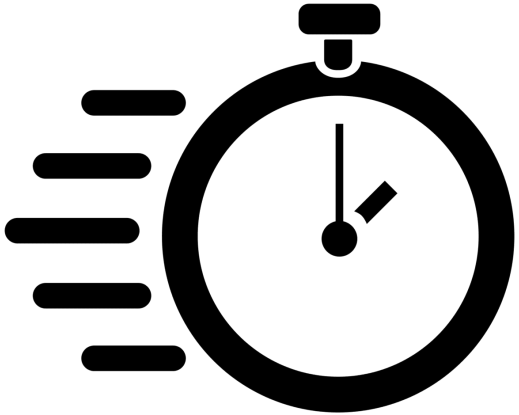
Hardware-Aware Deep Neural Architecture Search

Peizhao Zhang
Mobile Vision, Facebook

Why efficient ConvNets?

Goal: Accuracy and Efficiency

- Mobile and embedded computer vision applications require accurate and efficient ConvNets



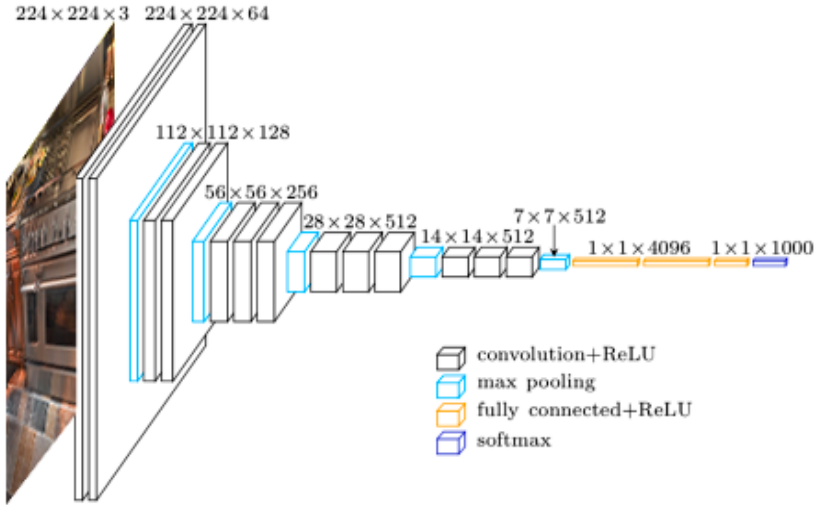
Accuracy: Essential for many applications such as security cameras and autonomous driving

Efficiency: Real-time inference on embedded processors with limited compute & power budgets

Designing accurate and efficient ConvNets is challenging.

Challenge #1: Intractable design space

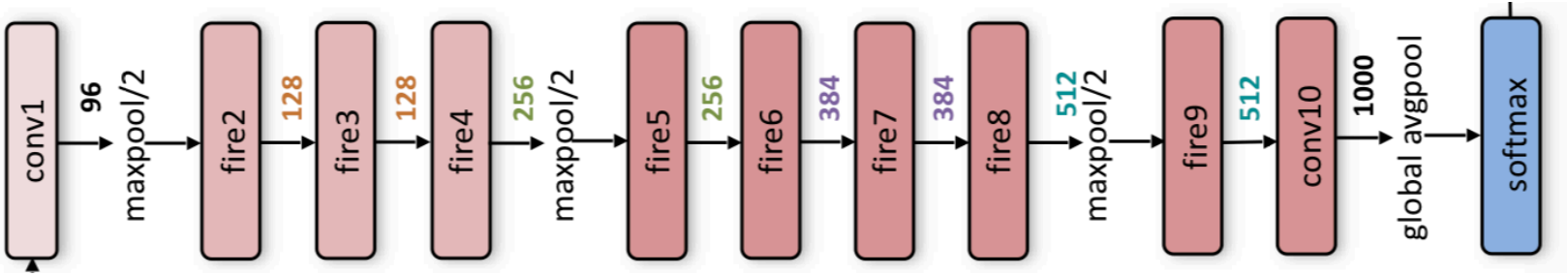
- Design space of Deep Neural Nets is huge!
  - VGG16[1] has 13 conv layers
  - Design choices for each layer:
    - kernel size = {1, 3, 5}
    - channel size = {32, 64, 128, 256, 512}
  - Search space = (3x5)^13 = 2e15



[1] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Challenge #2: Conditional optimality

