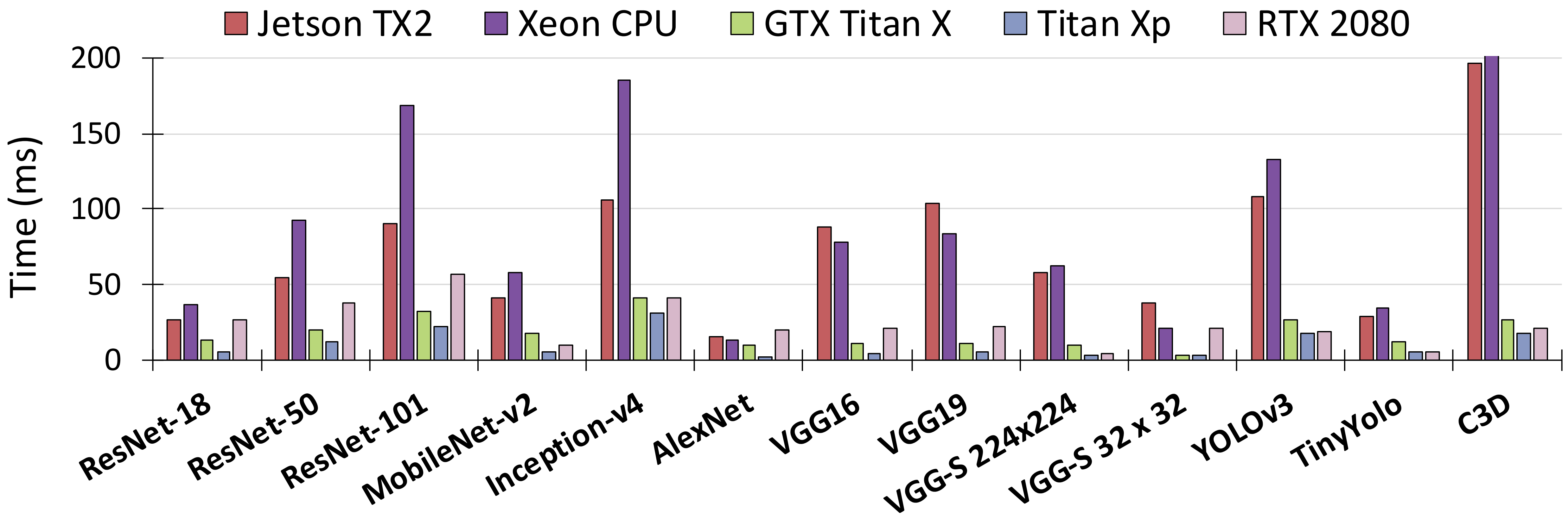
Question

For edge specific single-batch inferences…

Are HPC platforms really good at them?

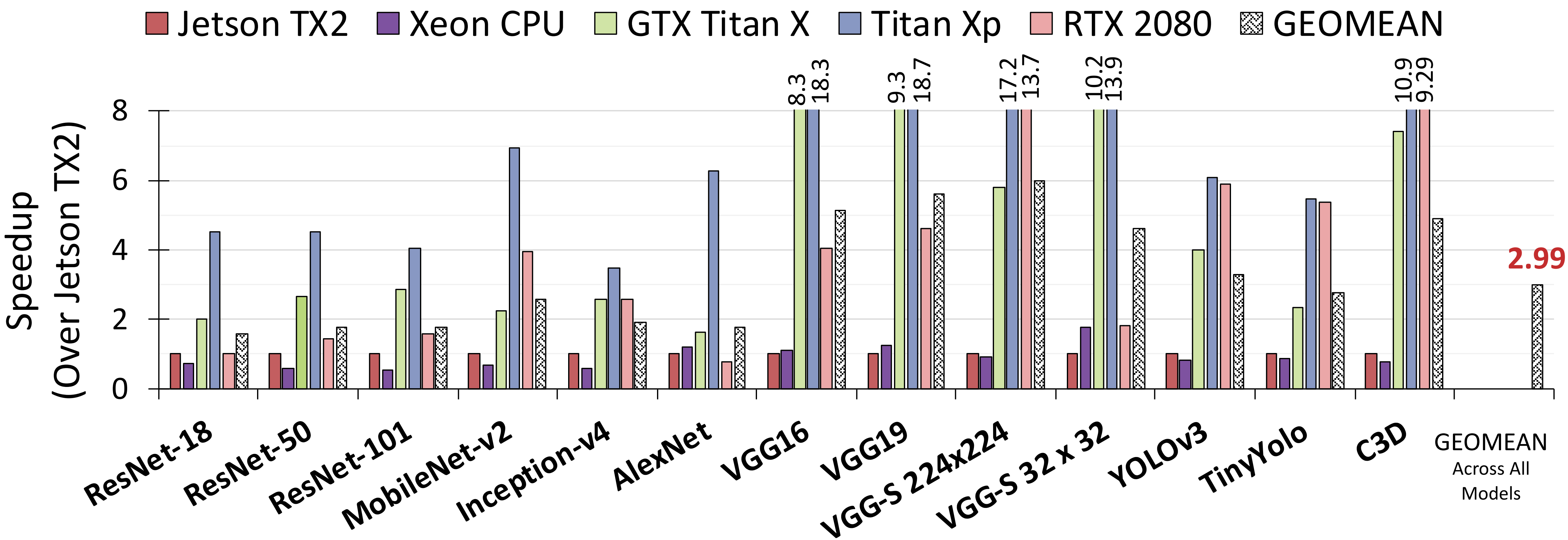
Edge vs. HPC Platforms - Time

Time per inference between edge and HPC platforms with **PyTorch**



Edge vs. HPC Platforms - Speedup

Time per inference between edge and HPC platforms with **PyTorch**



Georgia Tech comparch

Takeaways

- HPC platforms are designed to be **throughput-oriented** for **multi-batch** DNN computations
- Single-batch inferencing is **latency-sensitive**
  - Requires new design philosophy
- Then, CPUs should perform better, they are latency sensitive…
  - No, our benchmarks are compute-bounded on CPU
- **HPC Platforms are not as good for single-batch inferecing**

