AI Robotics KR Network Acceleration study #2

Neural Network Inference on Mobile SoCs

Siqi Wang, et al.

Archive 2020

Presenter : Ji-ye Jeon (jyeah05@gmail.com)

2020-04-28

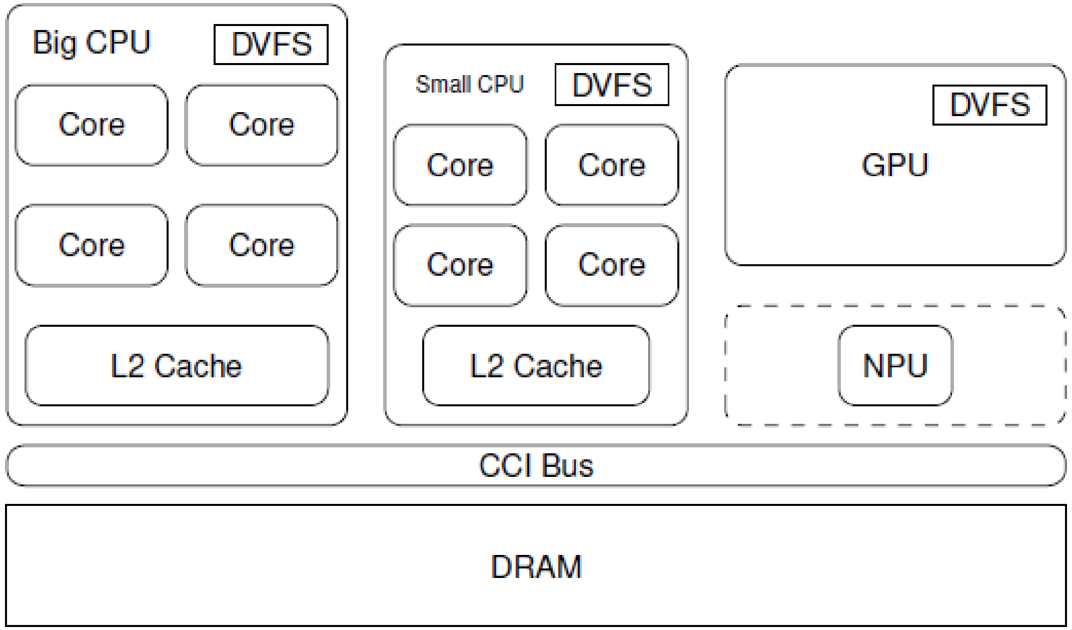
Index

- Introduction
- Experimental Setup
- Experiments and Results
    - Comparing Each components of SoCs
    - Roofline Analysis
    - How to Improve?
- Conclusion and Insights
- Discussion

Introduction

- Heterogenous Multi-processor SoCs for ML inference
  - CPU cluster (ARM big.Little)
  - GPUs
  - NPUs

- *However, "The majority of mobile inference run on CPUs[1]"*
  - Programmability
  - Little performance gap between CPUs and GPUs

