

DeepEdge- Deep Neural Networks on Edge Devices

Presented by

Akanksha Raina

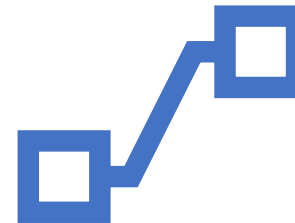
Srujana Malisetti

Rohit Singh

Nov 21, 2019

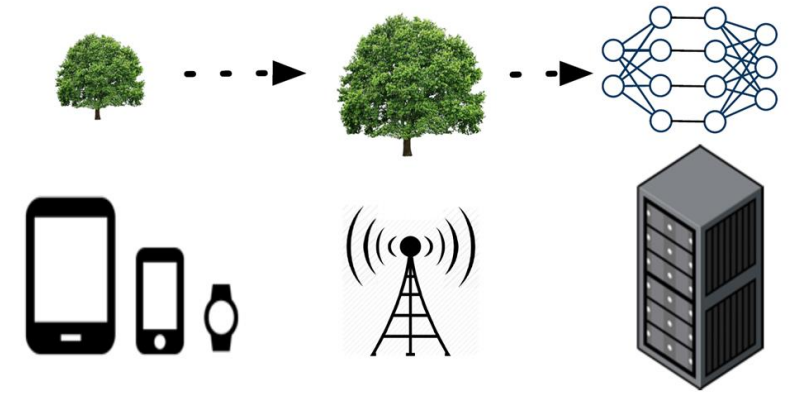
Instructor

Dr.In Kee Kim

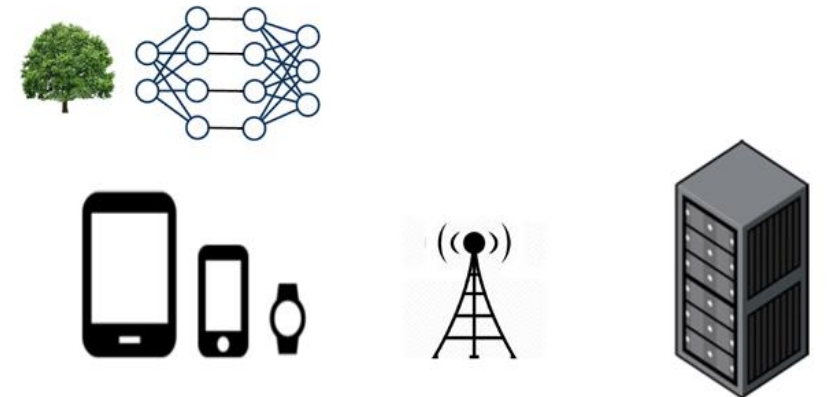


Motivation

- Many Cloud Providers now a days are providing Machine Learning Services termed as MLaaS.
- Intelligent Personal Assistants running on SoC integration devices, have capability to run ML Models efficiently.
- How about leveraging this capability on edge devices?



State-of-the-art approach

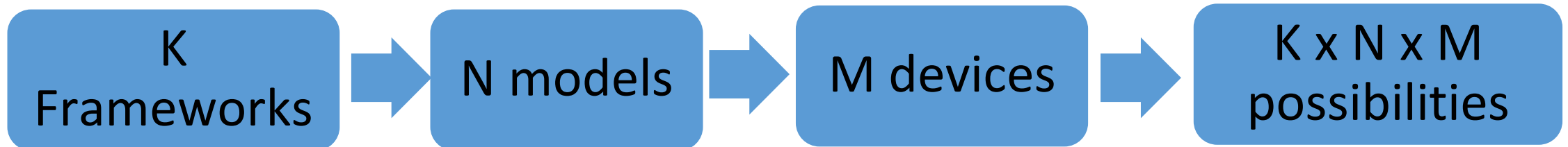


Proposed approach

Many Options



AlexNet VGG CaffeNet	Image Classification	Caffe	Apple Siri
			Microsoft Cortana
DeepFace FaceNet NormFace	Face Recognition	TensorFlow	Google Now
			Amazon Alexa
Kaldi DeepSpeech	Speech Recognition	Keras	Raspberry Pi
			Jetson Nano
SENNA Tesseract	Text Recognition	PyTorch	Cloud - VM, Container, Functions



Help from!!