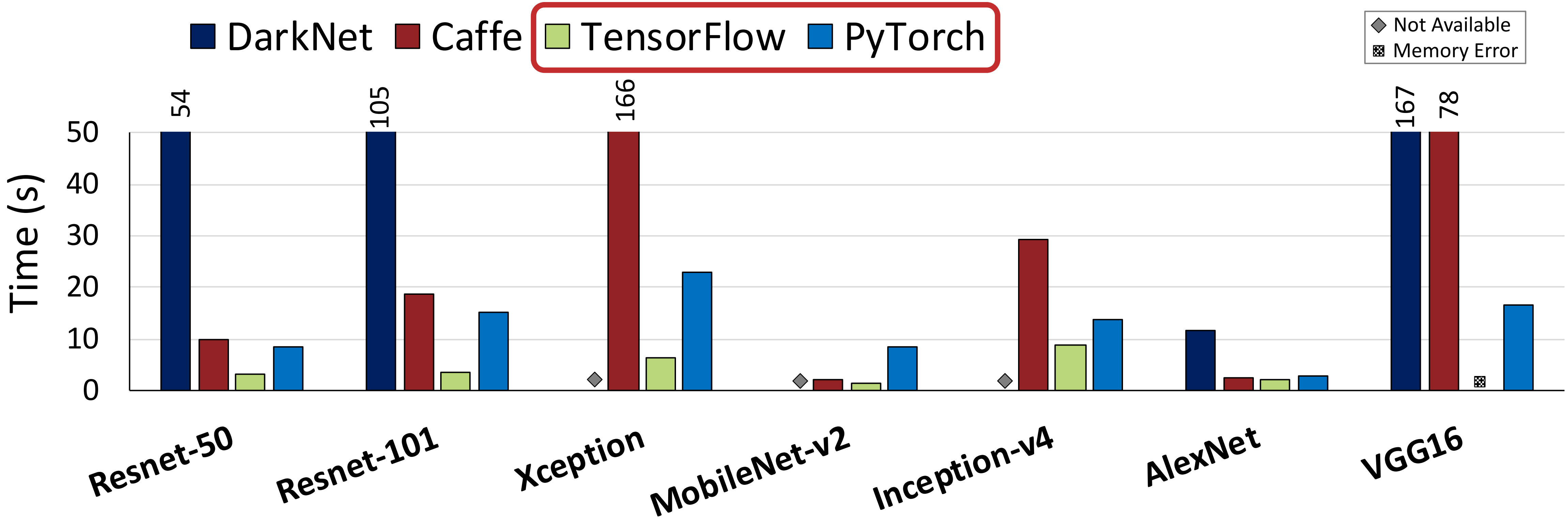
Question

Does the choice of which general framework matter?

(we saw a case for edge-specific frameworks before)

Frameworks Comparison - RPi

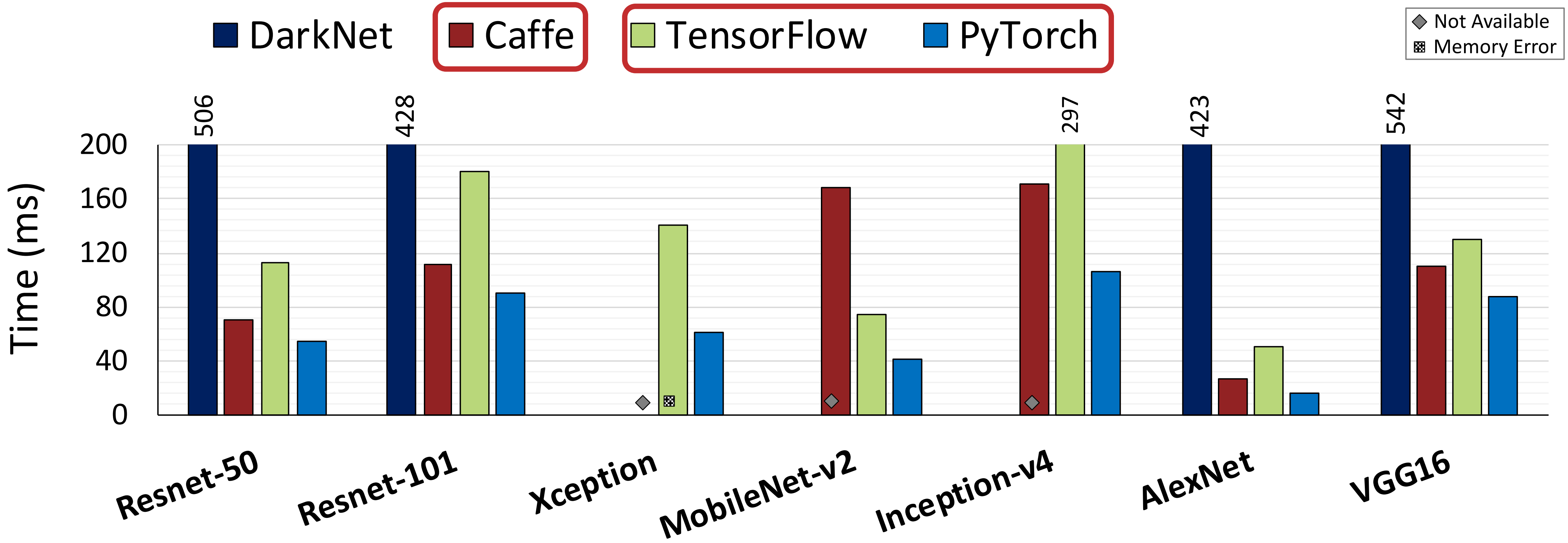
Time per inference on **Raspberry Pi** across different frameworks.



TensorFlow perform better than PyTorch

Frameworks Comparison - TX2

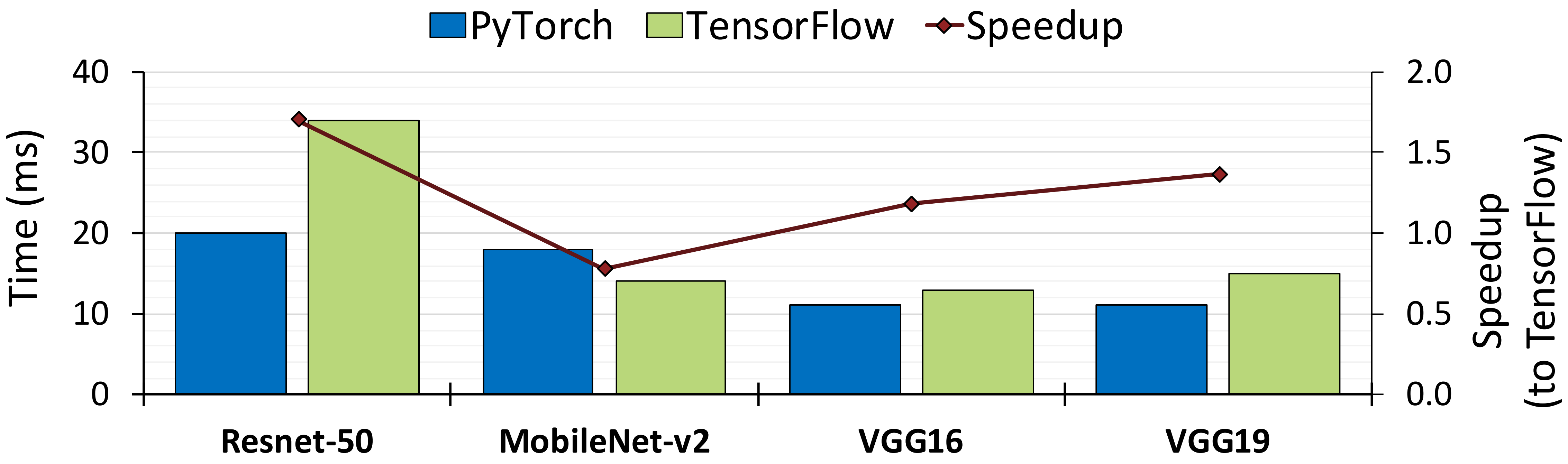
Time per inference on **Jetson TX2** across different frameworks