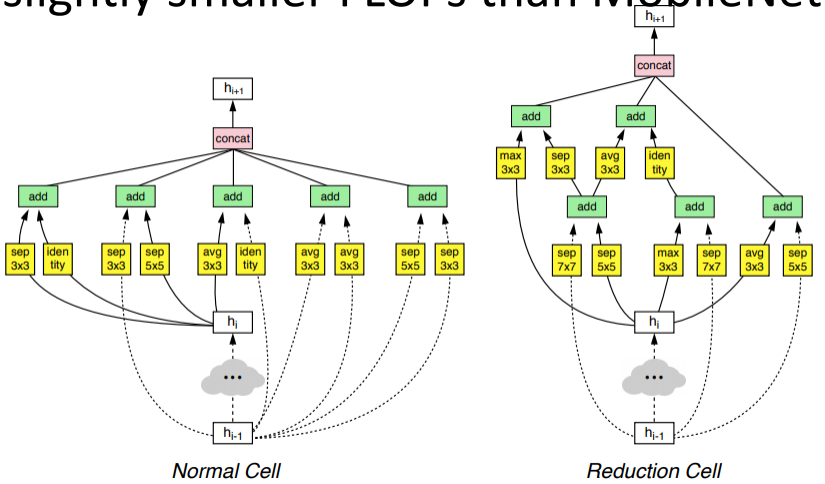
- Ideally, we should design different ConvNets for different devices
- In reality, due to the cost of design & training ConvNets, we can only afford to **design one** and **deploy to all** conditions



GPUs iPhones Android Low end phones Wearable IoT

Challenge #3: Inaccurate metrics

- Previous works focus on efficiency proxies: parameter size or FLOPs
- Proxies do not always reflect actual efficiency
  - NASNet-A[1] has slightly smaller FLOPs than MobileNetV1[2], but the latency is 1.6x slower



NASNet [2]

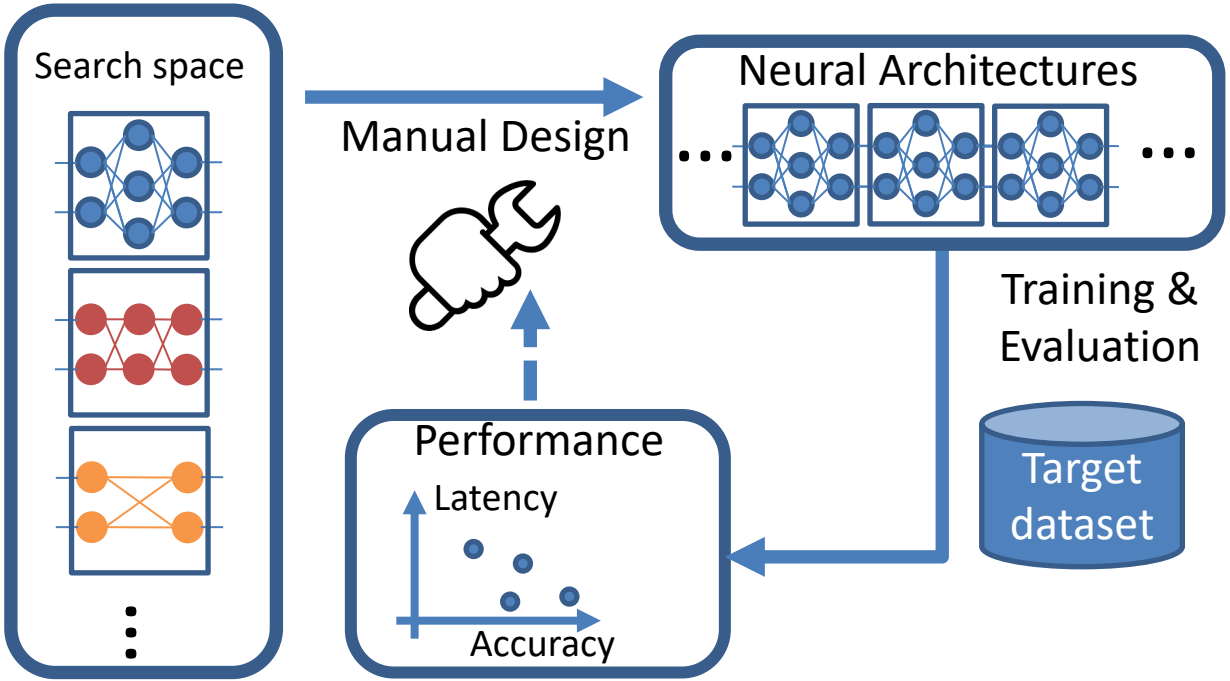
[1] Zoph, Barret, et al. "Learning transferable architectures for scalable image recognition." *CVPR18*