- **pCAMP: Performance Comparison of Machine Learning Packages on the Edges**

<https://www.usenix.org/system/files/conference/hotedge18/hotedge18-papers-zhang.pdf>

- **Distributed Perception by Collaborative Robots**

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8411096>

- **Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge**

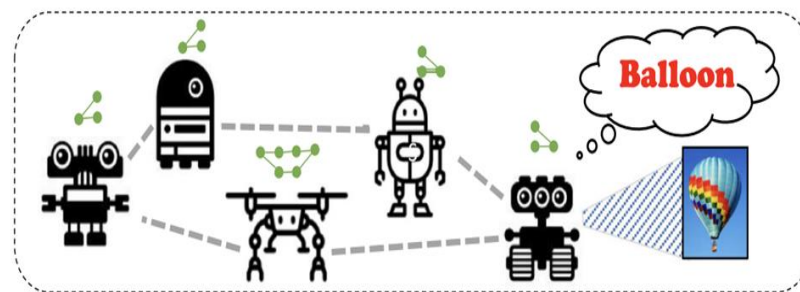
<http://web.eecs.umich.edu/~jahausw/publications/kang2017neurosurgeon.pdf>

	MacBook Pro	Intel FogNode	NVIDIA Jetson TX2	Raspberry Pi	Nexus 6P
TensorFlow	✓	✓	✓	✓	×
Caffe2	✓	✓	✓	×	×
MXNet	✓	✓	×	×	×
PyTorch	✓	✓	✓	×	×
TensorFlow Lite	×	×	×	×	✓



Computation Domain

(a)



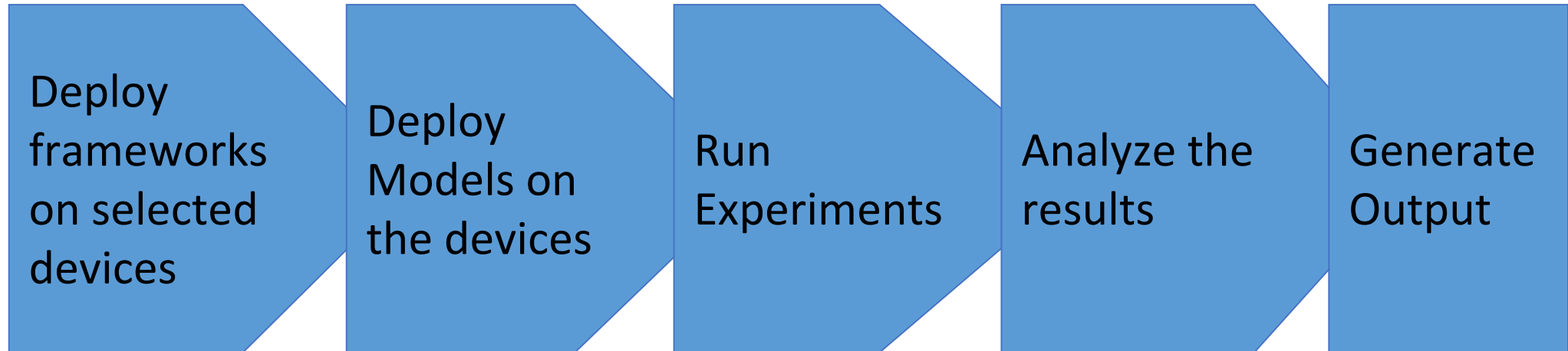
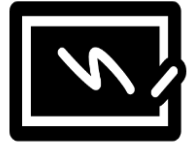
Computation Domain

(b)

Across 8 benchmarks		Average	Maximum
Latency	↗	3.1x	40.7x
Mobile energy Consumption	↻	59.5%	94.7%
Datacenter Throughput	↗	1.5x	6.7x

Putting together..

Approach



Measurement



- On each device,
 - Accuracy
 - Processing Time
 - CPU Usage
 - Memory usage
 - Battery consumption on Mobile.