PyTroch perform better than TensorFlow

Frameworks Comparison - Titan X

Time per inference on **Titan X** (TensorFlow and PyTorch)



None of PyTroch & TensorFlow are always the best

Takeaways

- On Raspberry Pi, TensorFlow performs the best
  - But, not as good as edge-specific platforms
- On Jetson TX2, PyTorch performs the best
- Interestingly, on Jetson, TX2 Caffe, not updated after 2017, achieves a similar results
- Why?
  - Dynamic vs. static computation graph
  - Tensorflow numerous APIs and hard usability

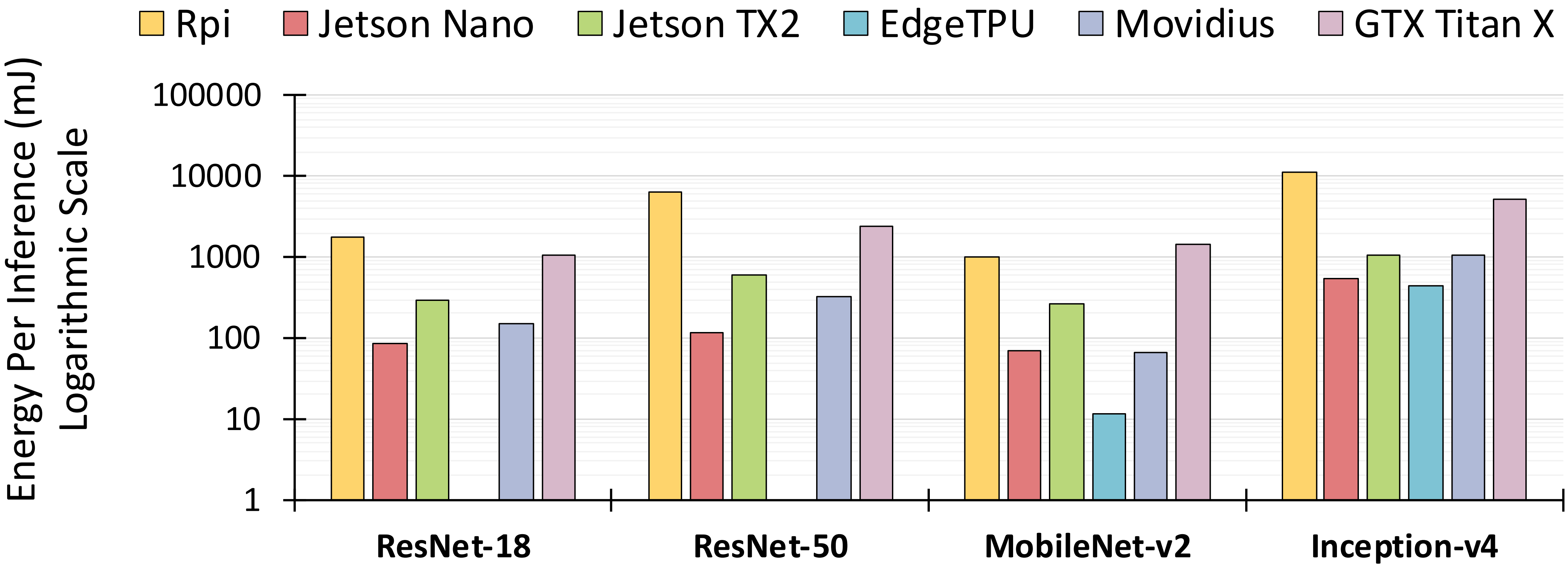
Question

Energy is important for edge devices.

How do devices compare if we add energy?

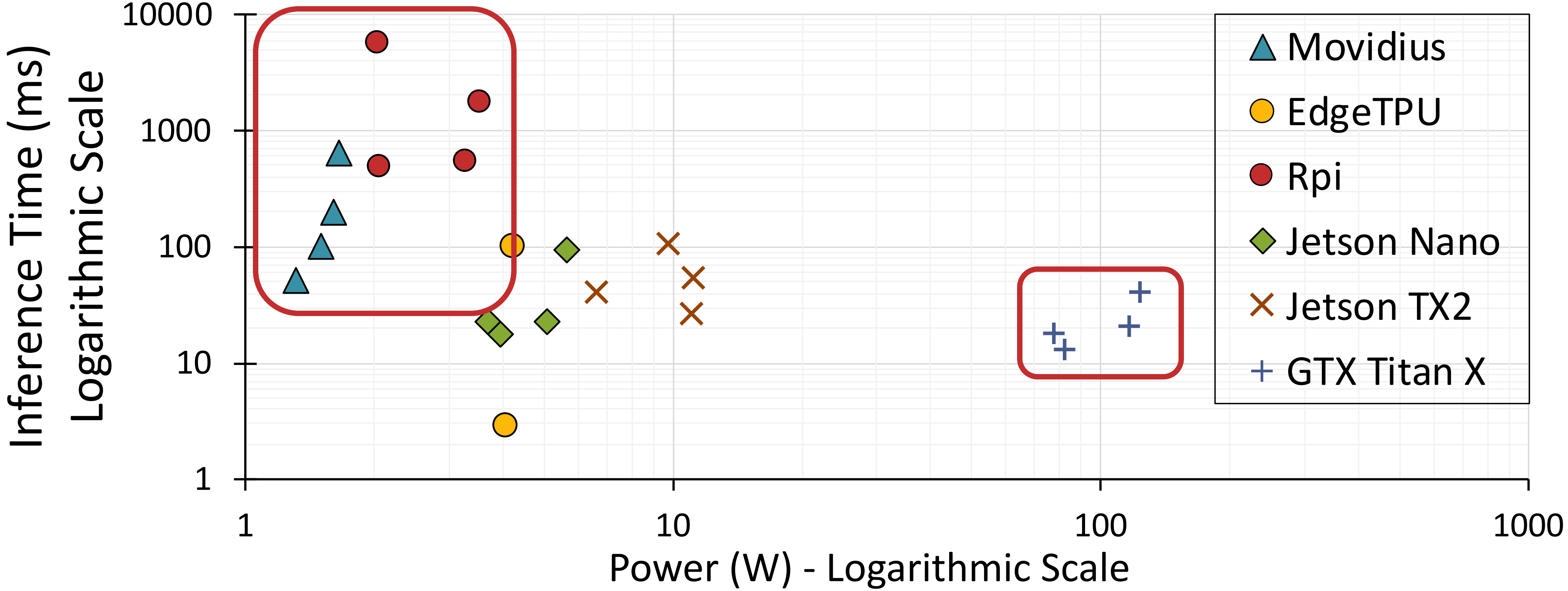
Energy Measurements

Energy per Inference for a single inference.