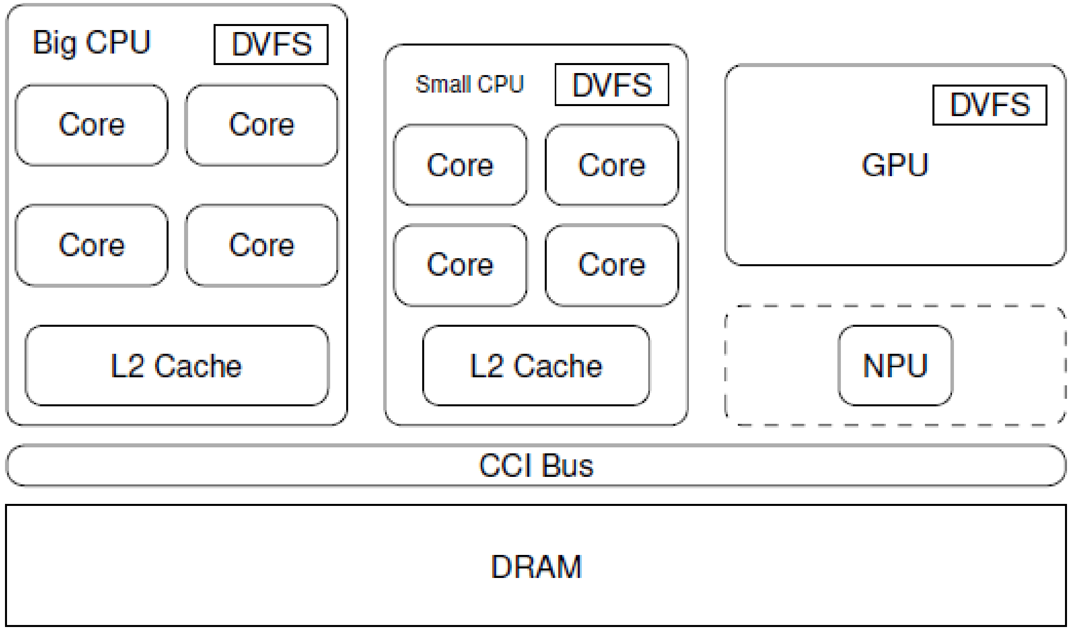


[1] *"Machine Learning at FaceBook: Understanding Inference at the Edge",* FaceBook Inc, HPCA, 2019.

Introduction

1) Conducting quantitative experiments on each components of Mobile SoCs
   - Characterize inference capability of each components

2) Roofline Model Analysis
   - Find clues for improvements
   - Comparing effect of quantization etc…



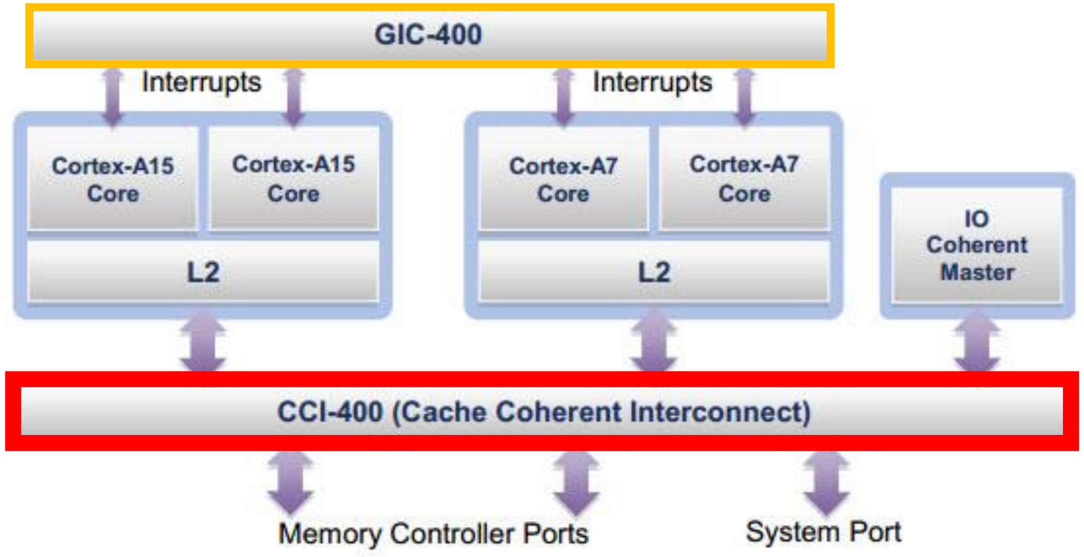
[1] *"Machine Learning at FaceBook: Understanding Inference at the Edge",* FaceBook Inc, HPCA, 2019.

Index

- Introduction
- Experimental Setup
- Experiments and Results
  - Comparing Each components of SoCs
  - Roofline Analysis
  - How to Improve?
- Conclusion and Insights
- Discussion

Background –ARM big.LITTLE Architecture

- Heterogeneous Architecture
  - big : High performance, High Power consumption
  - LITTLE : Low performance, Low Power consumption
  - DVFS (Dynamic Voltage and Frequency Switching)
  - CCI/GIC
    - GIC: Generic Interrupt Controller
    - CCI : Cache Coherence Interconnect



Experimental Setup – Exynos 5422 vs. Kirin 970

- Mid-end and High-end AP

	Exynos 5422	Kirin 970	
Release date	2014	2017	
area	28nm	10nm	
CPU-big	Cortex-A15(In-order) 2GHz	Cortex-A73 (OoO) 2.36GHz	1.18x
CPU-LITTLE	Cortex-A7(In-order) 1.4GHz	Cortex-A53(In-order) 1.8GHz	1.29x
GPU	Mali T628 MP6 57.6GFLOPs	Mali G72 MP12 244.8GFLOPs	
NPU	X	Cambricon-1A	HiAi DDK API