Evaluation SetUp

Device	Specifications	ML Framework	Model
Atomic Pi	Intel Atom x5-8350 quad core with 2M cache 2GB RAM. Ubuntu 18.0.	Tensorflow 1.5	MobileNet_v1_224
Android (Samsung C9Pro)	Android version 8.0.0 Octa-Core 4x1.95 GHz ARM Cortex-A72 + 4x1.44 GHz ARM Cortex-A53 RAM- 6GB 4,000 mAh Battery Capacity	TensorflowLite	MobileNet_v1_224_quant
AWS EC2 Instance	p2.xlarge 4vCPUs, 61GB RAM AMI – DeepLearning , Ubuntu 18.04 V25.3	Tensorflow1.14	MobileNet_v1_224
AWS Docker	p2.xlarge 4vCPUs, 61GB RAM AMI – DeepLearning , Ubuntu 18.04 V20.0 NVIDIA Docker, 1GPU	Caffe1.0	BVLC_Alexnet
Laptop	Intel(R) Core(™) i-7 8750H CPU@2.20GHZ RAM 16GB(15.2 GB usable)	Tensorflow 1.5	SSD_MobileNet_v1

Methodology

Development:

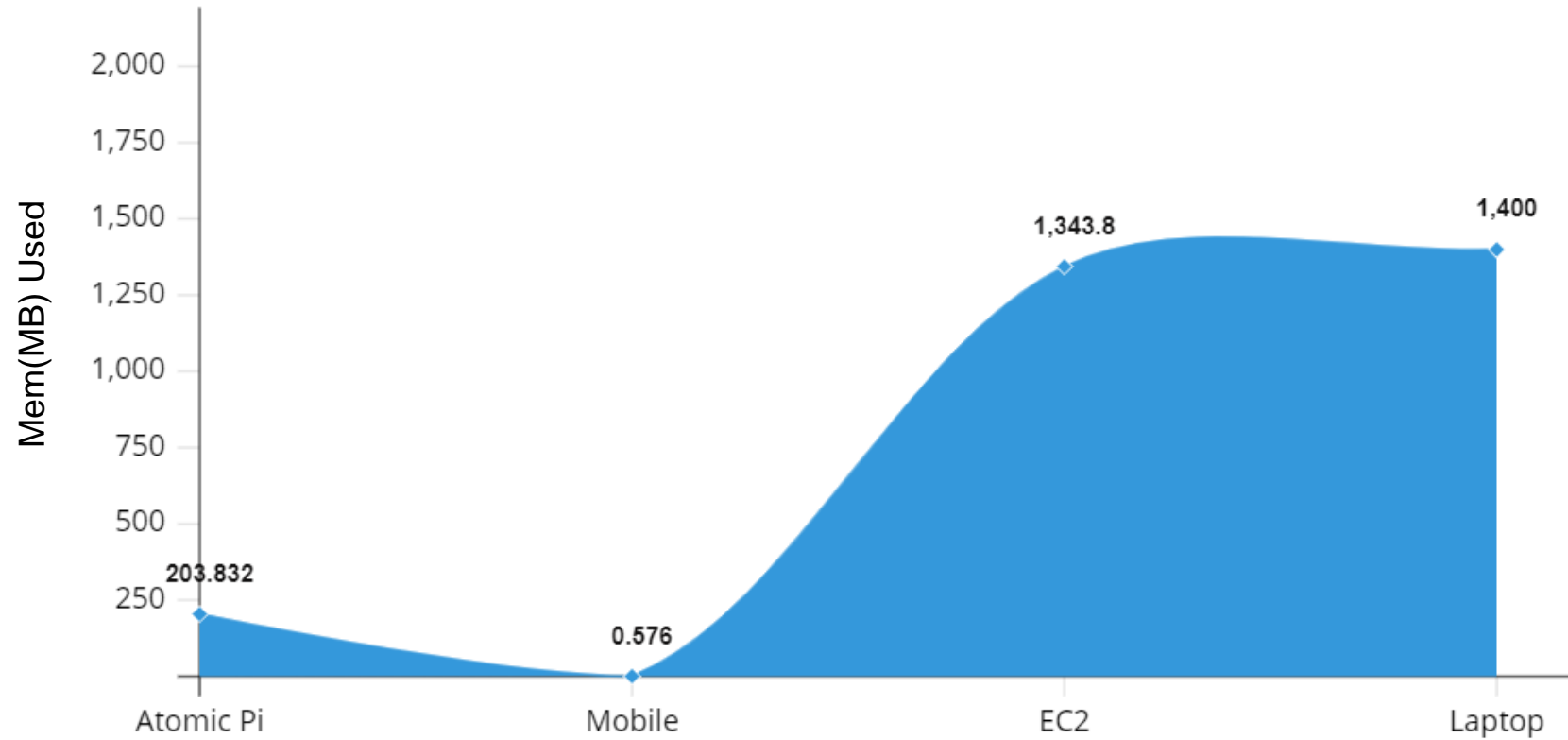
- For Mobile, Android studio to create Java app.
- For Atomic Pi, Python script.
- For EC2 instance & Docker, Jupyter notebook (Python).
- For Laptop Python, Jupyter notebook.