Power & Time Correlation

Measuring correlation between power and execution time.



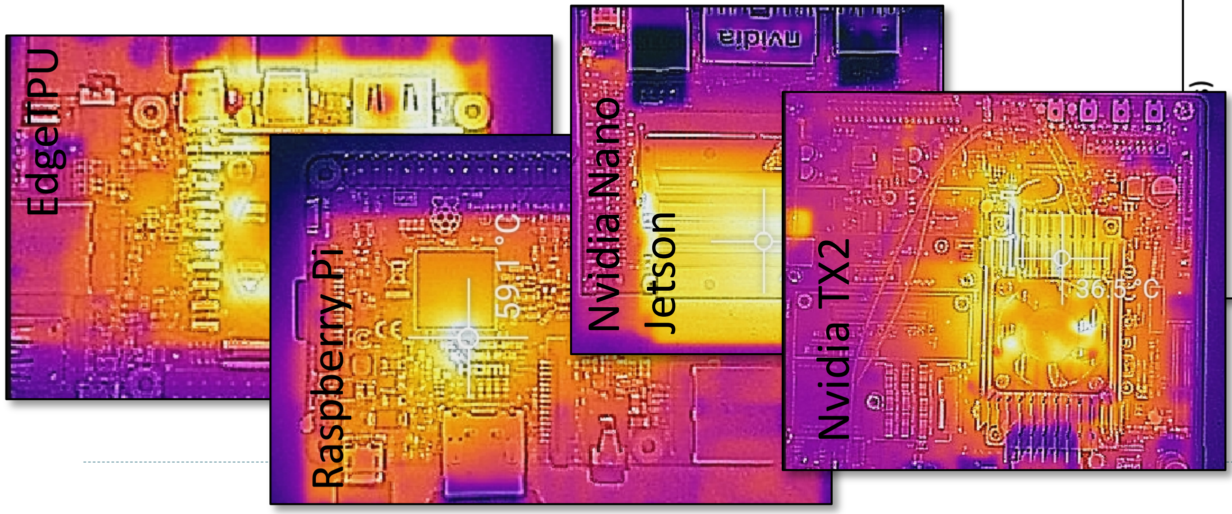
Takeaways

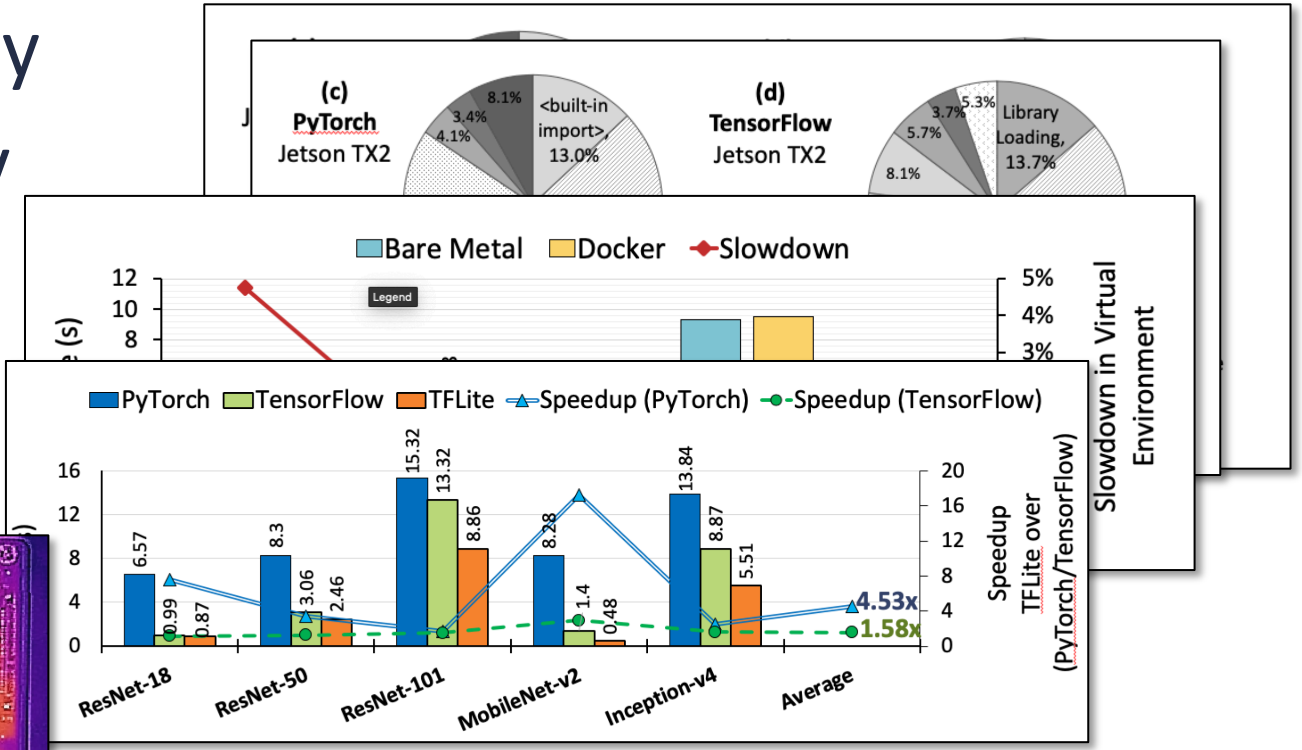
- GPU-based platforms have 5x energy saving than their HPC-based counterparts
- Raspberry Pi, when considering time-power graph, is actually a good device!
  - Besides Raspberry Pi has several other components that consume energy
- Movidius is the most energy-efficient device
- EdgeTPU and Jetsons tradeoff energy efficiency with performance

Other Experiments

Please check paper for all the experiments

- Virtualization overhead study
- TF-lite and TensorFlow study
- Software stack analysis
- Temperature behavior





Georgia Tech comparch

Codes on GitHub

Our codebase and implementation guide are available on GitHub:

https://github.com/gthparch/edgeBench

Please help us in extending current models and frameworks.