Index

- Introduction
- Experimental Setup
- Experiments and Results
  - Comparing Each components of SoCs
  - Roofline Analysis
  - How to Improve?
- Conclusion and Insights
- Discussion

Individual Heterogeneous Components

- Throughput

  ✓ Exynos vs. Kirin
  - Big (4.4x) , LITTLE (2.6x), GPU (4.2x)

  ✓ Big vs. LITTLE
  - 4x ~ 2.5x

  ✓ NPU vs. GPU
  - NPU is only 1.6x better than G72

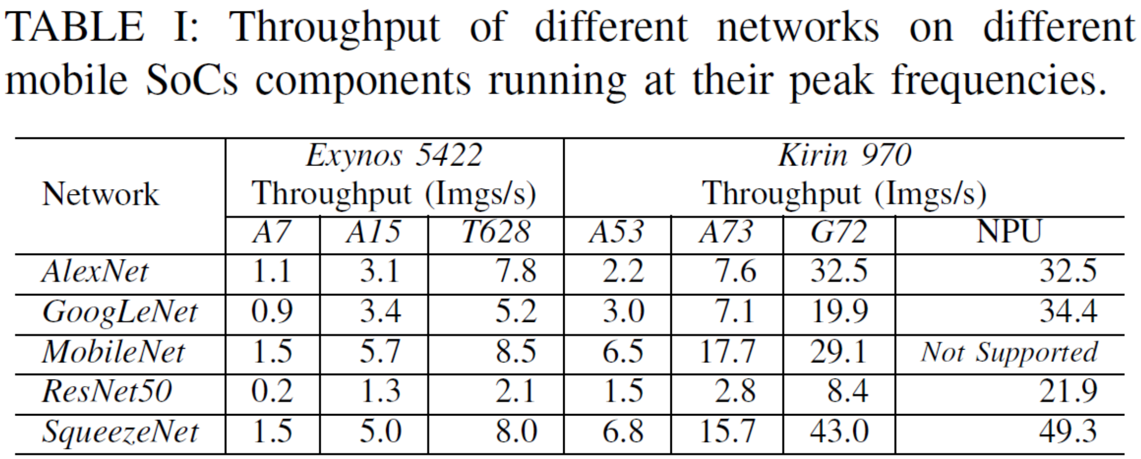


TABLE I: Throughput of different networks on different mobile SoCs components running at their peak frequencies.

Network	*Exynos 5422* Throughput (Imgs/s)			*Kirin 970* Throughput (Imgs/s)			
	A7	*A15*	*T628*	*A53*	*A73*	*G72*	NPU
AlexNet	1.1	3.1	7.8	2.2	7.6	32.5	32.5
GoogLeNet	0.9	3.4	5.2	3.0	7.1	19.9	34.4
MobileNet	1.5	5.7	8.5	6.5	17.7	29.1	*Not Supported*
ResNet50	0.2	1.3	2.1	1.5	2.8	8.4	21.9
SqueezeNet	1.5	5.0	8.0	6.8	15.7	43.0	49.3

Individual Heterogeneous Components

- Energy efficiency

✓ Comparing each component
  - NPU > GPU > LITTLE > big

✓ Comparing each platform
  - Kirin > Exynos
  - Except LITTLE cluster
  - A53 : larger TLB, complex branch prediction