[2] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv:1704.04861

Rethinking the flow of ConvNet design

Previous: Manual design

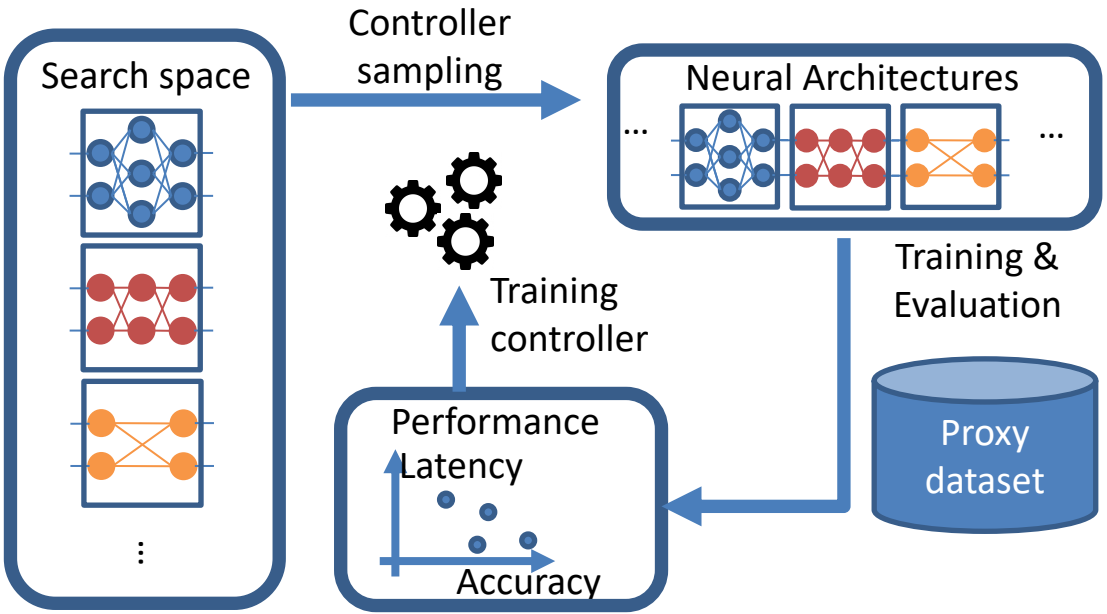
- Manual design:
  - Can only afford a few iterations -> suboptimal design



Previous: neural architecture search

Search based neural architecture search

- Discovered models surpassed manual design [1, 2]
- Computationally expensive: Need to enumerate and train thousands of NNs



