DeepEdge- Deep Neural Networks on Edge Devices

Presented by

Akanksha Raina

Srujana Malisetti

Rohit Singh

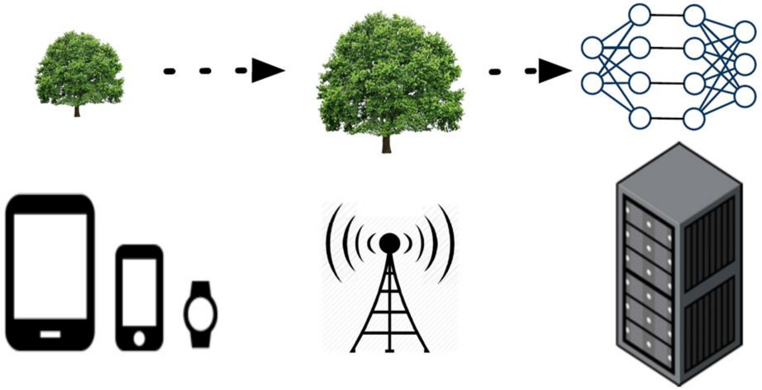
Nov 21, 2019

Instructor

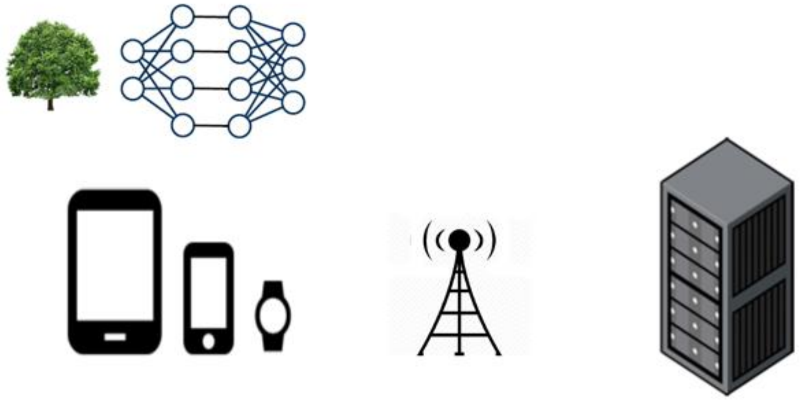
Dr.In Kee Kim

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
DeepFace FaceNet NormFace	Face Recognition
Kaldi DeepSpeech	Speech Recognition
SENNA Tesseract	Text Recognition

Caffe
TensorFlow
Keras
PyTorch

Apple Siri
Microsoft Cortana
Google Now
Amazon Alexa
Raspberry Pi
Jetson Nano
Cloud - VM, Container, Functions

K Frameworks → N models → M devices → K x N x M possibilities

Help from!!

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

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

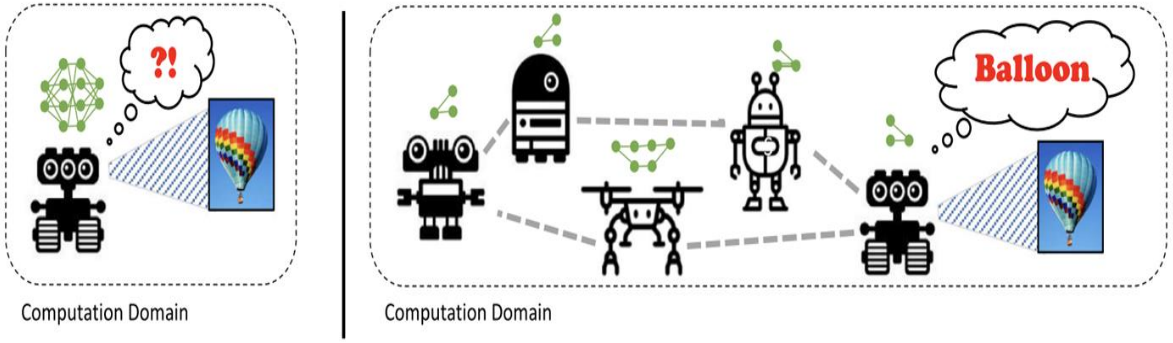
- **Distributed Perception by Collaborative Robots**

