Real-Time Image Recognition Using Collaborative IoT Devices

Ramyad Hadidi*, Jiashen Cao*, Matthew Woodward*, Michael S. Ryoo**, and Hyesoon Kim*

*Georgia Institute of Technology

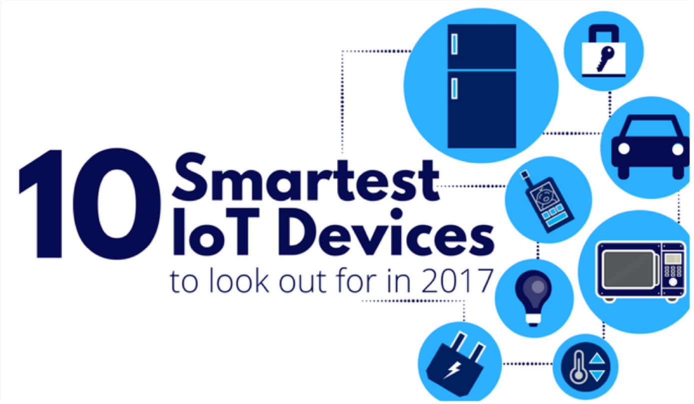
**Indiana University; EgoVid Inc.

Georgia Tech comparch

Prevalence of IoT Devices

Internet of Things (IoT) devices are everywhere

- Smart Locks, Smart Sprinklers, Smart Plugs, Smart Baby Monitors, Smart Cookers, Smart Thermostats, Smart Mirrors, Smart Cleaners, and Smart Refrigerators



Many of which generate/capture abundance of real-time raw data such as images.

https://www.pentasecurity.com/blog/10-smartest-iot-devices-2017/

How to Process IoT data?

Advancements of deep neural networks (DNN) provides many high-accuracy solutions to previously impossible tasks:

- Image Recognition
- Face Recognition
- Video (Action Recognition)
- Voice Recognition

Performing these tasks in real-time requires high computational power.

Where to Process (I)

- **(Option A)** Use the individual IoT device
  - Limited energy (e.g., battery powered)
  - Limited compute power
  - So, unable to meet time constrains
- **(Option B)** Offload to Cloud
  - Such as Voice recognition service of Apple’s Siri, Amazon’s Echo, Microsoft’s Cortana, and Google Home
  - Any problem?

Where to Process (II)

(Option B) Cloud processing is promising but:

- Not Scalable
  - More traffic, data, and storage
  - IoT devices outnumbered world population in 2017
- Privacy and Security
  - Voice recognition? Big Brother's spying devices in the novel 1984
  - Multiple layers: Network security, encryption, and etc.
- Quality of Service (QoS) and Reliability
  - We have a tight timing constraint for real-time recognition

F.Biscotti et al.,"The Impact of the Internet of Things on Data Centers," Gartner Research, vol. 18, 2014.

Where to Process (III)

(Option C) What if we could harvest the aggregated computational power of local IoT devices?

- At a given time, not all devices are fully utilized

Collaborative IoT Devices

(Option C) We study such collaboration between IoT devices in our paper, *Musical Chair*.

- Our performance metric: Inferences per second
- We use same models, so we have same accuracy

In this work, we showcase the application of Musical Chair for Image recognition models on a farm of Raspberry PIs

Hadidi et al. "Musical Chair: Efficient Real-Time Recognition Using Collaborative IoT Devices." *arXiv preprint arXiv:1802.02138* (2018).