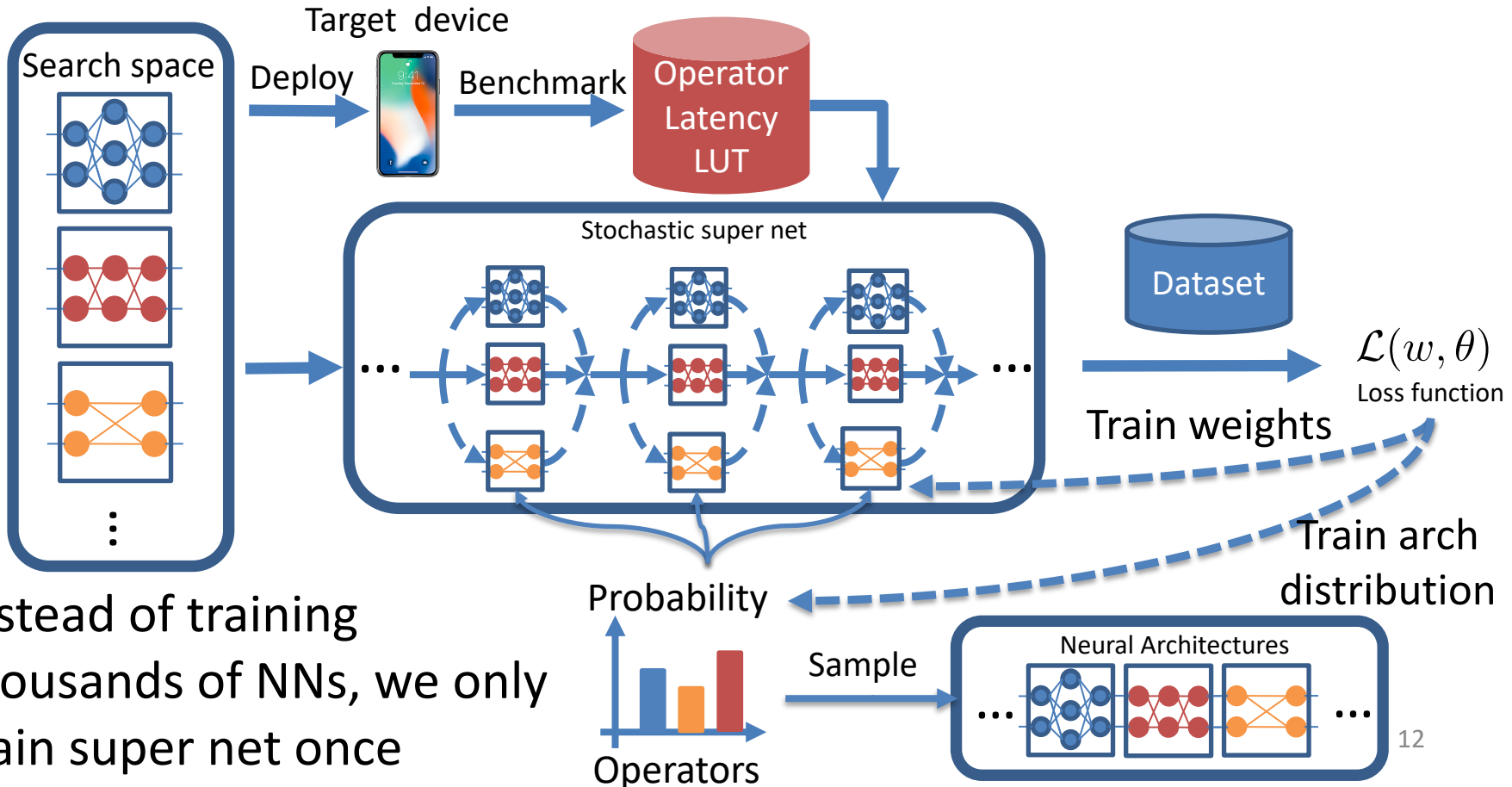
[1] Zoph, Barret, et al. "Learning transferable architectures for scalable image recognition." *CVPR18*
[2] Tan, Mingxing, et al. "Mnasnet: Platform-aware neural architecture search for mobile." CVPR19

Our approach:

1) **FBNet: Differentiable neural architecture search**
2) ChamNet: Hardware aware model adaptation

[1] Wu et al., FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search, CVPR 2019
[2] Dai et al., ChamNet: Towards Efficient Network Design through Platform-Aware Model Adaptation, CVPR 2019

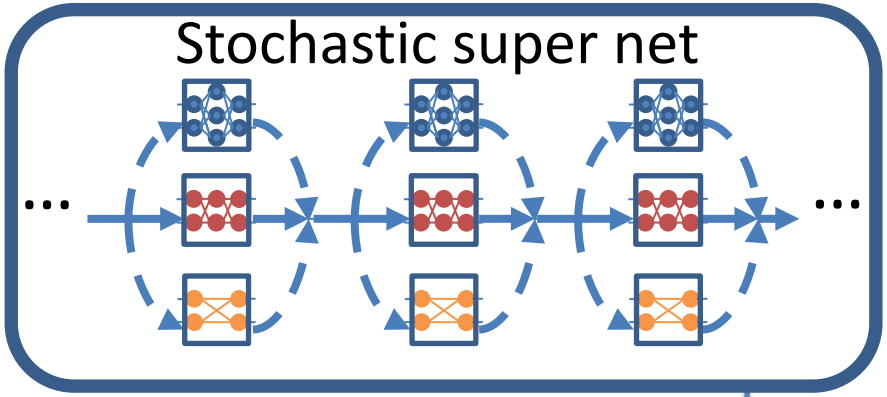
Differentiable Neural Architecture Search