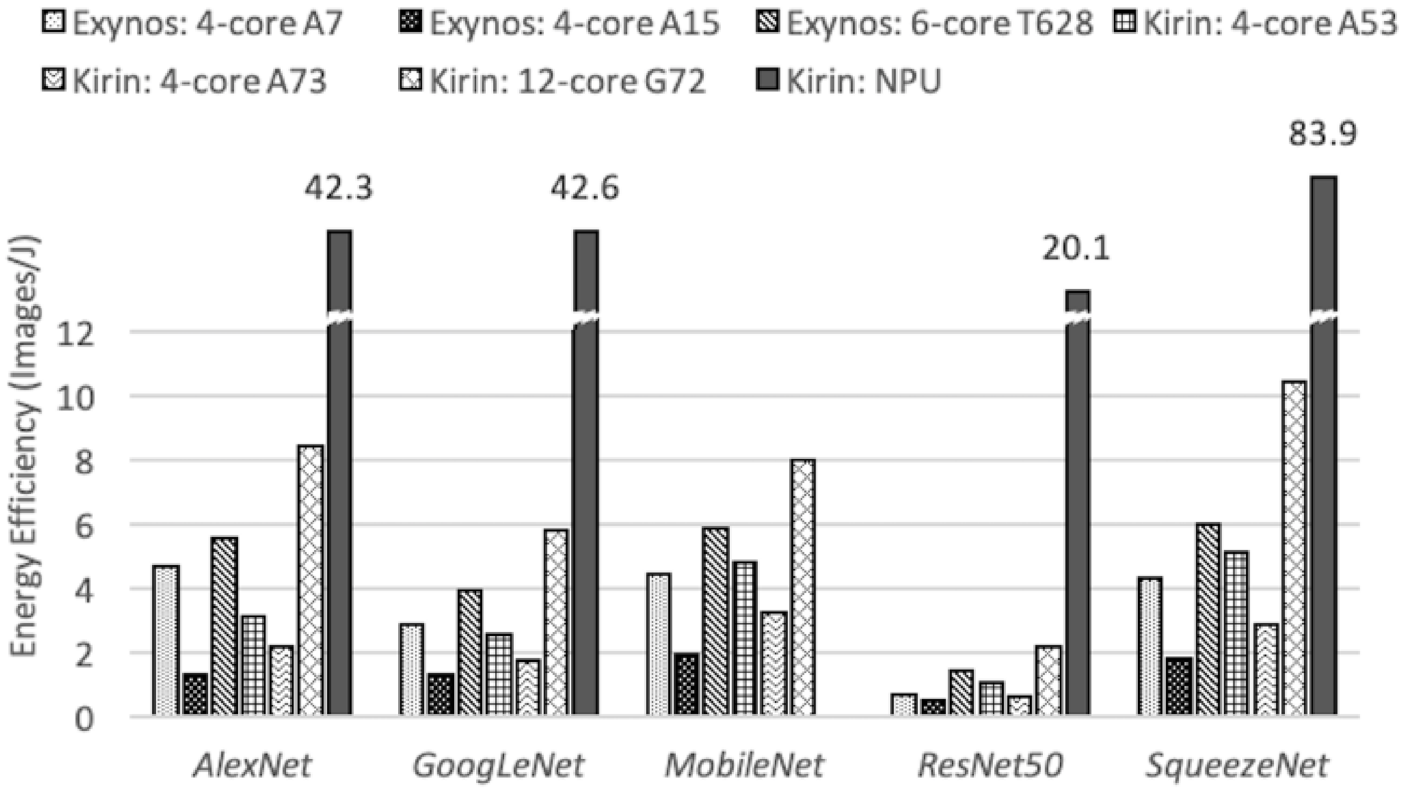


Fig. 2: Energy efficiency of different components while running at their peak frequencies.

Individual Heterogeneous Components

- Technology scaling vs. Architectural Innovation?

	Exynos 5422	**Kirin 970**	
Release date	2014	2017	
area	28nm	10nm	
CPU-big	Cortex-A15(In-order) 2GHz	Cortex-A73 (OoO) 2.36GHz	1.18x
CPU-LITTLE	Cortex-A7(in-order) 1.4GHz	Cortex-A53(in-order) 1.8GHz	1.29x

✓ *Impact of uArch Innovation !*

- 4.4x ~ 2.5x throughput improvements
- Larger cache, branch predictor, cache prefetecher etc..

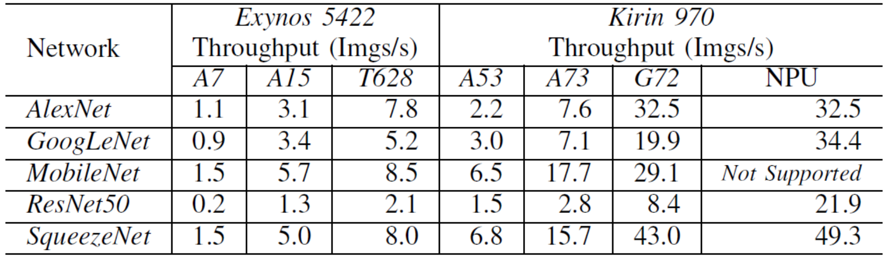
✓ Meanwhile, power consumption increased in A53

- A53 has 2x power consumption compared to A7
- Increased area

Individual Heterogeneous Components

- Insight
  - ✓ GPUs can be better option than NPUs
    - Generality, easy optimization
    - Has comparable performance(1.6x) and satisfactory Energy Efficiency
  - ✓ CPUs, are still critical for inferencing
    - Especially in Low-end mobile SoCs
  - ✓ Running individual component alone is not enough
    - Co-execution is neeeded

TABLE I: Throughput of different networks on different mobile SoCs components running at their peak frequencies.