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	×	×	×	×	√

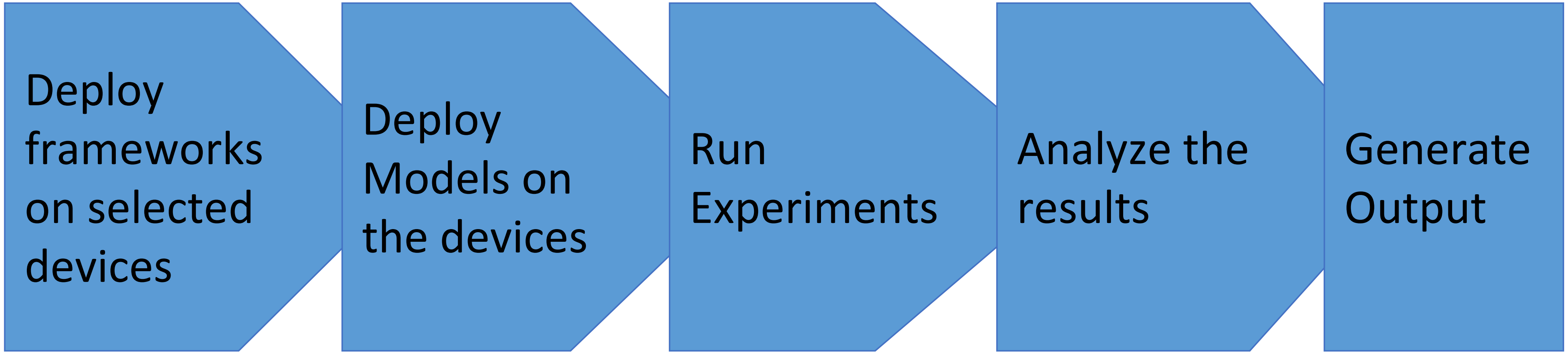


(a) (b)

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



Deploy frameworks on selected devices
Deploy Models on the devices
Run Experiments
Analyze the results
Generate Output

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 4×1.95 GHz ARM Cortex-A72 + 4×1.44 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:

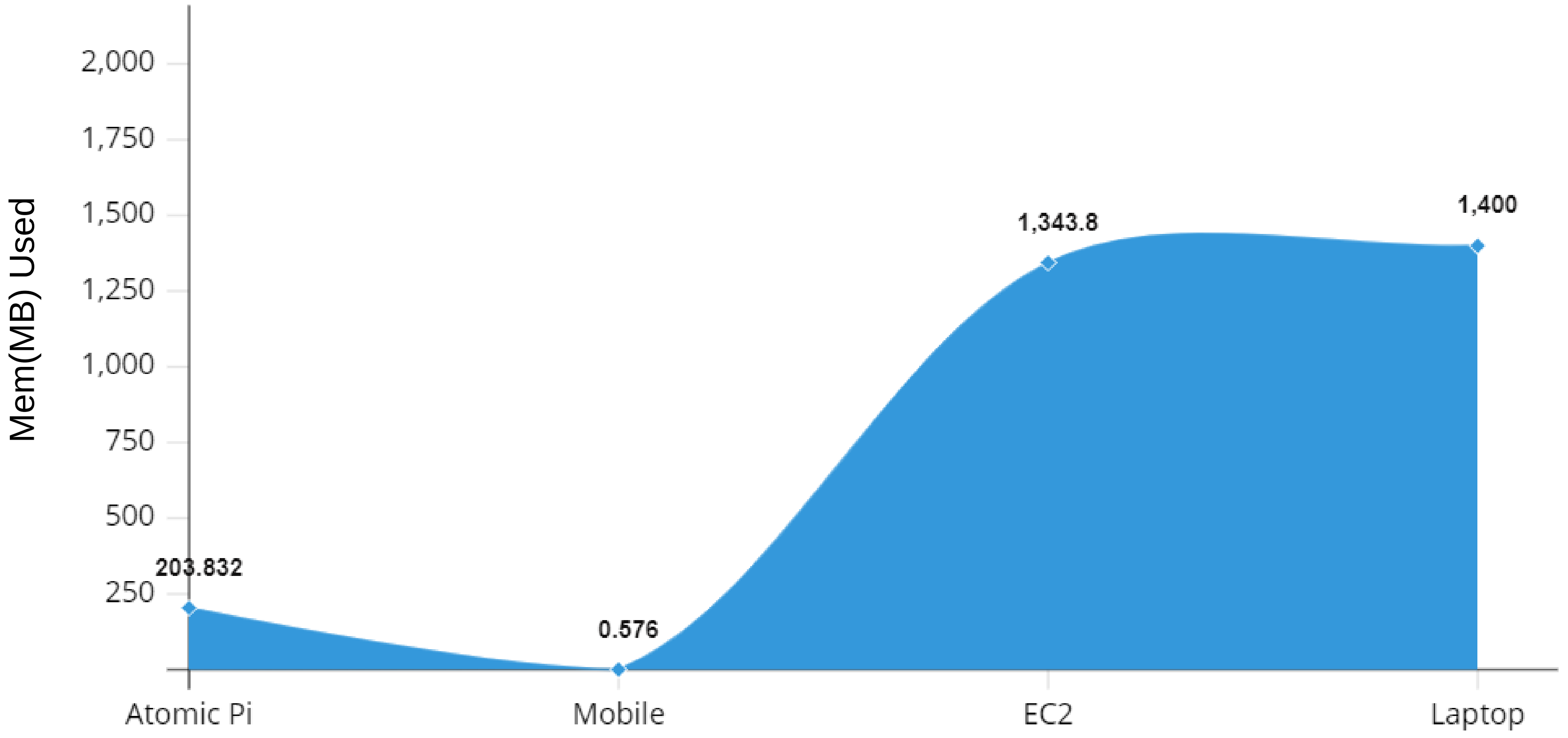
- For Mobile, Android studio to create Java app.
- For Atomic Pi, Python script.
- For EC2 instance & Docker, Jupyter notebook (Python).
- For Laptop Python, Jupyter notebook.

Measurement:

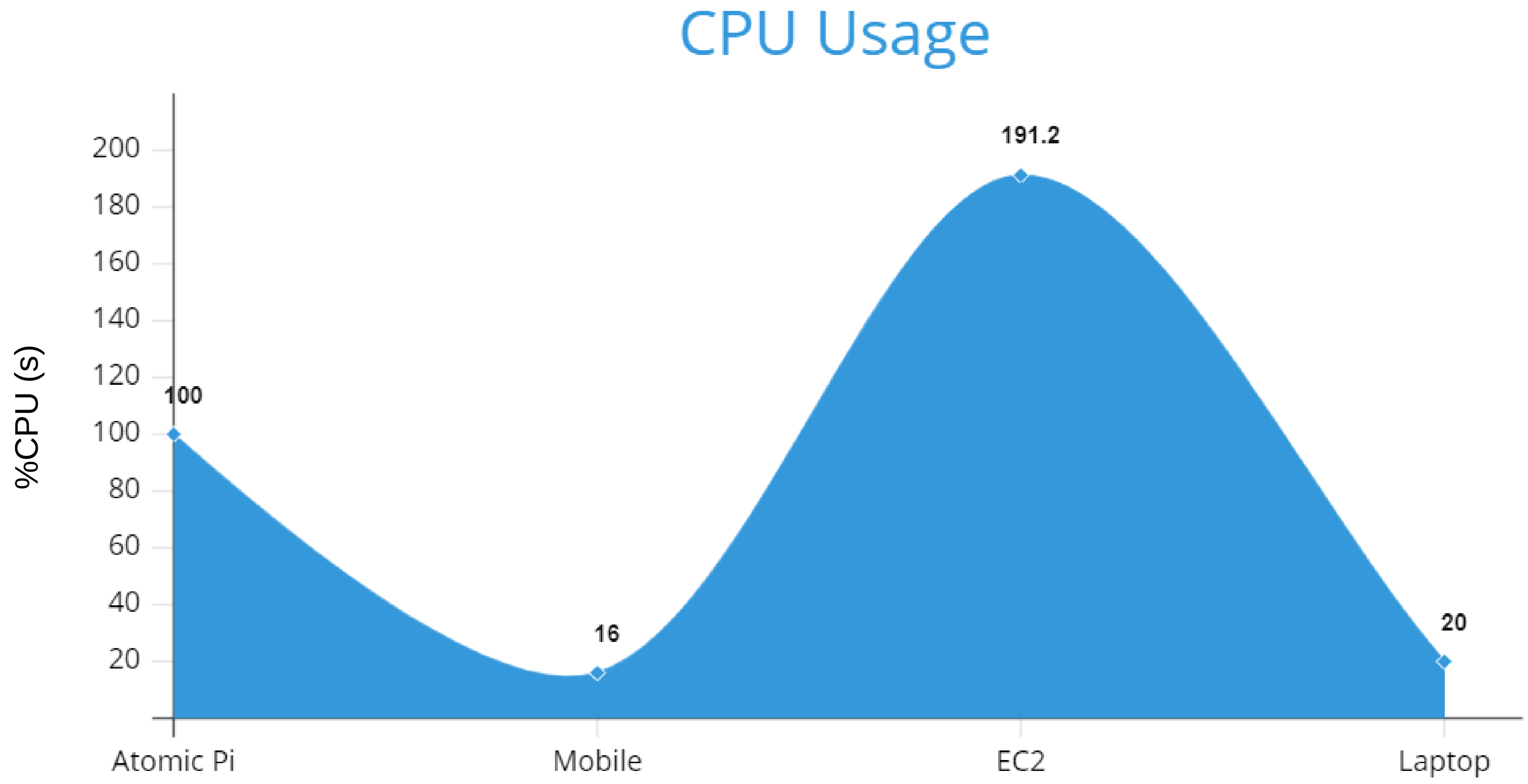
- Battery consumption on mobile device, GSam Battery Monitor Application with access to BATTERY_STAT is used. (0.2%)
- For CPU & Memory Usage, Inference Time measurement in Atomic Pi, Docker, EC2 Instance ***top*** command is used.
- For Laptop CPU, Memory Usage, GPU Usage task manager is used.