README.md

Edge Bench

Table of Contents

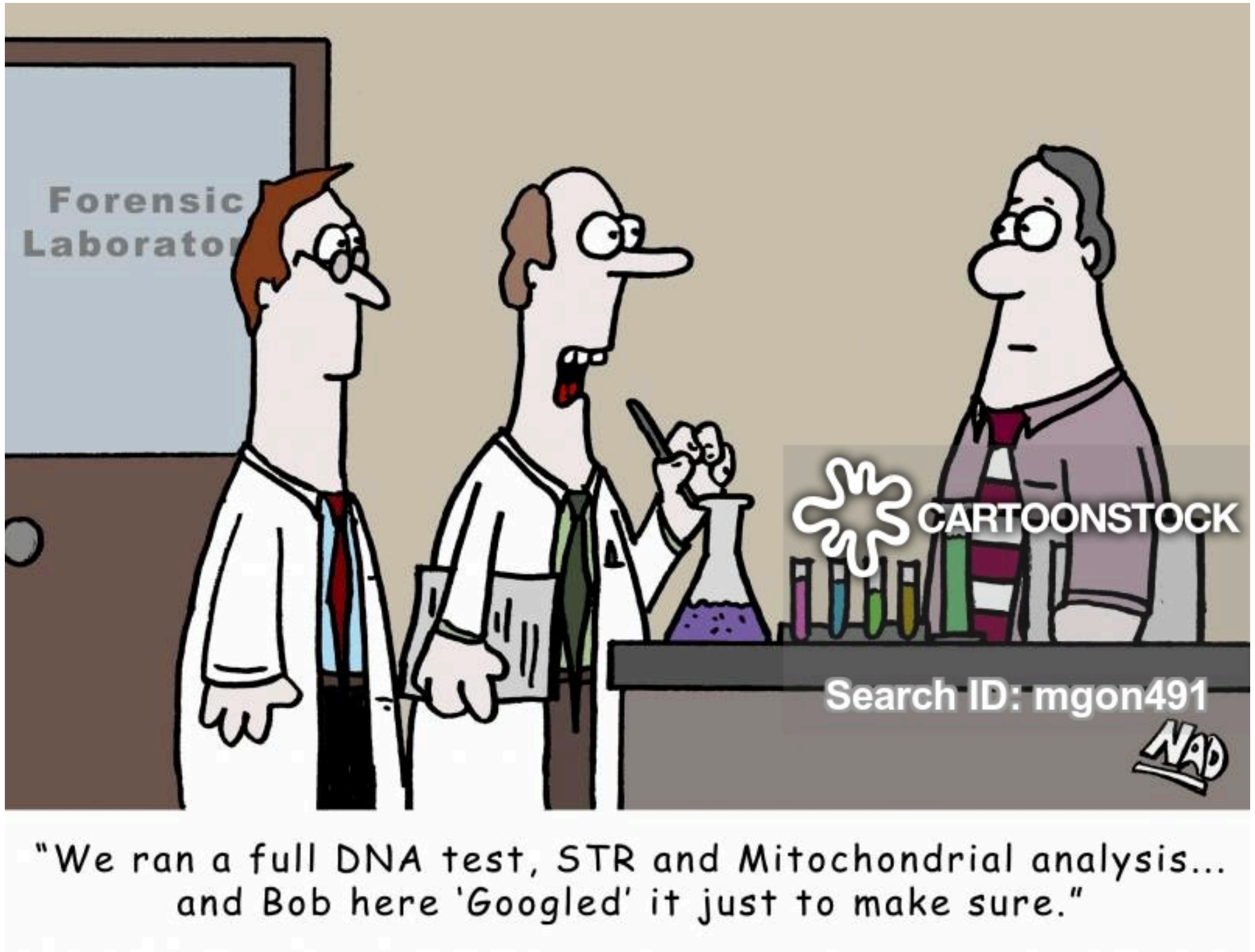
Supported Models

	PyTorch	TensorFlow	DarkNet	Caffe
ResNet-18	✓	✓	-	-
ResNet-50	✓	✓	✓	✓
ResNet-101	✓	✓	✓	✓
Xception	✓	✓	-	✓
MobileNet-v2	✓	✓	-	✓
Inception-v4	✓	✓	-	✓
AlexNet	✓	✓	✓	✓
VGG-11 (224x224)	✓	-	-	-
VGG-11 (32x32)	✓	-	-	-
VGG-16	✓	✓	✓	✓
VGG-19	✓	✓	-	✓
CifarNet (32x32)	✓	-	-	-
SSD MobileNet-v1	✓	-	-	-
YOLOv3	✓	-	✓	-
Tiny YOLO	✓	✓	✓	-
C3D	✓	-	-	-

Conclusions

- Which edge device is the best? Depends
- Are HPC platforms good for single-batch inferences? Only 3x
- Does edge-specific platforms help? Yes, but with a cost
- Does the choice of general framework matter? Yes, but no definite answer on which
- What does help the performance the most? HW-SW codesigns
- What does energy measurements show? Tradeoff between energy consumption and inference time

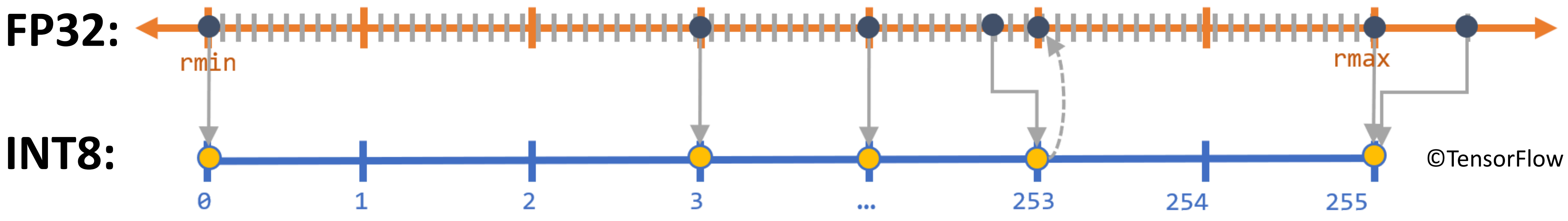
Conclusions



Backup Slides

Optimizations: Quantization

Commonly Supported: For inference, it has been shown that instead of **FP32**, we can use **INT8** without any accuracy loss:



- Easy to implement
- Every hardware supports
- Great gains!