Network	*Exynos 5422* Throughput (Imgs/s)			*Kirin 970* Throughput (Imgs/s)			
	A7	*A15*	*T628*	*A53*	*A73*	*G72*	NPU
AlexNet	1.1	3.1	7.8	2.2	7.6	32.5	32.5
GoogLeNet	0.9	3.4	5.2	3.0	7.1	19.9	34.4
MobileNet	1.5	5.7	8.5	6.5	17.7	29.1	*Not Supported*
ResNet50	0.2	1.3	2.1	1.5	2.8	8.4	21.9
SqueezeNet	1.5	5.0	8.0	6.8	15.7	43.0	49.3

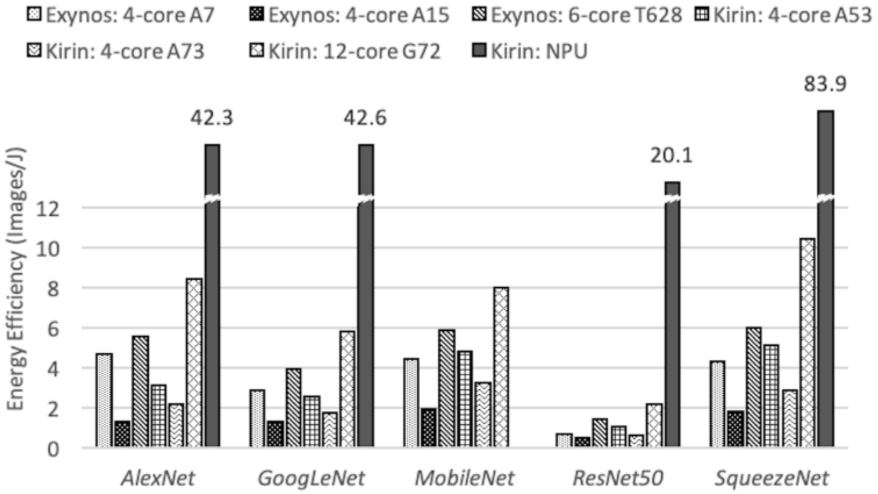


Fig. 2: Energy efficiency of different components while running at their peak frequencies.

Index

- Introduction
- Experimental Setup
- Experiments and Results
    - Comparing Each components of SoCs
    - Roofline Analysis
    - How to Improve?
- Conclusion and Insights
- Discussion

Individual Heterogeneous Components

- Building Roofline Model for Exynos 5422
  - Exynos 5422 specifications
    - 3.44 GB/s (A15), 0.49 GB/s (A7), 6.15 GB/s (T628)

  - Theoretical $OI_t = GOPS/Mem\_Access$
    - Calculated by analyzing the code

  - Empirical $OI_e = GOPS/DRAM\_Access$
    - Aware of cache
    - Calculated by using actual DRAM access

Individual Heterogeneous Components

- Across Different Component

✓ A7 (LITTLE) and T628 (GPU)