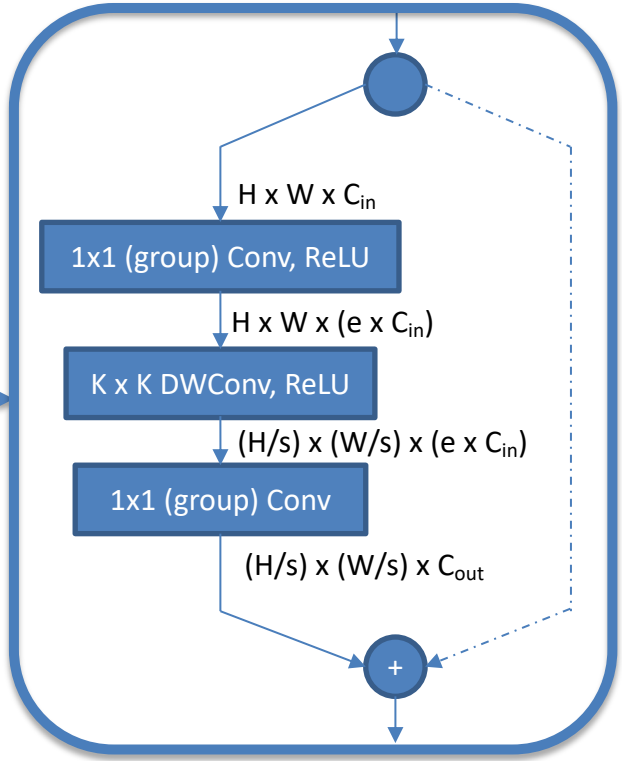


- Instead of training thousands of NNs, we only train super net once

FBNet search space

- Each “layer” of a network can choose a different module





Candidate modules with different hyper-parameters

- Kernel size: 3, 5
- Expansion rate: 1, 3, 6
- Skip: no-operation

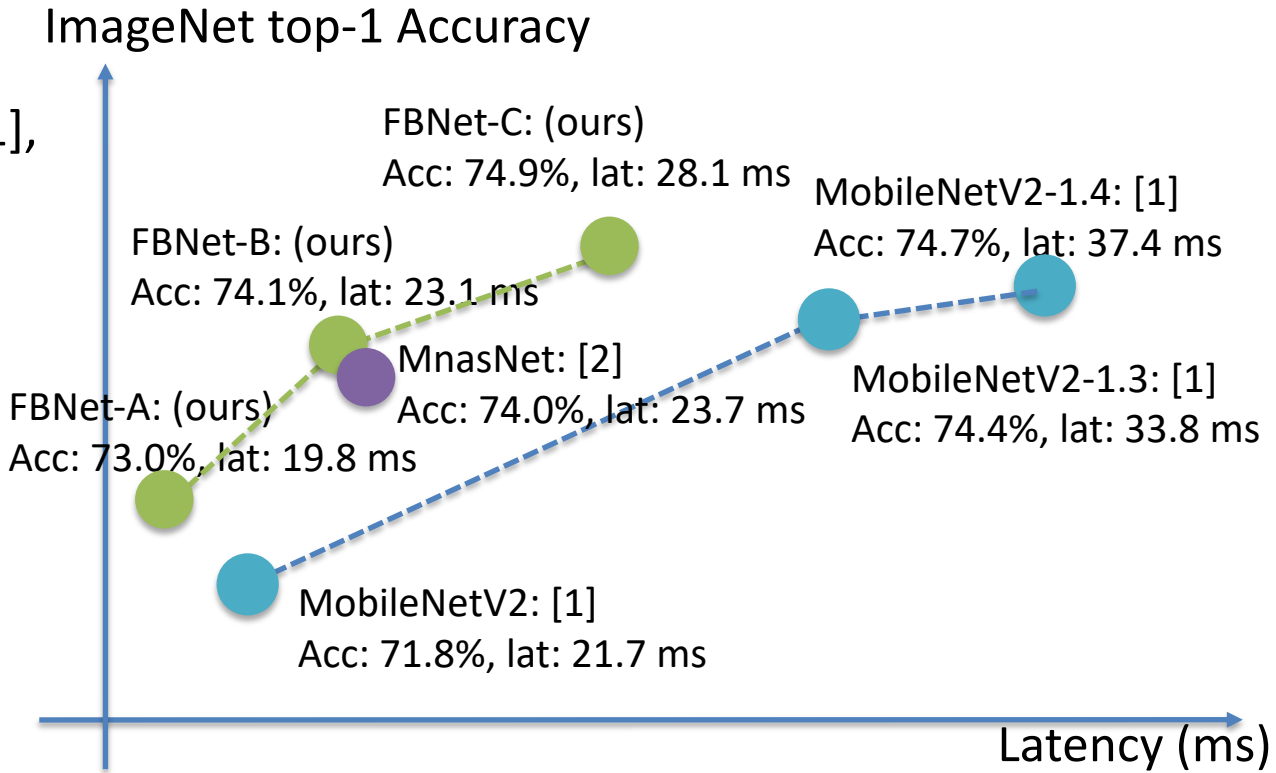
FBNets vs. previous state-of-the-art

FBNet-B vs MobileNetV2[1],

- Same accuracy,
- **1.5x faster,**
- **2.4x smaller FLOPs**

FBNet vs. MnasNet[2]