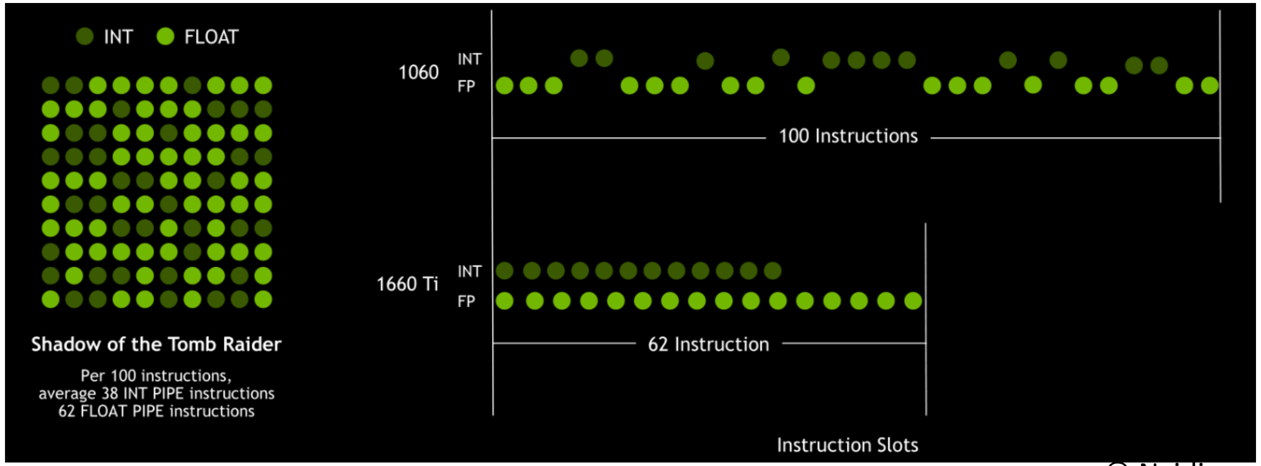
INT8 Operation	Energy Saving vs FP32	Area Saving vs FP32
Add	30x	116x
Multiply	18.5x	27x

*Dally, 2015

Optimizations: Mixed-Precision

Not Commonly Supported: Use a mix of INT8, INT4 units.

- Need to ensure if a DNN model tolerate INT4 precision.
- Hardware support needed
- Not easy to implement, needs hardware support
  - For instance: NVIDIA Turing Architecture (e.g., Nvidia Nano Jetson)



Hardware Platforms

THE SPECIFICATIONS OF HARDWARE PLATFORMS USED IN THIS PAPER.

Category	IoT/Edge Devices	GPU-Based Edge Devices		Custom-ASIC Edge Accelerators		FPGA Based	HPC Platforms CPU	HPC Platforms GPU		
Platform	Raspberry Pi 3B [34]⋆	Jetson TX2 [69]	Jetson Nano [36]	EdgeTPU [35]	Movidius NCS [37]♦	PYNQ-Z1 [64]	Xeon	RTX 2080	GTX Titan X	Titan Xp
CPU	4-core Ctx.A53 @1.2 GHz*	4-core Ctx.A57 2-core Denver2 @2 GHz	4-core Ctx.A57 @1.43 GHz	4-core Ctx.A53 & Ctx.-M4 @1.5 GHz	N/Ap	4-core Ctx.A9 @650 MHz	2x 22-core E5-2696 v4 @2.20GHz	N/Ap*	N/Ap	N/Ap
GPU	No GPGPU	256-core Pascal μA	128-core Maxwell μA	N/Ap	N/Ap	N/Ap	N/Ap	2944-core Turing μA	3072-core Maxwell μA	3840-core Pascal μA
Accelerator	N/Ap	N/Ap	N/Ap	EdgeTPU	Myriad 2 VPU	ZYNQ XC7Z020	N/Ap	N/Ap	N/Ap	N/Ap
Memory†	1 GB LPDDR2	8 GB LPDDR4	4 GB LPDDR4	N/Av*	N/Av	630 KB BRAM 512 MB DDR3	264 GB DDR4	8 GB GDDR6	12 GB GDDR5	12 GB GDDR5X
Idle Power‡	1.33	1.90	1.25	3.24	0.36	2.65	≈70	≈39	≈15	≈55
Average Power‡	2.73	9.65	4.58	4.14	1.52	5.24	300 TDP	≈	≈100	≈
Platform	All	All	All	TFLite	NCSDK	TVM/FINN	All	All	All	All

† Effective memory size used for acceleration/execution of DNNs, e.g., GPU/CPU/Accelerator memory size. * Ctx.: Arm Cortex. N/Ap: Not applicable. N/Av: Not available.
‡ : Measured idle and average power while executing DNNs, in Watts. ⋆ : Raspberry Pi 4B [70], with 4-core Ctx.A72 and maximum of 4 GB LPDDR4, was released after this paper acceptance. With better memory technology and out-of-order execution, Raspberry Pi 4B is expected to perform better. ♦ Intel Neural Compute Stick 2 [61] with a new VPU chip and support for several frameworks was announced during paper submission, but the product was not released.

Experiments Frameworks

THE SUMMARY OF EXPERIMENTS DONE IN THIS PAPER.

Experiments	**Execution Time**	**Framework Analysis)**						**Edge vs. HPC**		**Virtualization Overhead**	**Energy Measurments**		**Temperature**
Section/Figure	VI-A/2	VI-B/3	VI-B/4	VI-B/6	VI-B/7	VI-B/8	VI-B/5	VI-C/9	VI-C/10	VI-D/13	VI-E/11	VI-E/12	VI-F/14
Metric	Inference Time (ms or s)						Latency Breakdown	Inference Time (ms)	Speedup Over TX2	Inference Time (s)	Energy per Inference (mJ)	Inf. Time (ms) vs. Power (w)	Temp-erature (°C)
FW/Devices	RPi/TFLite,TF Nano/T-RT TX2/PT EdgeTPU/TFLite Mavidus/NCSDK PYNQ/TVM	RPi/DarkNet RPi/Caffe RPi/TF RPi/PT	TX2/DarkNet TX2/Caffe TX2/TF TX2/PT	GTX/TF GTX/PT	Nano/T-RT Nano/PT	RPi/TF RPi/T-Lite	RPi/PT RPi/TF TX2/PT TX2/TF	TX2/PT Xeon/PT GTX/PT T-XP/PT 2080/PT	TX2/PT Xeon/PT GTX/PT T-XP/PT 2080/PT	Bare Metal RPi/TF Docker RPi/TF	RPi/TFLite Nano/T-RT TX2/PT EdgeTPU/T-Lite Mavidus/NCSDK GTX/PT	RPi/TFLite Nano/T-RT TX2/PT EdgeTPU/T-Lite Mavidus/NCSDK GTX/PT	RPi/TFLite Nano/T-RT TX2/PT EdgeTPU/T-Lite Mavidus/NCSDK GTX/PT

FW: Framework, **TX2**: Jetson TX2, **Nano**: Jetson Nano, **PT**: PyTorch, **TF**: TensorFlow, **TFLite**: TensorFlow Lite, **T-RT**: Tensor RT, **GTX**: GTX Titan X, **T-XP**: Titan Xp, **2080**: RTX 2080