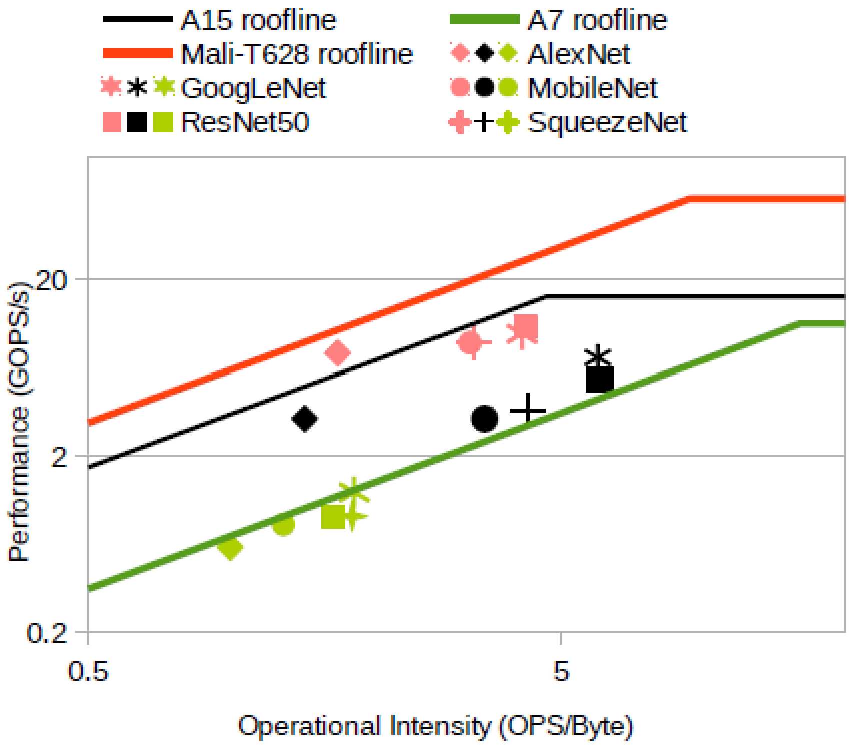
- Severe memory bottleneck

✓ A15 (big)

- Larger cache size (L2: 2MB)
- Performance falls in both region
- Compute-bound : ResNet50, GoogLeNet

✓ Network Characteristics

- AlexNet : huge parameter
- Small filter size leads sub-optimal parallelization

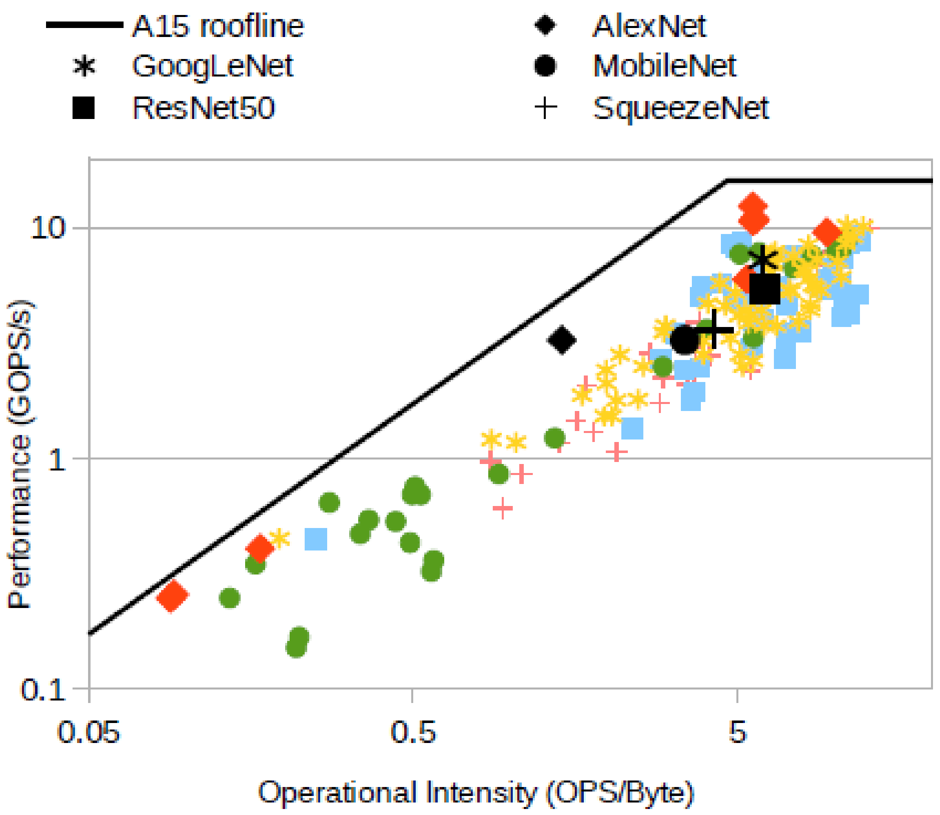


Individual Heterogeneous Components

- Major layer analysis on A15 (big)

✓ AlexNet (◆)
- Conv : compute-bound
- FC : memory-bound

✓ Layer-level optimization could be key of improvement
- Per-layer DVFS
- Fine-grain layer level co-execution