Outline

Motivation

Musical Chair

- Data and Model Parallelism

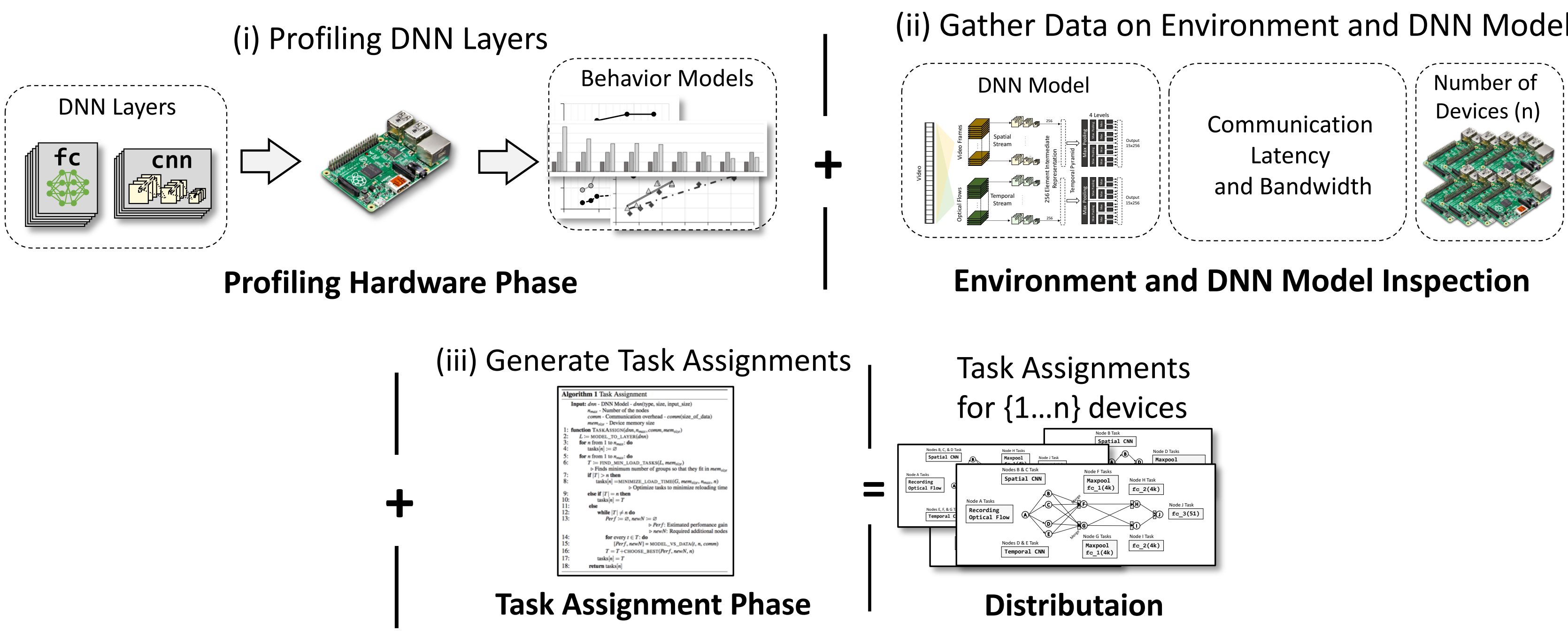
Hardware and Software Overview

System Evaluations

Conclusion

Musical Chair

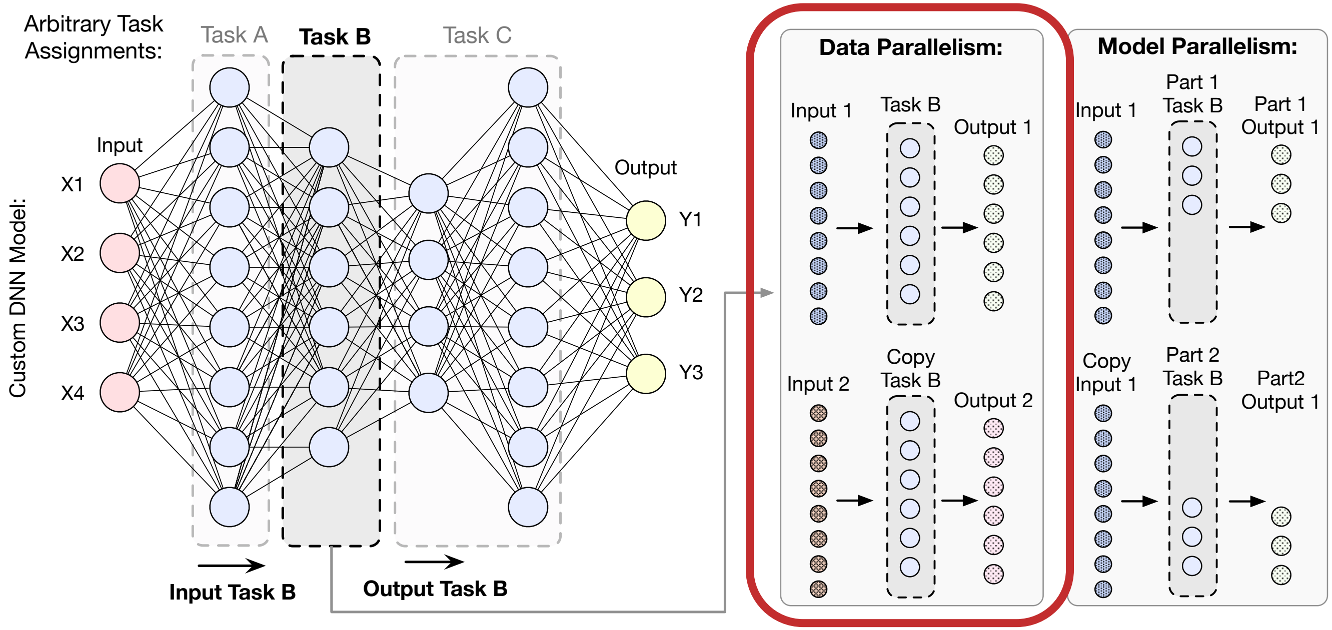
Musical Chair is a technique for distributing DNN computations over multiple IoT devices.