Memory Usage

TensorFlow+MobileNet_v1_224
TensorFlow Lite + MobileNet_v1_224_quant

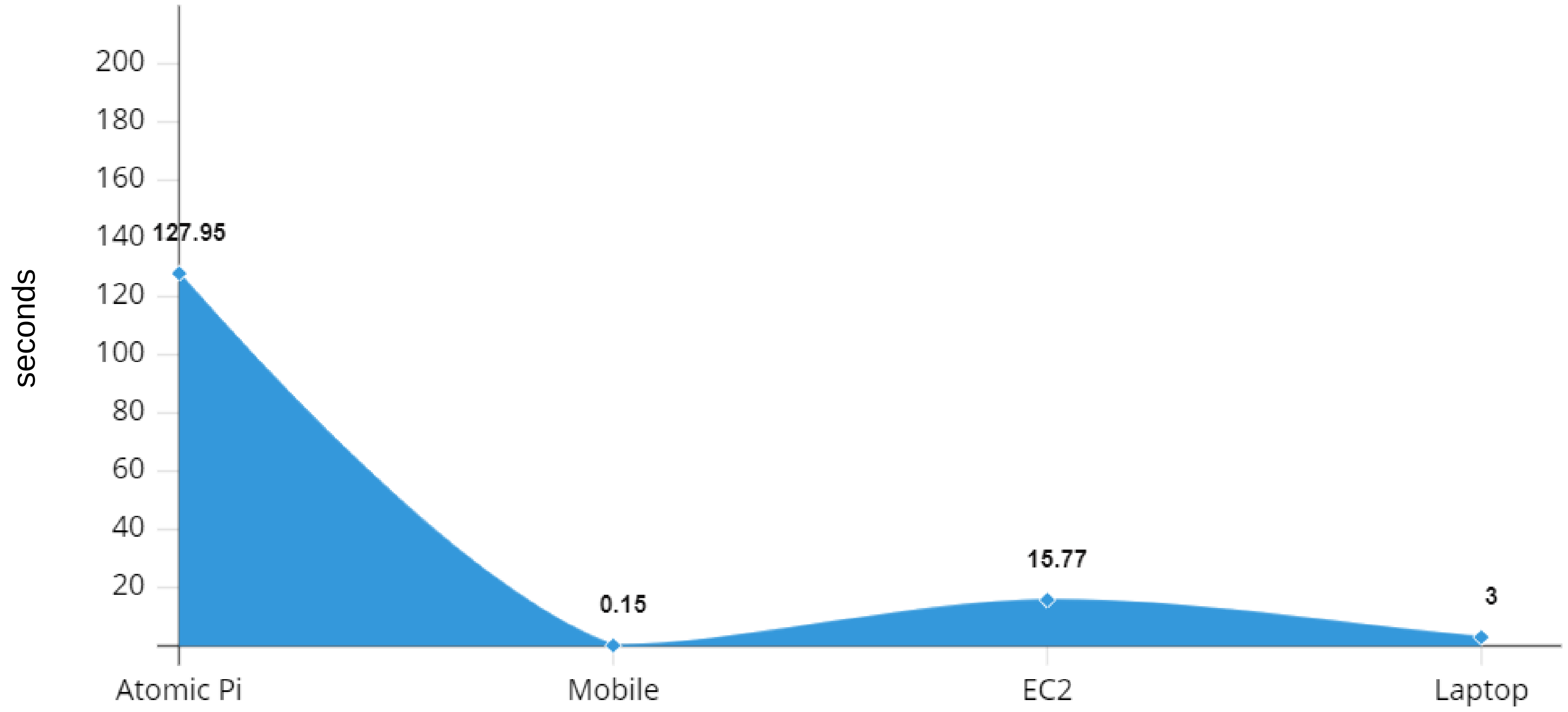


TensorFlow+MobileNet_v1_224
TensorFlow Lite + MobileNet_v1_224_quant