- 8 GPUs x 24 hours search cost
- **421x** lower



[1] Sandler, Mark, et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." *CVPR18*

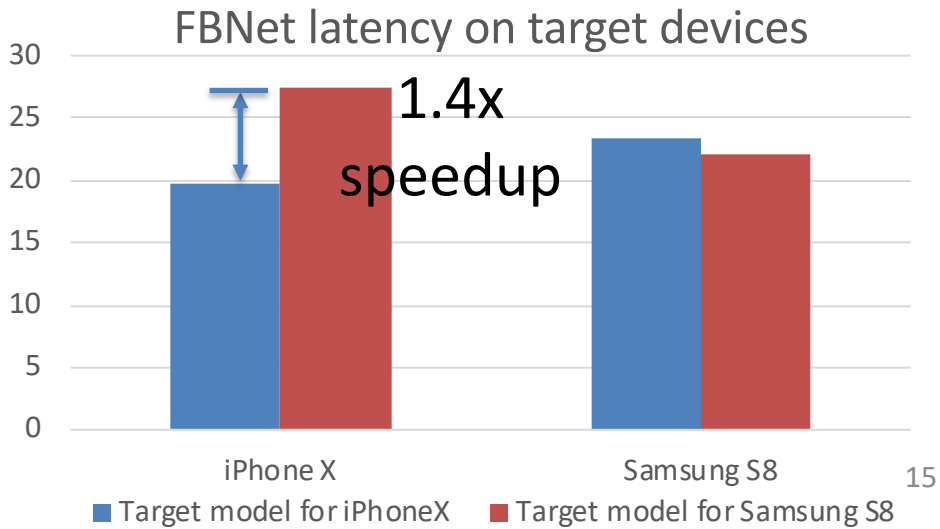
[2] Tan, Mingxing, et al. "Mnasnet: Platform-aware neural architecture search for mobile." *CVPR19*