Measurement:

- Battery consumption on mobile device, GSam Battery Monitor Application with access to BATTERY_STAT is used. (0.2%)
- For CPU & Memory Usage, Inference Time measurement in Atomic Pi, Docker, EC2 Instance **top** command is used.
- For Laptop CPU, Memory Usage, GPU Usage task manager is used.

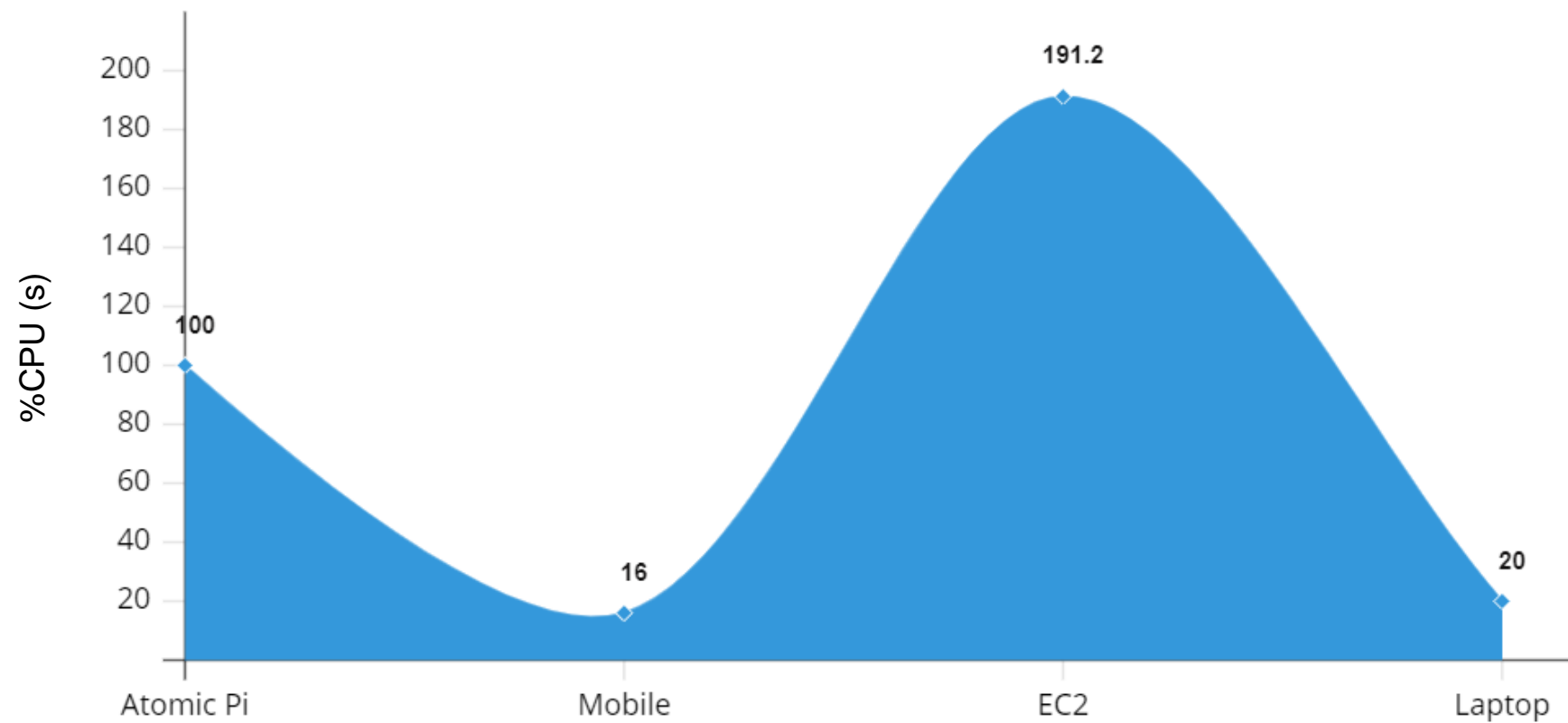
Memory Usage

TensorFlow+MobileNet_v1_224
TensorFlow Lite + MobileNet_v1_224_quant



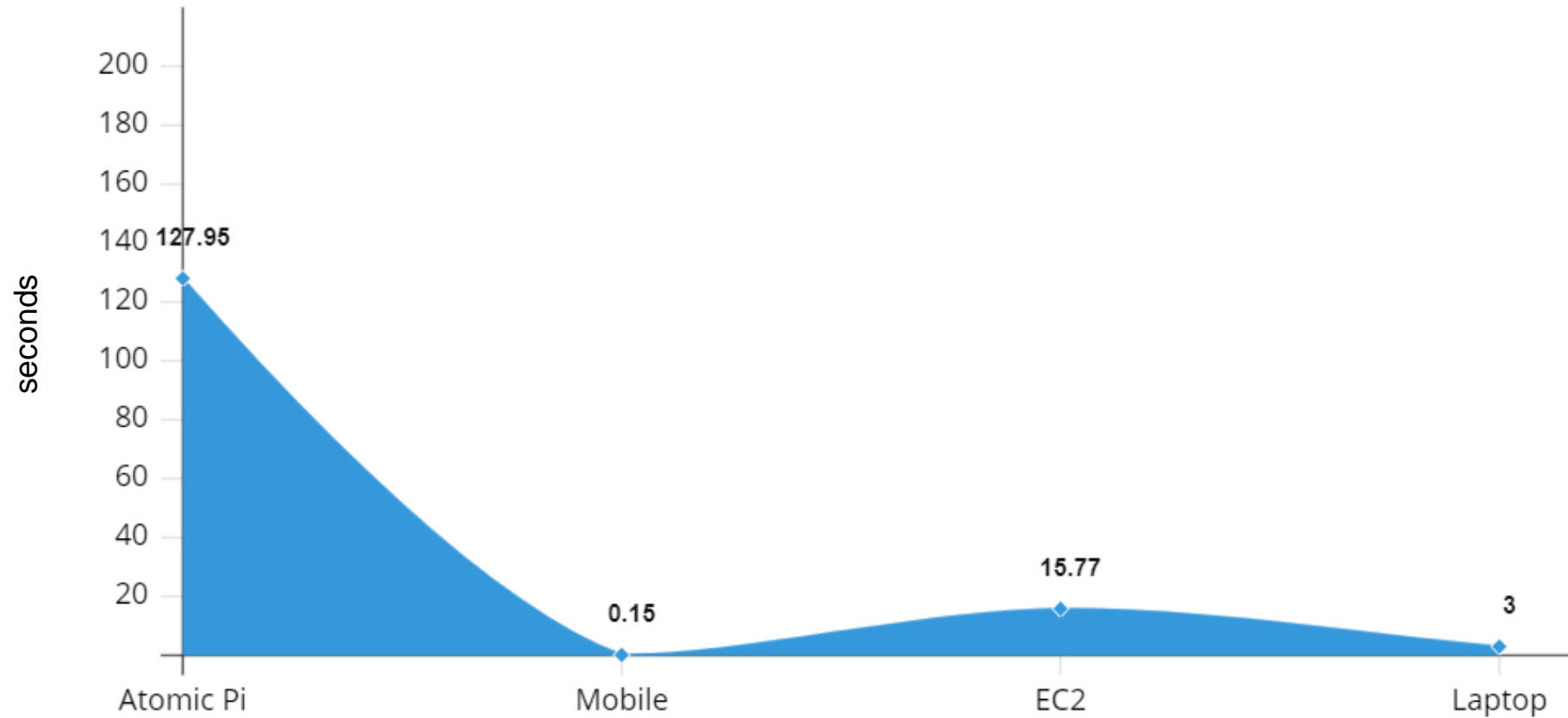
TensorFlow+MobileNet_v1_224
TensorFlow Lite + MobileNet_v1_224_quant