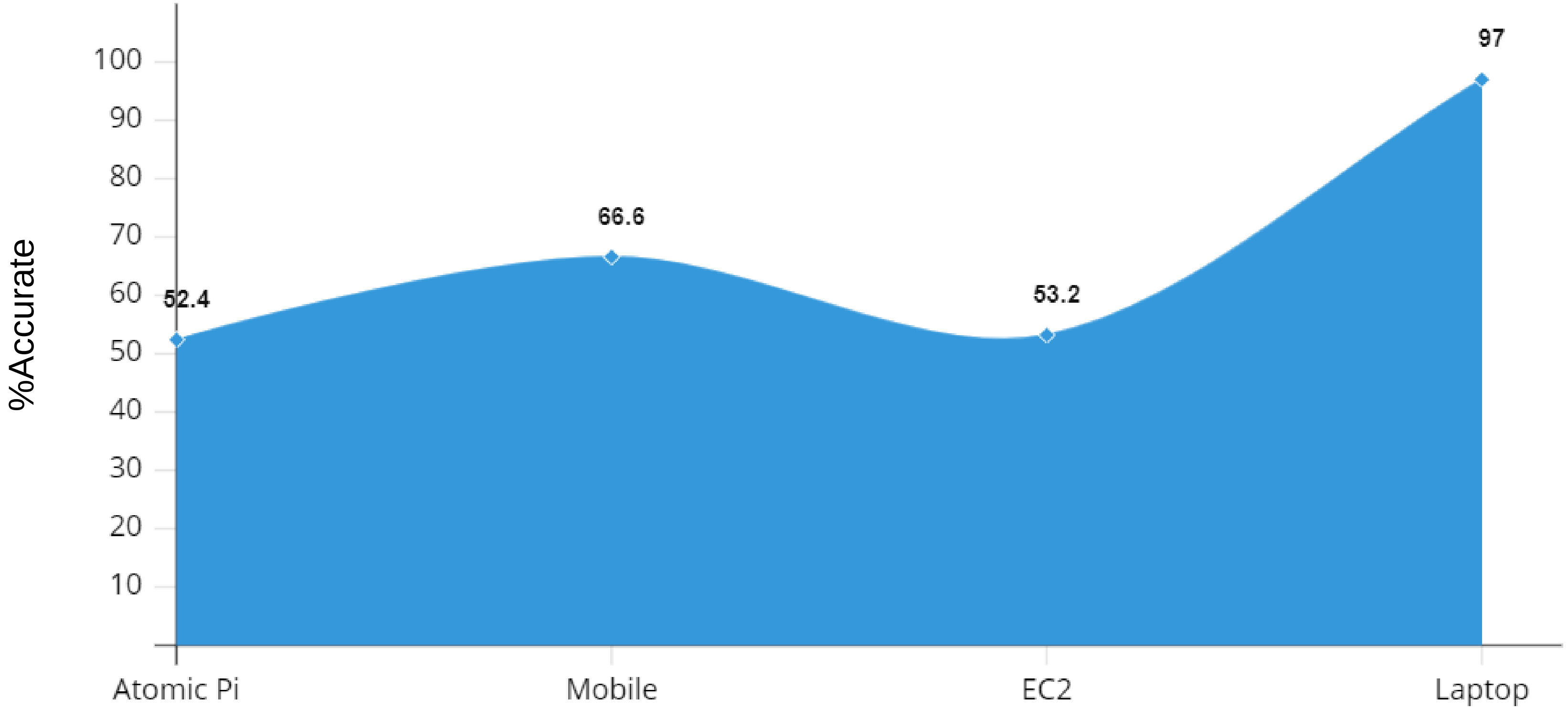
Processing Time

TensorFlow+MobileNet_v1_224
TensorFlow Lite + MobileNet_v1_224_quant

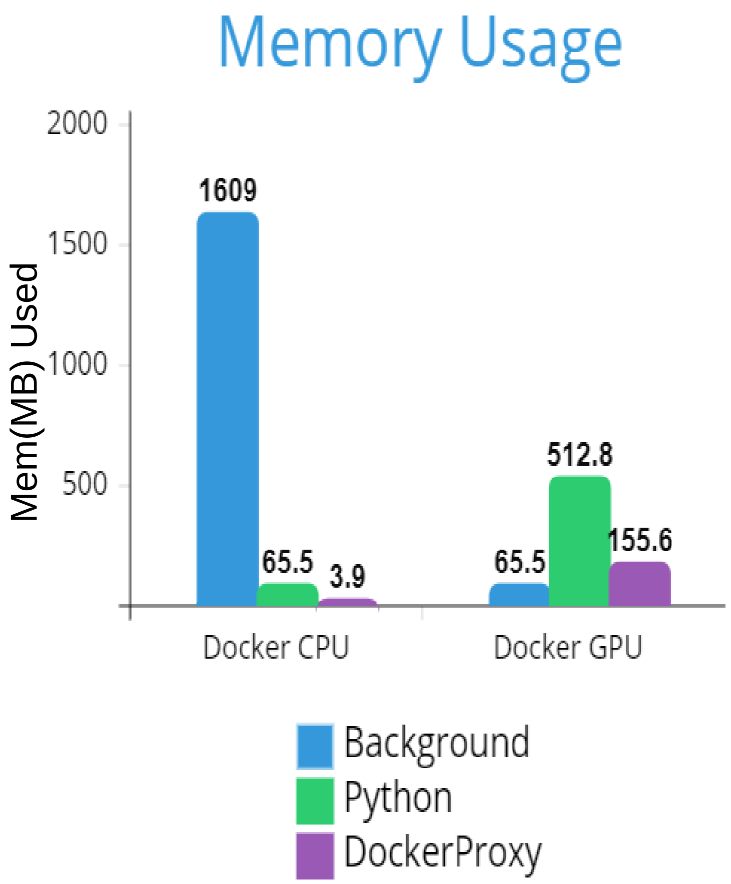


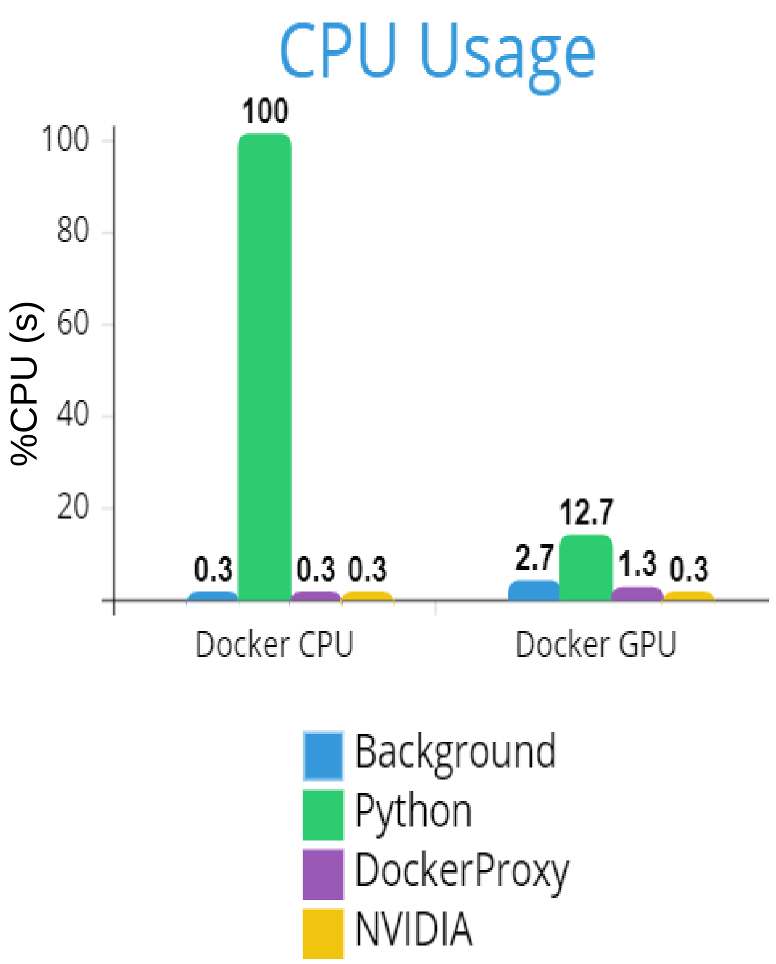