Execution Time Analysis - Legend

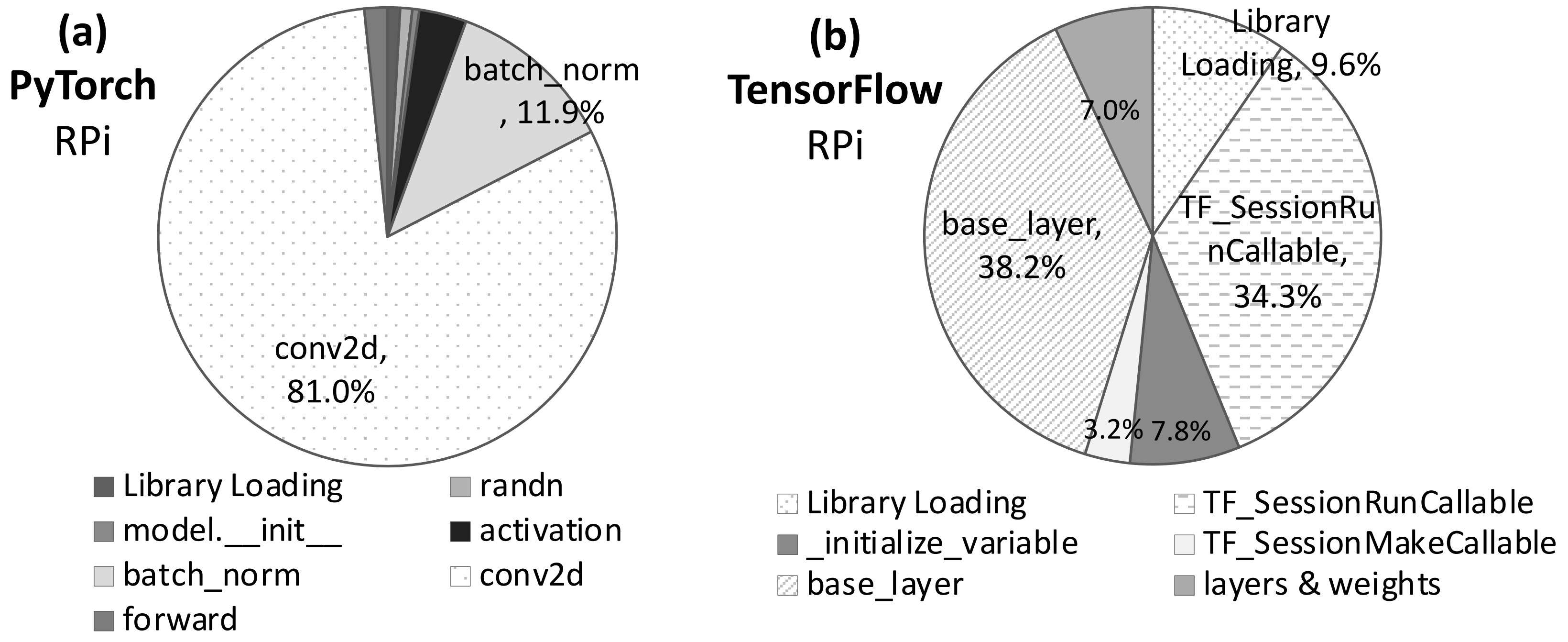
MODELS AND PLATFORMS COMPATIBILITY MATRIX.

Model \ Platform	RPi3	Jetson TX2	Jetson Nano	EdgeTPU	Movidius	PYNQ
ResNet-18	✓	✓	✓	△	✓	✓
ResNet-50	✓	✓	✓	✓	✓	◇◇
MobileNet-v2	✓	✓	✓	✓	✓	◇◇
Inception-v4	✓	✓	✓	✓	✓	◇◇
AlexNet	◇	✓	✓	△	✓	◇◇
VGG16	◇	✓	✓	✓	✓	◇◇
SSD MobileNet-v1	▽	✓	✓	✓	✓	◇◇
TinyYolo	✓	✓	✓	△	✓	◇◇
C3D	◇	✓	✓	△	✓	◇◇

◇ Large memory usage, uses dynamic graph.
▽ Code incompatibility. ◇◇ Large BRAM usage. Requires accessing host DDR3, considerably slowdowns execution.
△ Barriers in converting models to TFLite. Check §VI-A.

Software-Stack Analysis - RPi

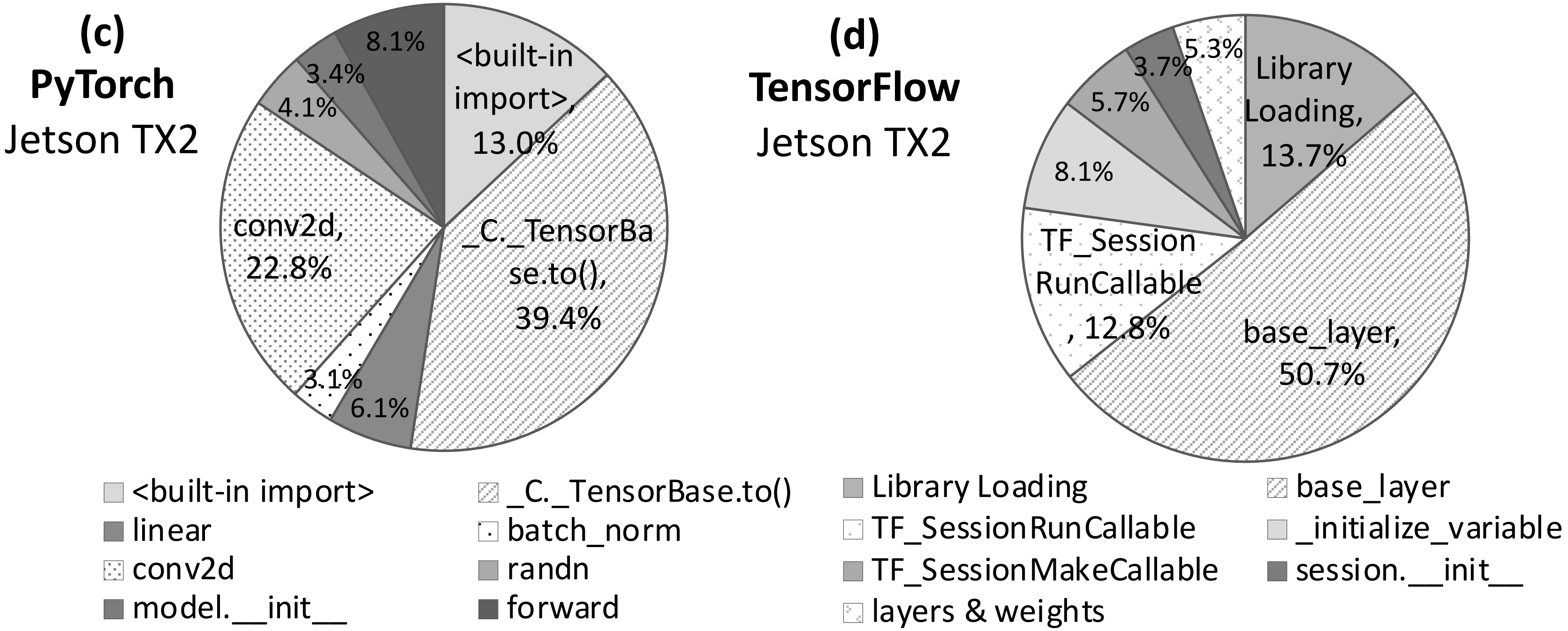
Time Profiling PyTorch and TensorFlow software stacks on **Raspberry Pi**