CPU Usage



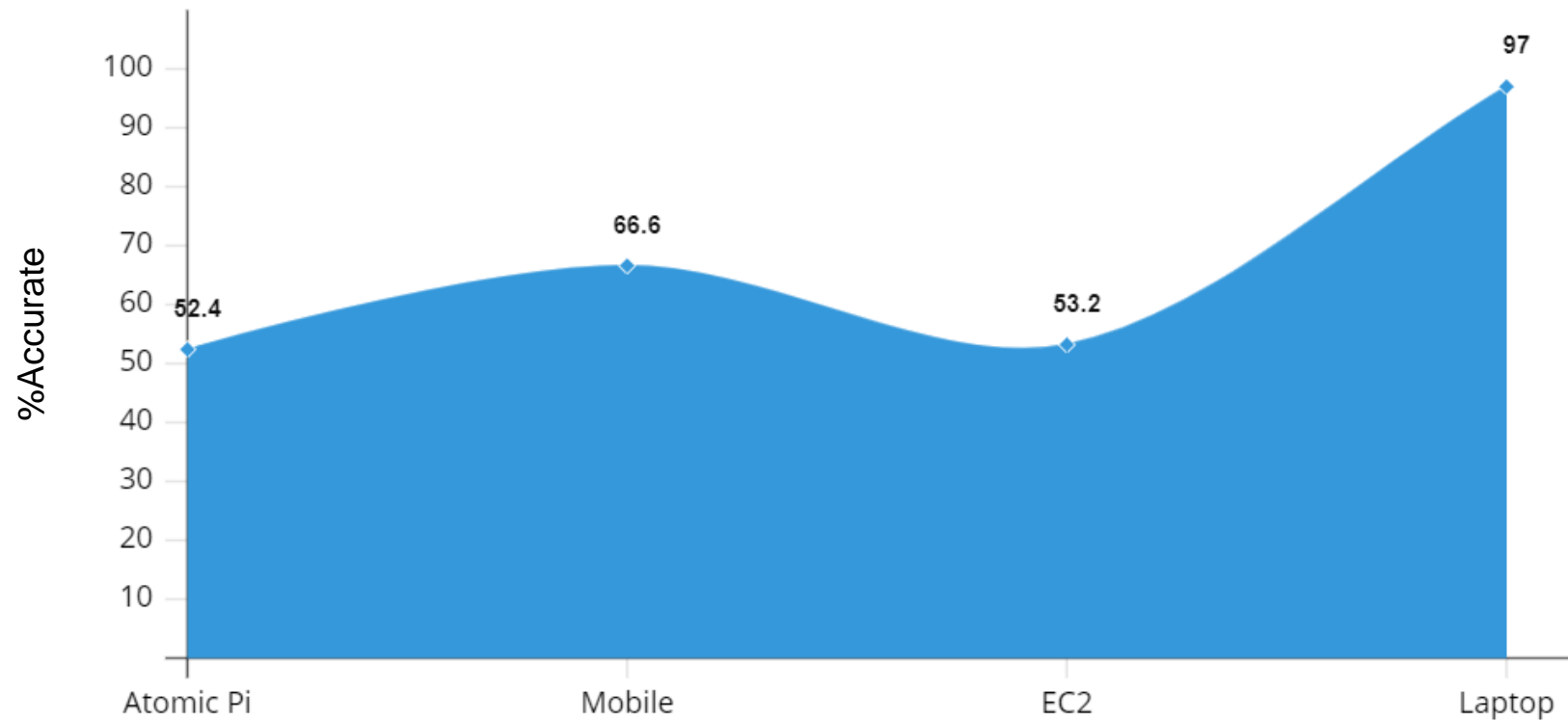
Processing Time

TensorFlow+MobileNet_v1_224
TensorFlow Lite + MobileNet_v1_224_quant