Accuracy

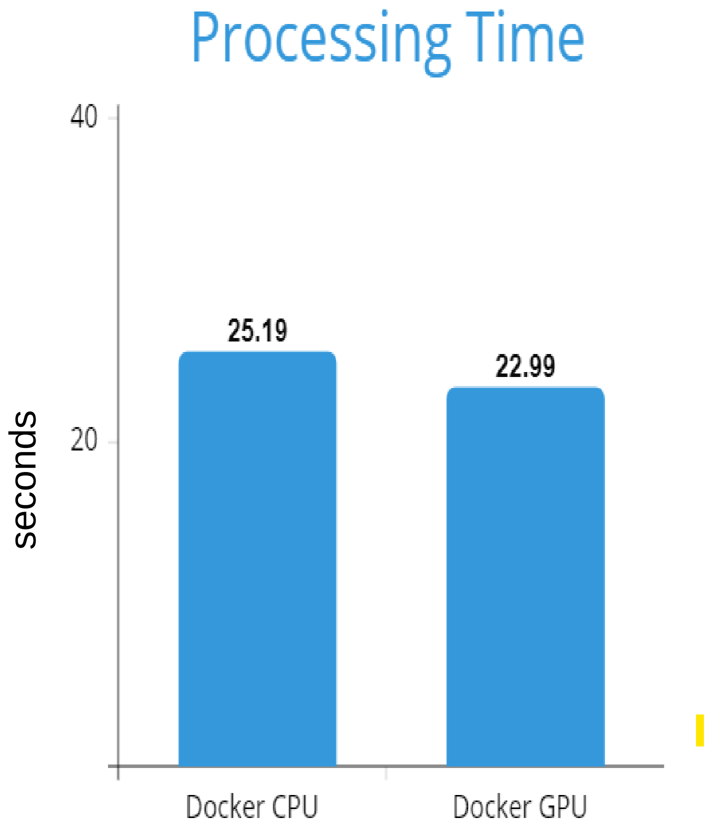
TensorFlow+MobileNet_v1_224
TensorFlow Lite + MobileNet_v1_224_quant



Caffe1.0 + AlexNet







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.