Index

- Introduction
- Experimental Setup
- Experiments and Results
  - Comparing Each components of SoCs
  - Roofline Analysis
  - How to Improve?
- Conclusion and Insights
- Discussion

Improving Performance

✓ Quantization was not enough
- Quantized MobileNet in ARM-CL to 8bit-weight
- Reduced memory access
- De-quantization / re-quantization overhead

✓ NPU is memory-bound

✓ Co-execution with multiple components
- 50% throughput improvement over GPU
- Has better energy efficiency than big cluster

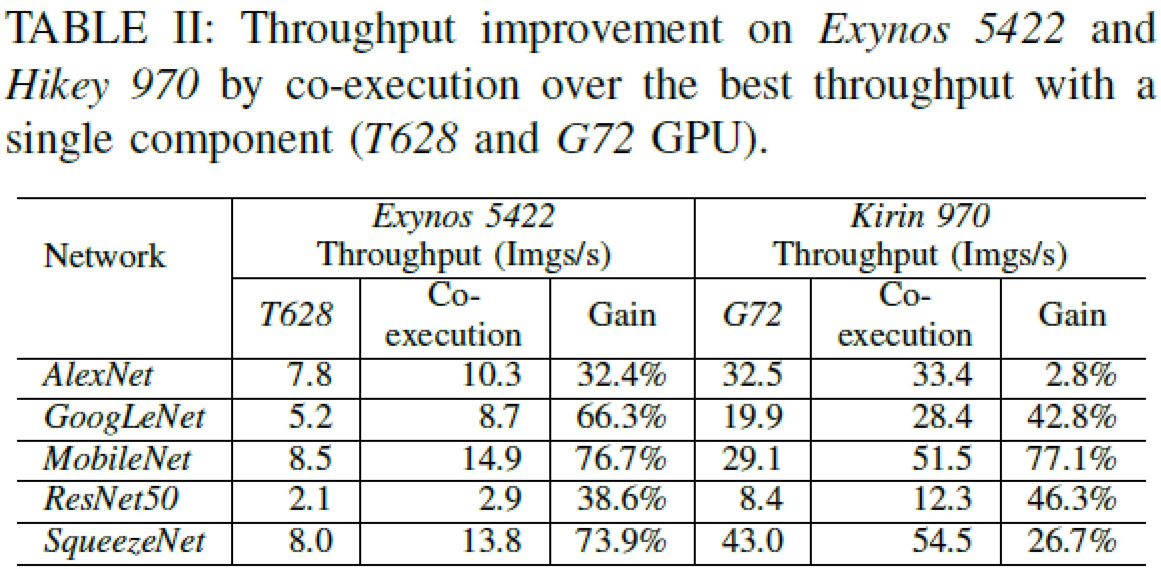


TABLE II: Throughput improvement on *Exynos 5422* and *Hikey 970* by co-execution over the best throughput with a single component (*T628* and *G72* GPU).

Network	*Exynos 5422* Throughput (Imgs/s)			*Kirin 970* Throughput (Imgs/s)		
	T628	Co-execution	Gain	*G72*	Co-execution	Gain
AlexNet	7.8	10.3	32.4%	32.5	33.4	2.8%
GoogLeNet	5.2	8.7	66.3%	19.9	28.4	42.8%
MobileNet	8.5	14.9	76.7%	29.1	51.5	77.1%
ResNet50	2.1	2.9	38.6%	8.4	12.3	46.3%
SqueezeNet	8.0	13.8	73.9%	43.0	54.5	26.7%

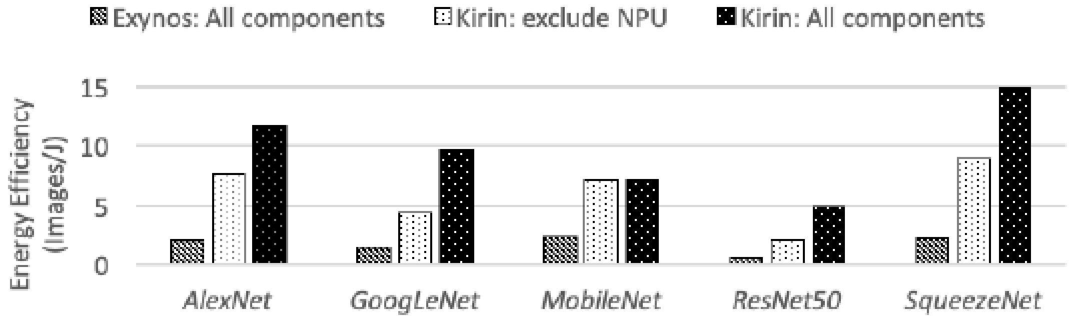


Fig. 4: Energy efficiency of co-execution on *Exynos 5422* with all components, on *Kirin 970* with CPU and GPU (excluding NPU) and all components (including NPU).