Hadidi et al. "Musical Chair: Efficient Real-Time Recognition Using Collaborative IoT Devices." *arXiv preprint arXiv:1802.02138* (2018).

Model & Data Parallelism

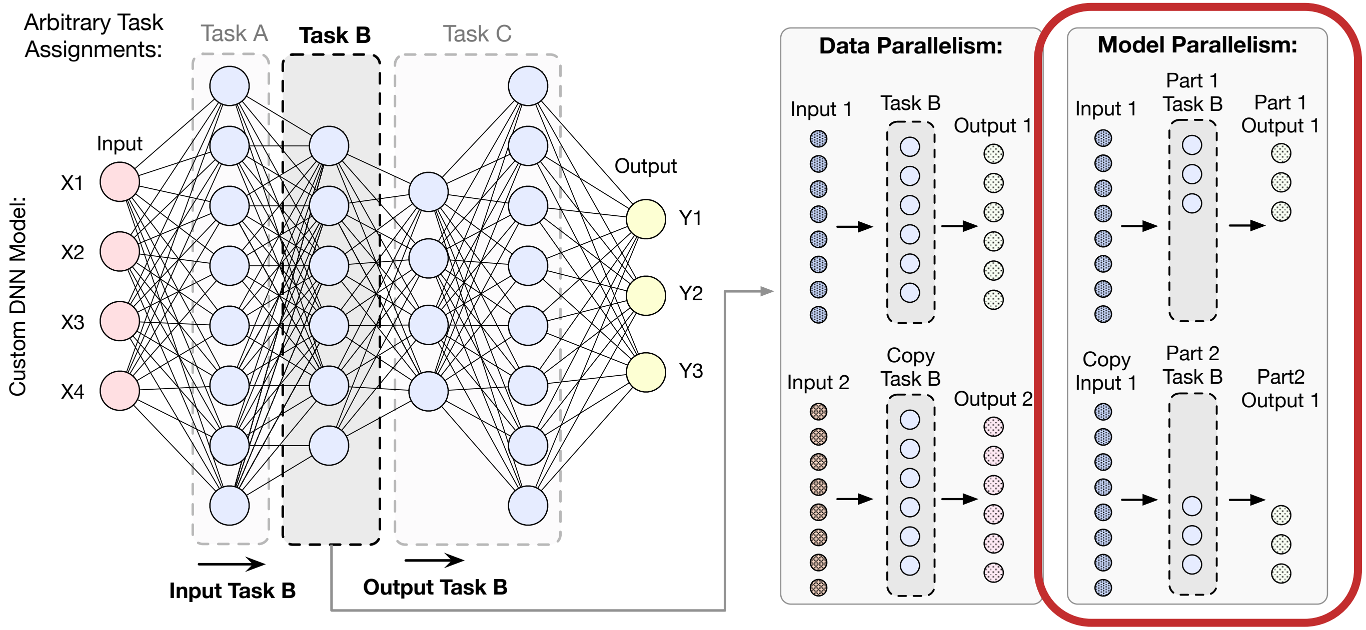
Two forms of distribution:



Data parallelism is providing the next input to multiple devices in a network.