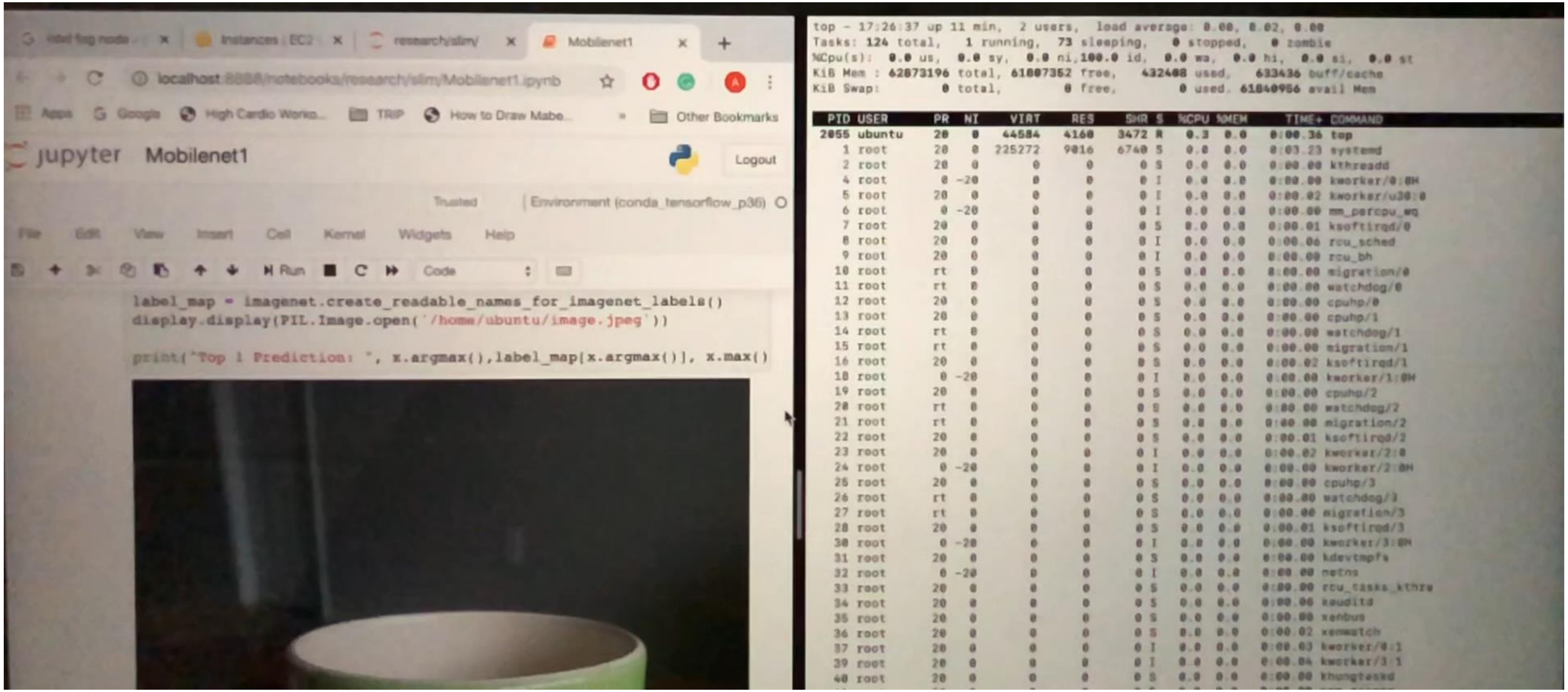
Conclusion

- 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.


jupyter
Mobilenet1
Logout
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()
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

Thank you

Questions?