Georgia Tech comparch

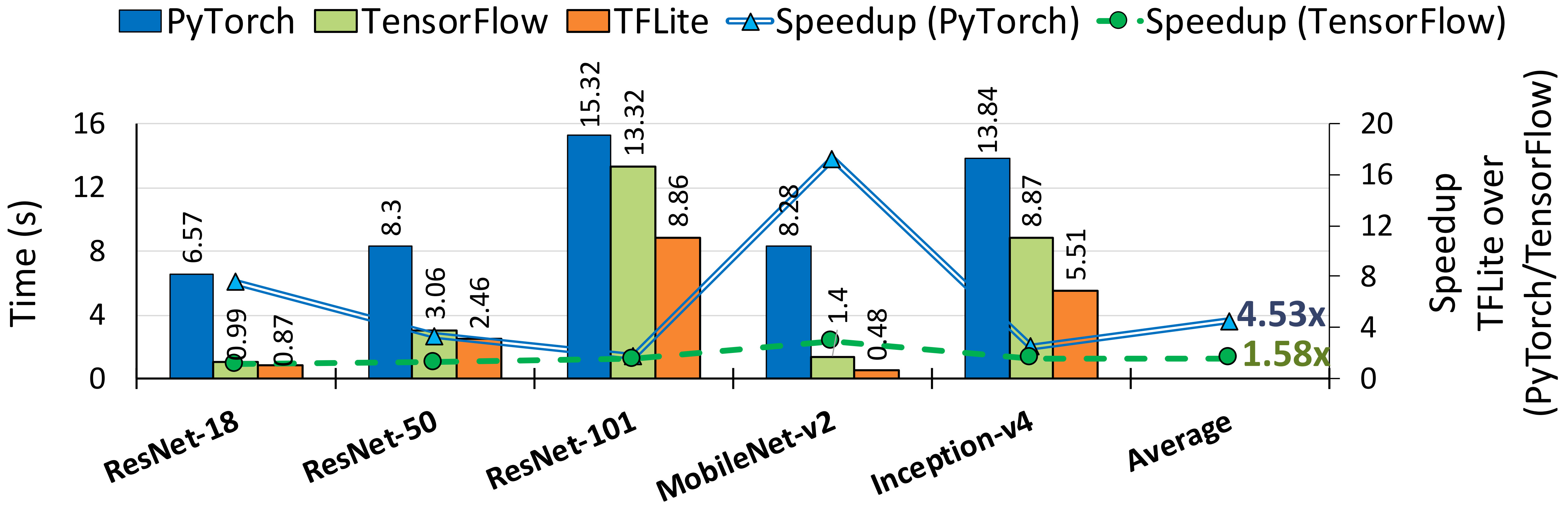
Software-Stack Analysis – TX2

Time Profiling PyTorch and TensorFlow software stacks on **Jetson TX2**



Edge-Specific Frameworks - RPi

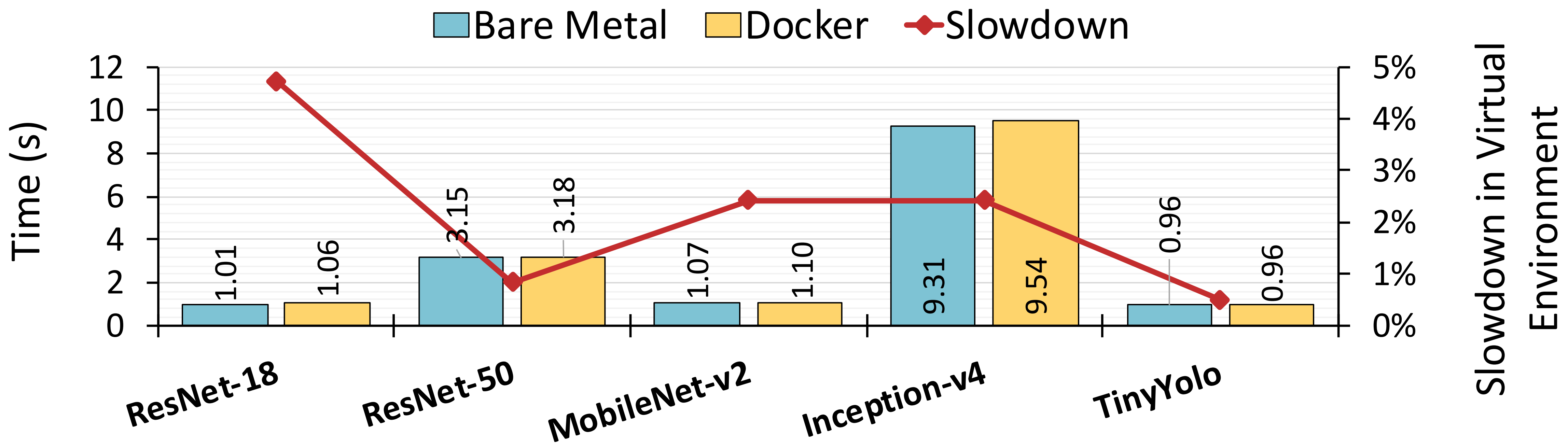
Time per inference on **RPi** with TensorFlow, PyTorch, and TFLite



Virtualization Overhead Study

Virtualization is a common solution for platform diversity.

Does it has performance impact? How much?



Temperature Measurements (I)

Measuring correlation between temperature and DNN execution.

DEVICE SPECIFICATIONS FOR TEMPERATURE EXPERIMENTS.

Device	Heatsink	Cooling Fan	Idle Temperature	Fan Activated?
Raspberry Pi	✗ 14x14 mm	✗	43.3 °C	✗
Jetson TX2	✓ 80x55x20 mm	✓	32.4 °C	✓
Jetson Nano	✓ 59x39x17 mm	✗	35.2 °C	✗
Edge TPU	✓ 44x40x9 mm	✓	33.9 °C	✗
Movidius	✓† 60x27x14 mm	✗	25.8 °C	✗

† USB stick is designed as a heatsink.

Temperature Measurements (II)

Measuring correlation between temperature and DNN execution.