Model & Data Parallelism

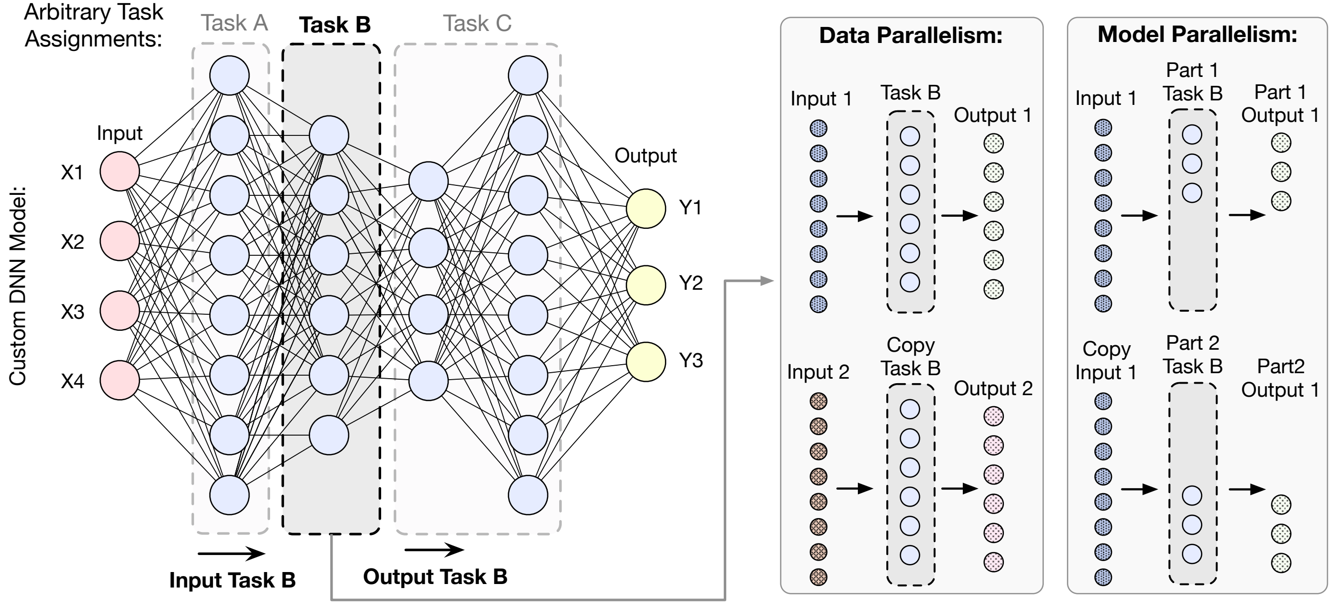
Two forms of distribution:



Model parallelism is splitting parts of a given layer or group of layers over multiple devices.

Model & Data Parallelism

Two forms of distribution:



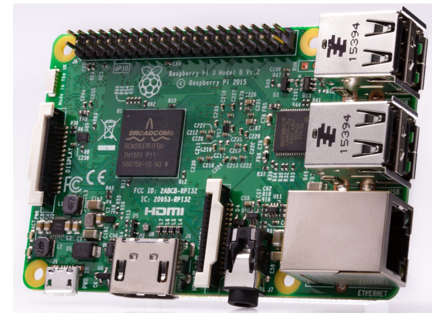
Convolution Layers: Mostly data parallelism

Fully Connected Layers: Either data or model parallelism depending on size of the layer, input, and memory

Hardware Overview

Raspberry PI 3:

- Cheap and accessible platform
- Connected via a Wifi router
- No GPU



Nvidia Jetson TX2:

- High-end embedded platform
- Has a GPU

Moreover, we measured whole system power with a power analyzer

Software Overview

Dependencies:

- Ubuntu 16.04
- Keras 2.1
  - With Tensorflow backend for Raspberry Pis
  - With Tensorflow-GPU backend for TX2
- Apache Avro for procedure call and data serialization

Image Recognition Models:

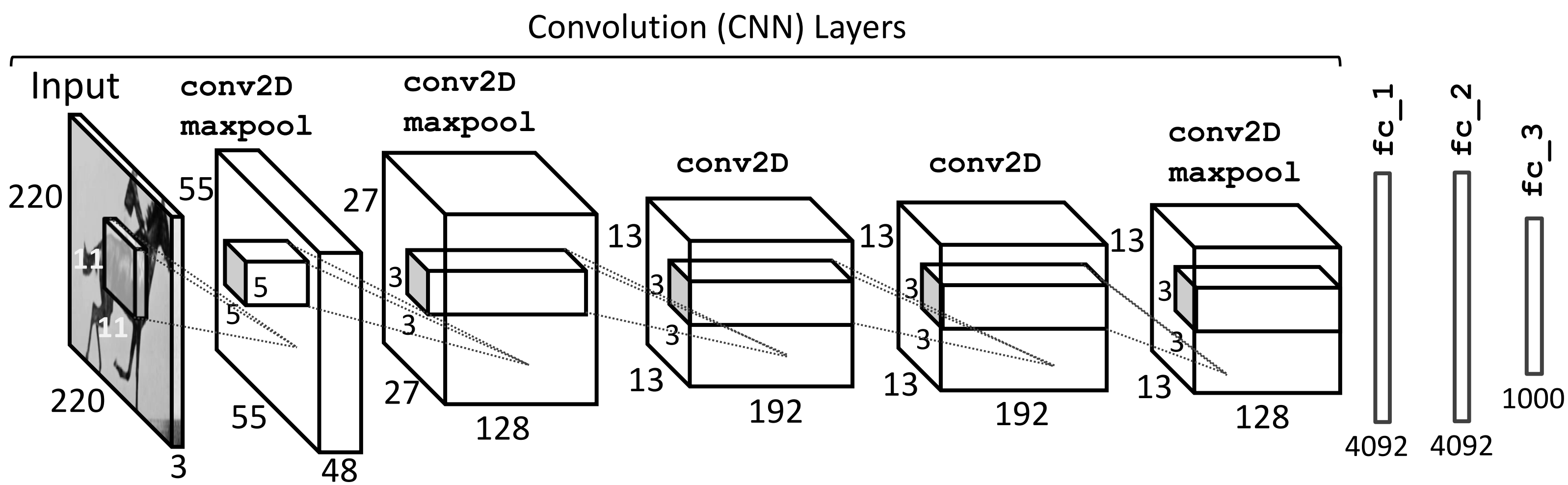
- AlexNet
- VGG16

AlexNet

Input Size: 220x220x3

Five convolution layers

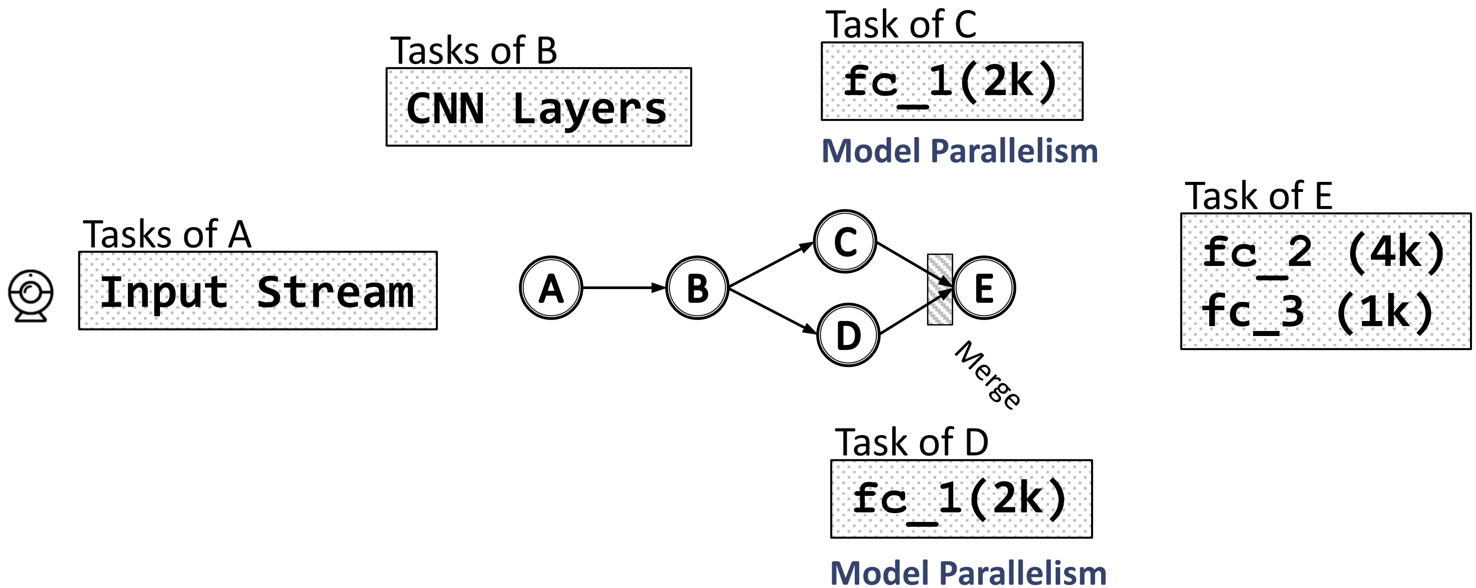
Three fully connected layers

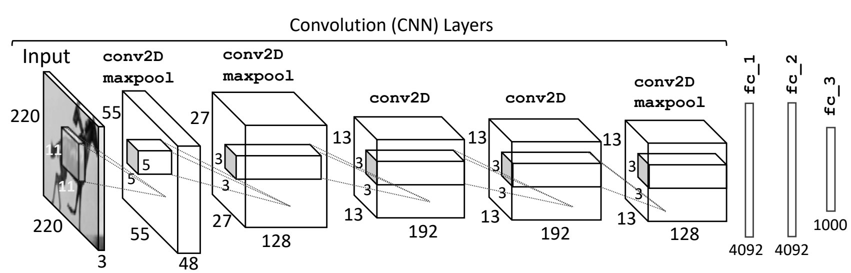


A. Krizhevsky et al., “Imagenet Classification With Deep Convolutional Neural Networks,” in NIPS 2012

AlexNet Distribution I

Five-device system:





AlexNet Distribution II

Six-device system:

Tasks of B & C

`CNN Layers`

Data Parallelism

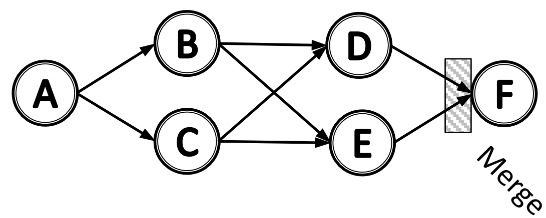
Task of D

`fc_1(2k)`

Model Parallelism

Tasks of A

`Input Stream`



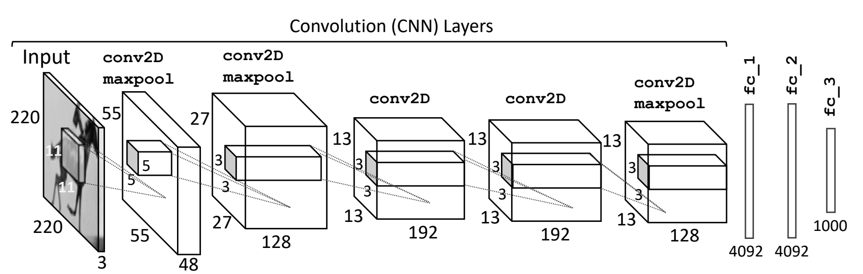
Task of F

`fc_2 (4k)`
`fc_3 (1k)`

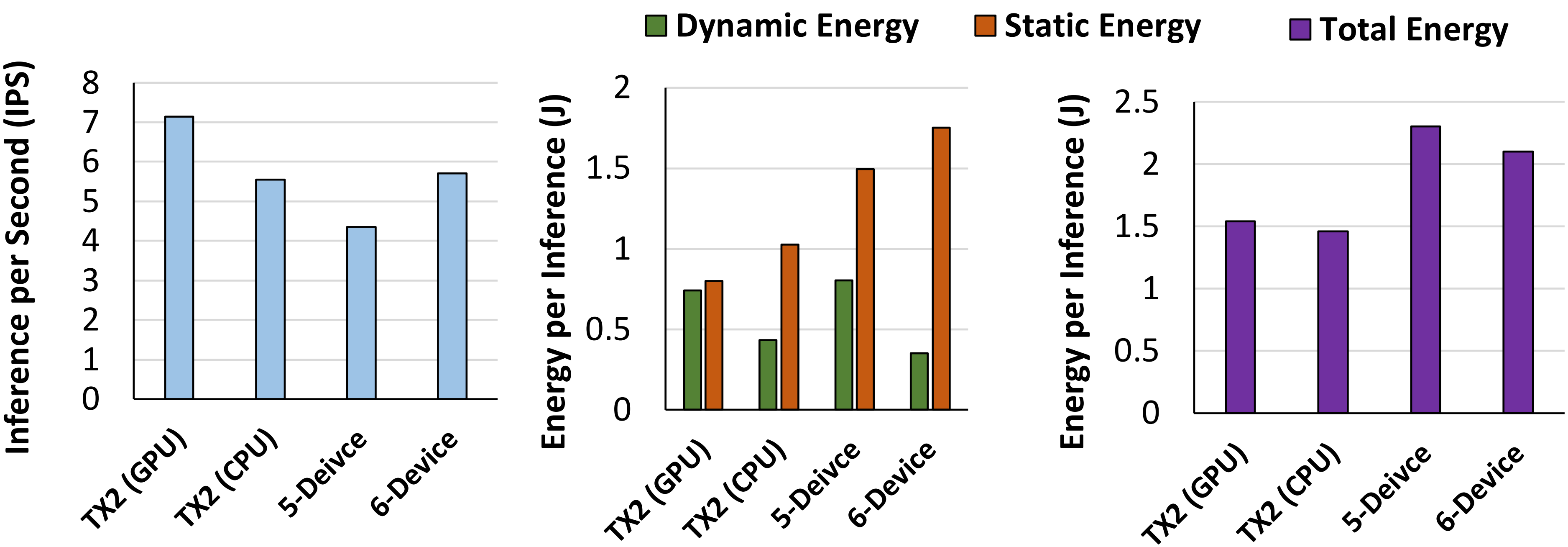
Task of E

`fc_1(2k)`

Model Parallelism



AlexNet Results



IPS | **Dynamic and Static Energy** | **Total Energy**

Comparable IPS with TX2 (-30%)