Index

- Introduction
- Experimental Setup
- Experiments and Results
  - Comparing Each components of SoCs
  - Roofline Analysis
  - How to Improve?
- Conclusion and Insights
- Discussion

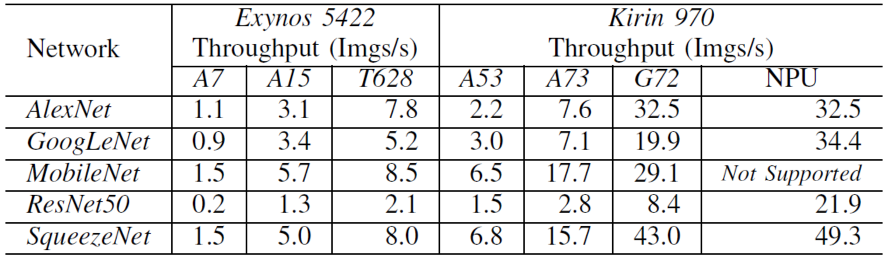
Conclusion and Insights

✓ Anyway, CPU clusters will take major part in mobile inference
  - Optimization / Flexibility
  - Performance gap between CPUs and GPUs was "actually" under 3x

✓ Rooms for improvements
  - LITTLE CPU / GPU / NPU : memory bounded
  - big CPU : memoty bounded / computation bounded
    - Especially for Conv

✓ Throughput increase by 2x using co-execution

TABLE I: Throughput of different networks on different mobile SoCs components running at their peak frequencies.



Network	*Exynos 5422* Throughput (Imgs/s)			*Kirin 970* Throughput (Imgs/s)			
	A7	*A15*	*T628*	*A53*	*A73*	*G72*	NPU
AlexNet	1.1	3.1	7.8	2.2	7.6	32.5	32.5
GoogLeNet	0.9	3.4	5.2	3.0	7.1	19.9	34.4
MobileNet	1.5	5.7	8.5	6.5	17.7	29.1	*Not Supported*
ResNet50	0.2	1.3	2.1	1.5	2.8	8.4	21.9
SqueezeNet	1.5	5.0	8.0	6.8	15.7	43.0	49.3

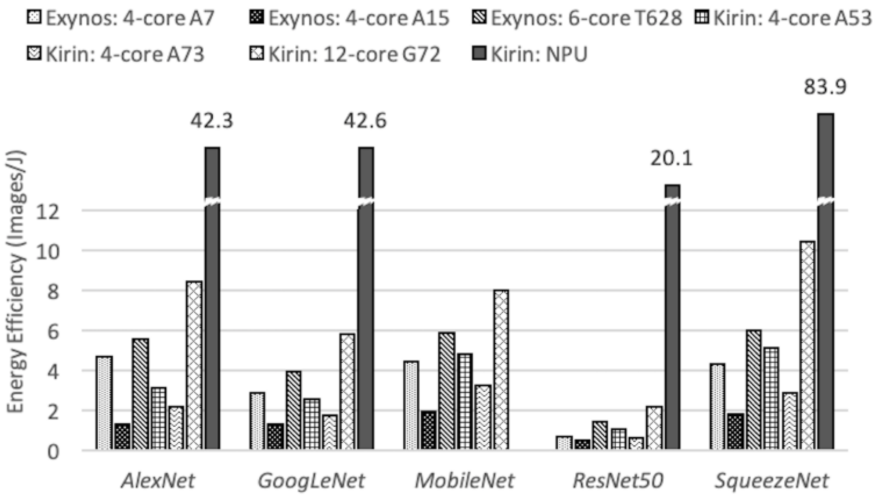


Fig. 2: Energy efficiency of different components while running at their peak frequencies.

Index

- Introduction

- Experimental Setup
  - Exynos & Kirin

- Experiments and Results
  - Comparing Each components of SoCs
  - Roofline Analysis
  - How to Improve?

- Conclusion and Insights

- Discussion

Question?