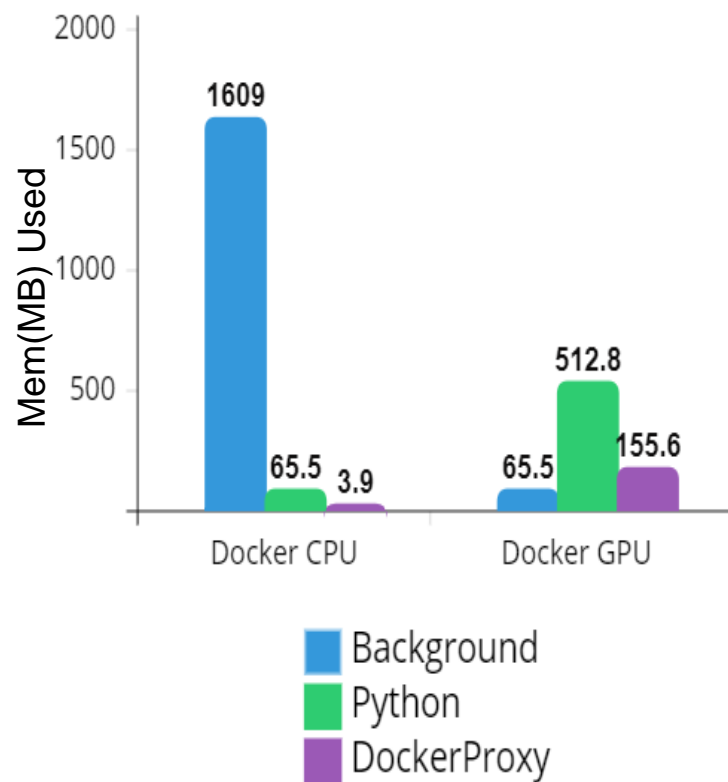


Accuracy

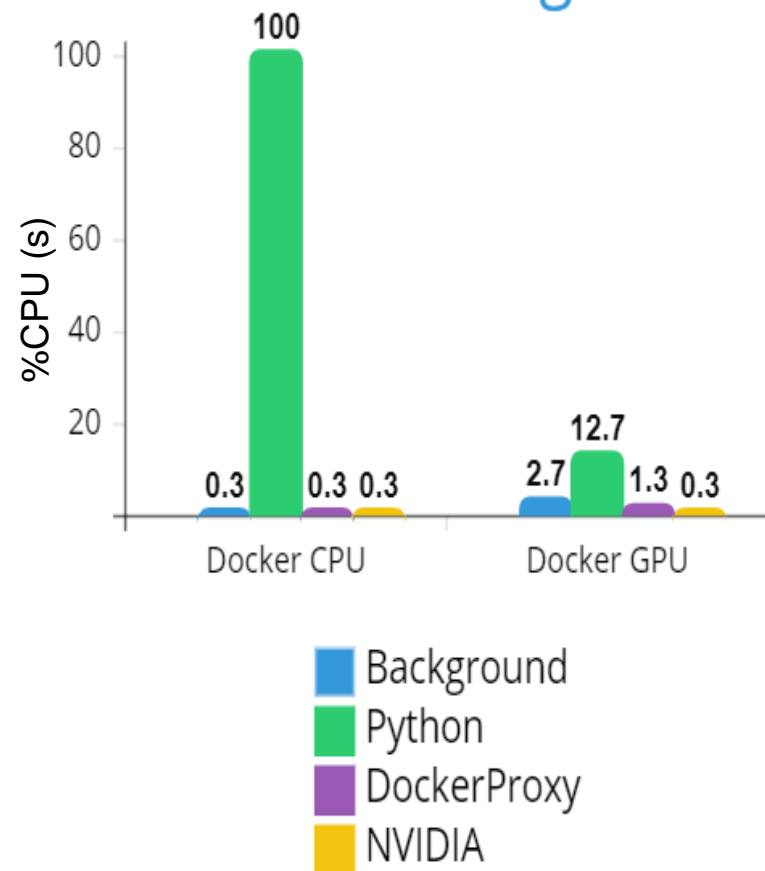
TensorFlow+MobileNet_v1_224
TensorFlow Lite + MobileNet_v1_224_quant