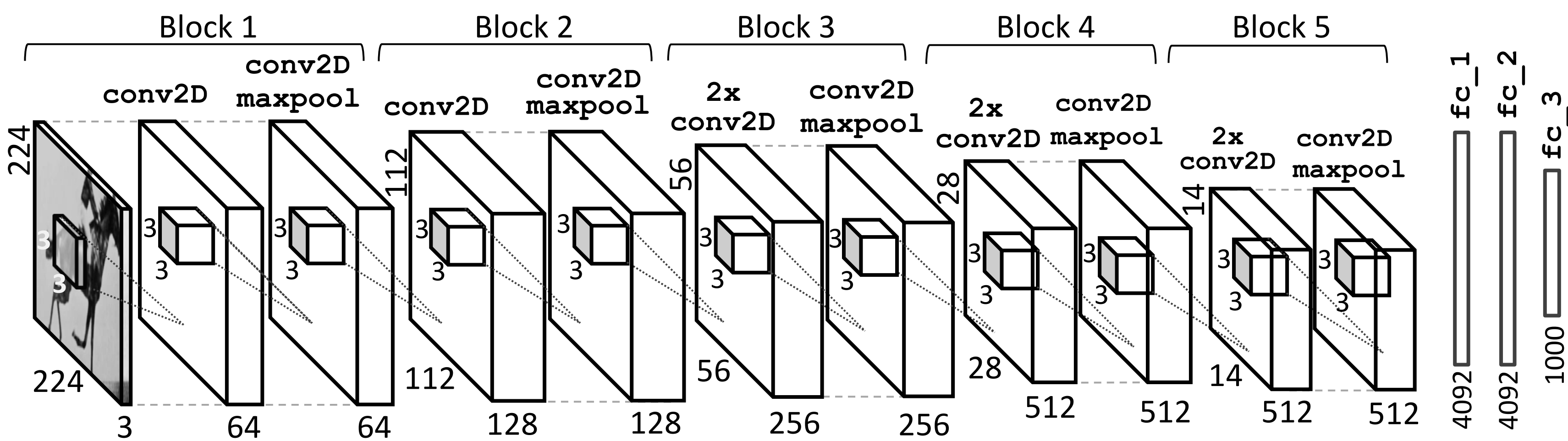
Lower dynamic energy consumption

VGG16

Input Size: 224x224x3

13 convolution layers

Three fully connected layers



K. Simonyan et al., "Very Deep Convolutional Networks for Large-Scale Image Recognition," in ICLR, 2015.

VGG16 Distribution I

Nine-device system:

Tasks of A

`Input Stream`

Tasks of C, D,& E

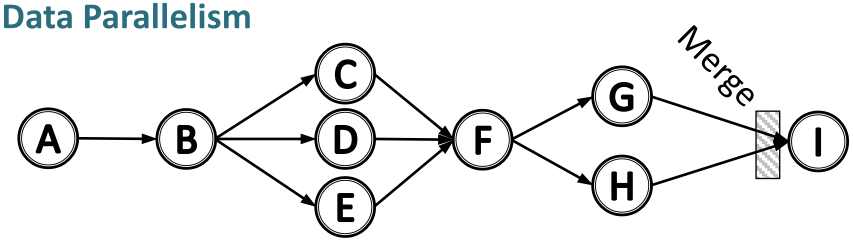
`Block 2,3,4`

Data Parallelism

Task of G

`fc_1(2k)`

Model Parallelism



Task of I

`fc_2 (4K)`
`fc_3 (1k)`

Tasks of B

`Block 1`

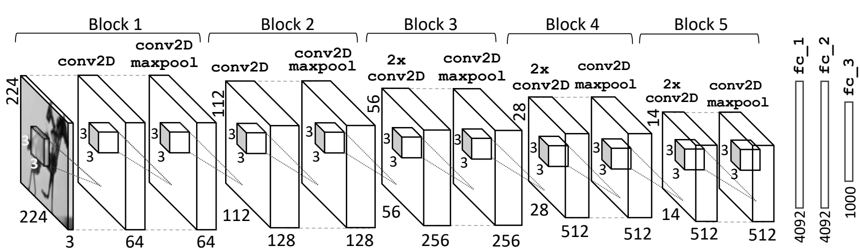
Tasks of F

`Block 5`

Task of H

`fc_1(2k)`

Model Parallelism



VGG16 Distribution II

11-device system:

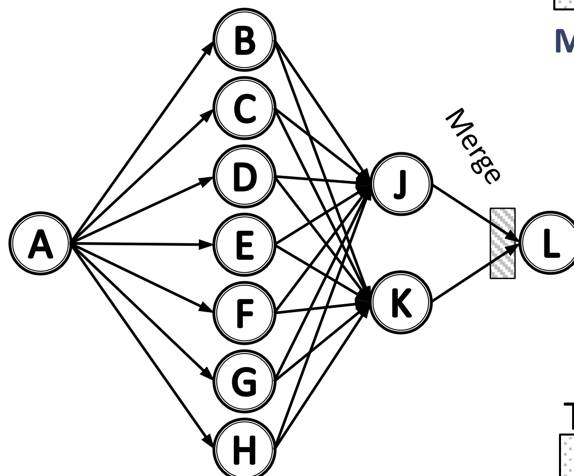
Tasks of A

`Input Stream`

Tasks of B, C, D, E
F, G,& H

```
Block 1,2,
      3,4,5
```

Data Parallelism



Task of J

`fc_1(2k)`

Model Parallelism

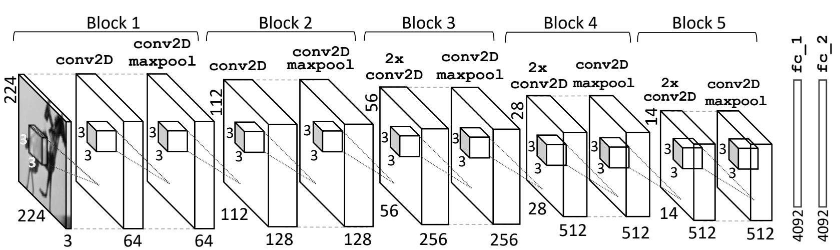
Task of L

```
fc_2 (4K)
fc_3 (1k)
```

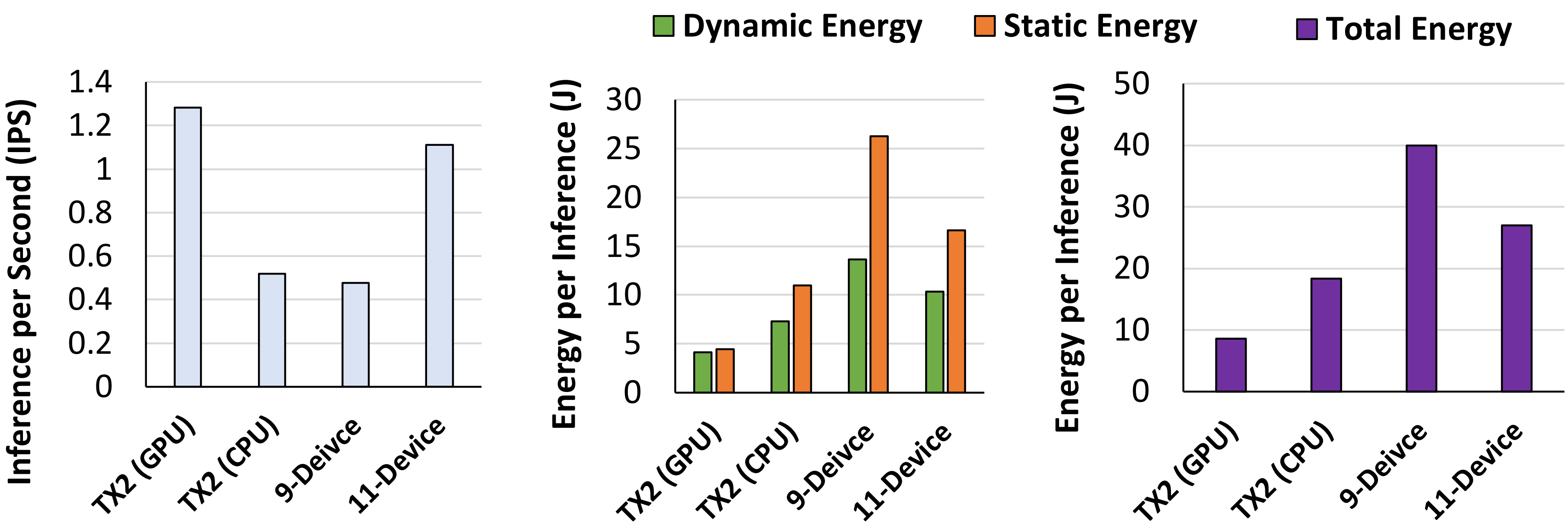
Task of K

`fc_1(2k)`

Model Parallelism



VGG16 Results



IPS **Dynamic and Static Energy** **Total Energy**

Comparable IPS with TX2 (-15%)

We achieve 2.3x speedup, by reassigning CNN blocks

Georgia Tech comparch

Conclusions

- We used a farm of Raspberry PIs for DNN processing
- We are able to process IoT data locally by distribution
- Our technique achieves acceptable real-time performance

Future Work:

- Study the robustness of such systems
- Apply our technique to more DNN models
- Implement our model on distributed robot systems