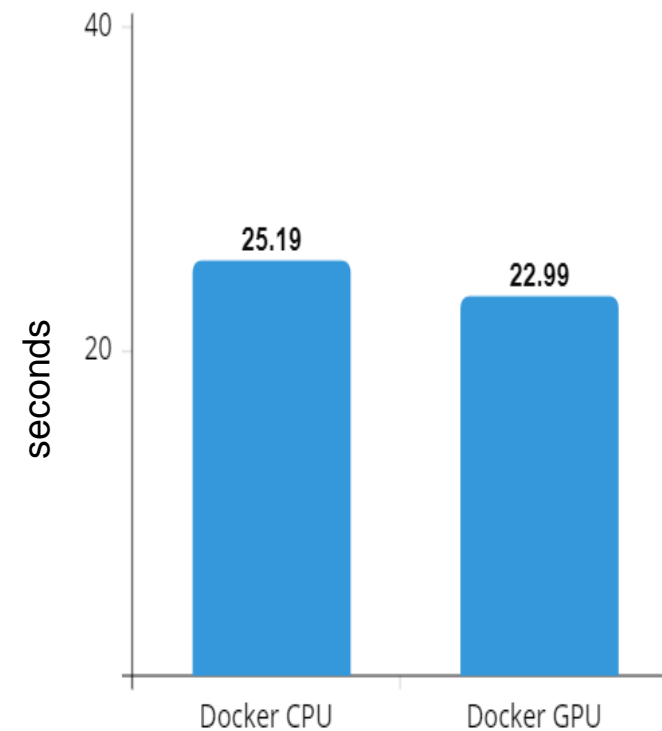
Memory Usage



CPU Usage



Processing Time



Lessons Learnt

- Caffe with Alexnet, Reference_caffenet are not deployable on all devices.
- TensorFlow framework with Mobilenet is easily deployable.
- Not many tools for measuring battery consumption on mobile devices.
- Failures:
 - Caffe with Caffenet on Android, AtomicPi & Laptop.
 - Caffe with Alexnet on Android, Atomic Pi & Laptop.
 - AtomicPi did not support newer version of Tensorflow.
 - Jupyter Notebook hangs AtomicPi.

Conclusion



Surprisingly, EC2 instance is consuming more resources.



AtomicPi has taken time to provide the result.



Tensorflow Lite with MobileNet combination is efficient and equally accurate even though it is quantized to support mobile devices.



Future work involves more devices and more frameworks.

Extra Work

Laptop – TensorFlow with RCNN

- Accuracy – 99%
- CPU Usage – 42%
- Memory Usage- 3 GB
- Total Time(Inference + Processing) – 180 seconds.

Intel tag node x Instances EC2 x research/slimy x Mobilenet1 x +

localhost:8888/notebooks/research/slimy/Mobilenet1.ipynb ☆ 🔒 🔒 🔒

Apps Google High Cardio Worko... TRIP How to Draw Mabe... Other Bookmarks

jupyter Mobilenet1 Logout


Trusted Environment (conda_tensorflow_p36) ○

File Edit View Insert Cell Kernel Widgets Help

Run C Code

```
label_map = imagenet.create_readable_names_for_imagenet_labels()
display.display(PIL.Image.open('/home/ubuntu/image.jpeg'))

print("Top 1 Prediction: ", x.argmax(),label_map[x.argmax()], x.max())
```



top - 17:26:37 up 11 min, 2 users, load average: 0.00, 0.02, 0.00
Tasks: 124 total, 1 running, 73 sleeping, 0 stopped, 0 zombie
N/Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 42673196 total, 61807352 free, 432408 used, 633436 buff/cache
KiB Swap: 0 total, 0 free, 0 used, 61840956 avail Mem

PID	USER	PR	NI	VI	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2055	ubuntu	20	0	44584	4160	3472	R	0.3	0.0	0:00.36	top
1	root	20	0	225272	9816	6740	S	0.0	0.0	0:03.23	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H
5	root	20	0	0	0	0	I	0.0	0.0	0:00.02	kworker/u32:0
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0.0	0.0	0:00.01	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:00.06	rcu_sched
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/1
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
16	root	20	0	0	0	0	S	0.0	0.0	0:00.02	ksoftirqd/1
18	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/2
20	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/2
21	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/2
22	root	20	0	0	0	0	S	0.0	0.0	0:00.01	ksoftirqd/2
23	root	20	0	0	0	0	I	0.0	0.0	0:00.02	kworker/2:0
24	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/2:0H
25	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/3
26	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/3
27	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/3
28	root	20	0	0	0	0	S	0.0	0.0	0:00.01	ksoftirqd/3
30	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/3:0H
31	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
32	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
33	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_kthre
34	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
35	root	20	0	0	0	0	S	0.0	0.0	0:00.00	xenbus
36	root	20	0	0	0	0	S	0.0	0.0	0:00.02	xenwatch
37	root	20	0	0	0	0	I	0.0	0.0	0:00.03	kworker/0:1
39	root	20	0	0	0	0	I	0.0	0.0	0:00.04	kworker/3:1
40	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd



Thank you

Questions?