FBNets for different target devices

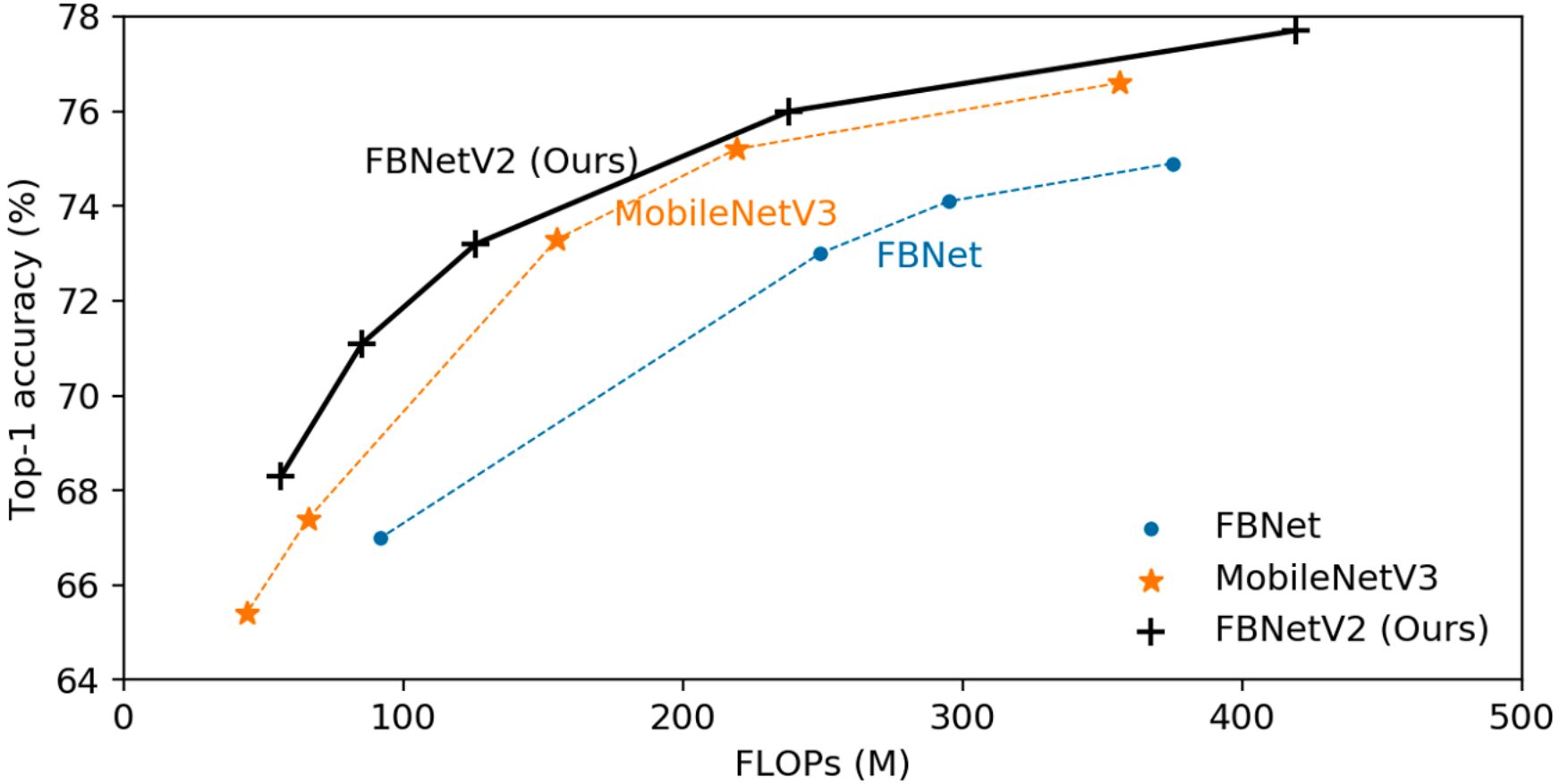
- Apple A11
- Big: 2 ARMv8 @ 2.5 GHz
- Little: 4 ARMv8 @ 1.4 GHz
- Vectorization: 4-wide 32-bit MAC
- LPDDR4x memory (30 GB/s)
- GPU + Neural Processing Engine

- Snapdragon 835
- Big: 4 ARMv8 @ 2.4 GHz
- Little: 4 ARMv8 @ 1.9 GHz
- Vectorization: 4-wide 32-bit MAC
- LPDDR4x memory (30 GB/s)
- Adreno 540 GPU

- Under similar accuracy constraint (73.27% vs 73.20%), FBNet optimized for iPhone-X achieves 1.4x speedup over the Samsung optimized model



FBNetV2: Differentiable Neural Architecture Search for Spatial and Channel Dimensions



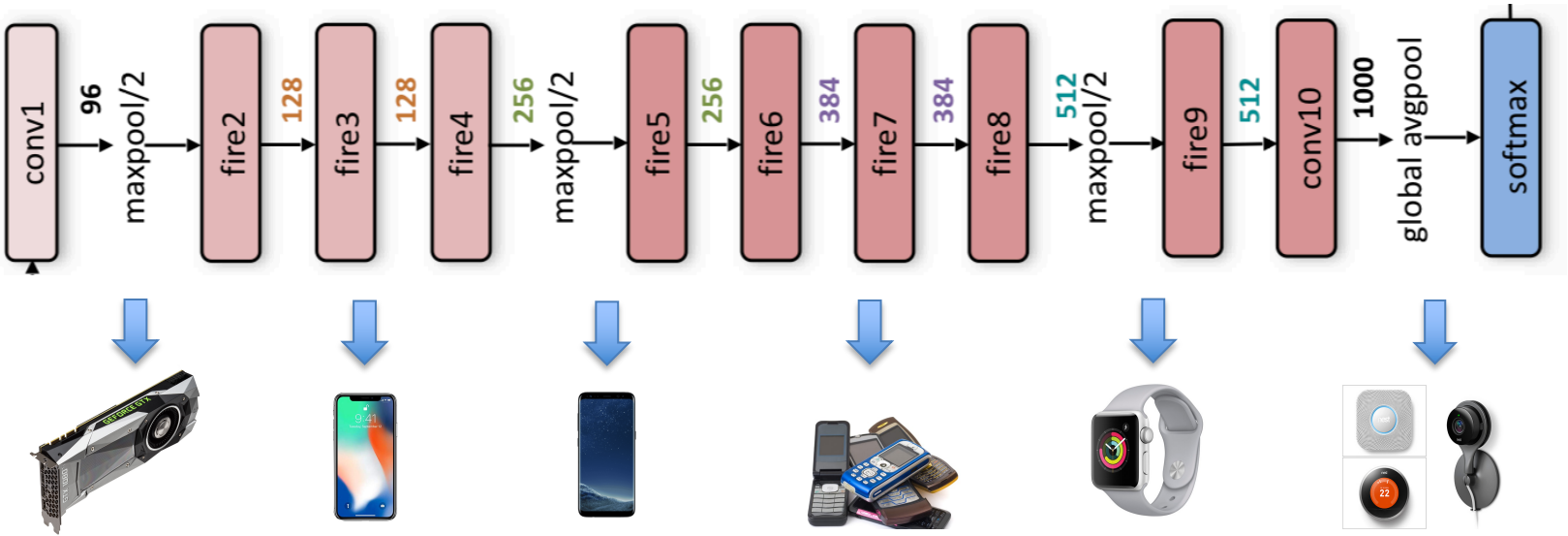
Our approach:

1. FBNet: Differentiable neural architecture search
2. **ChamNet: Hardware aware model adaptation**

[1] Wu et al., FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search, CVPR 2019
[2] Dai et al., ChamNet: Towards Efficient Network Design through Platform-Aware Model Adaptation, CVPR 2019

Challenge #2: Conditional optimality

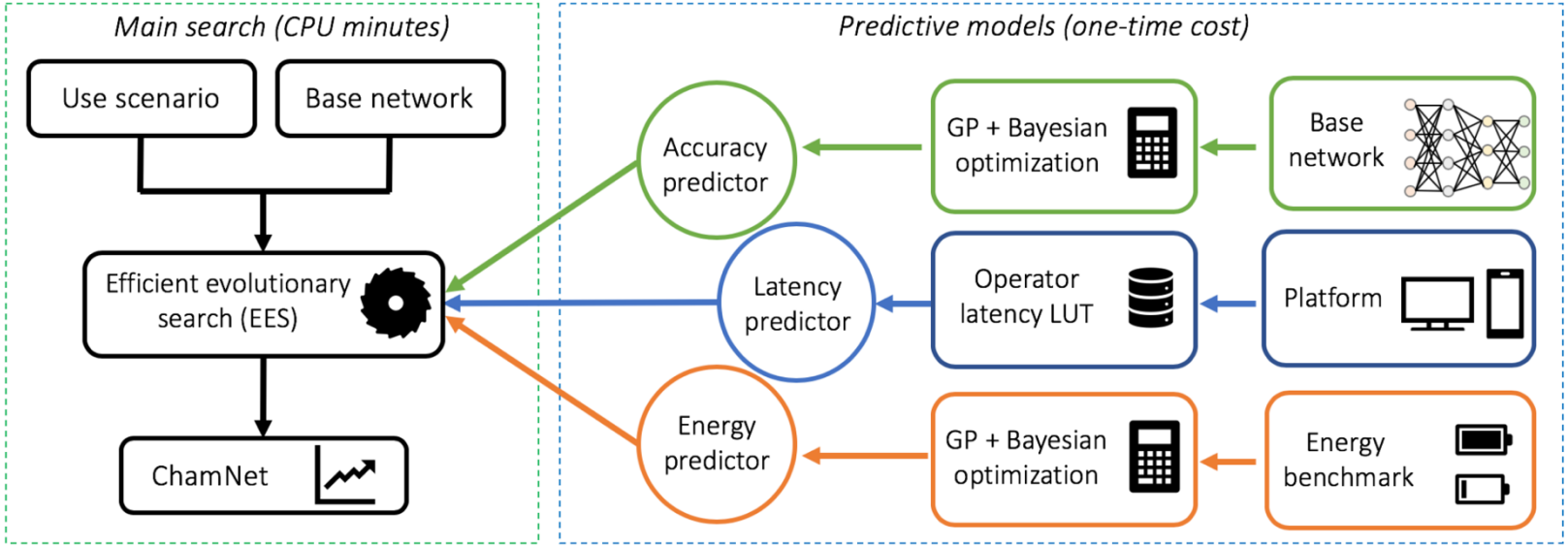
- **One model cannot fit all**
- Performing a NAS per device per task per use case: Too expensive



The Chameleon framework

Chameleon framework

- Predictive model based
- Search in CPU minutes with adaptive genetic algorithm



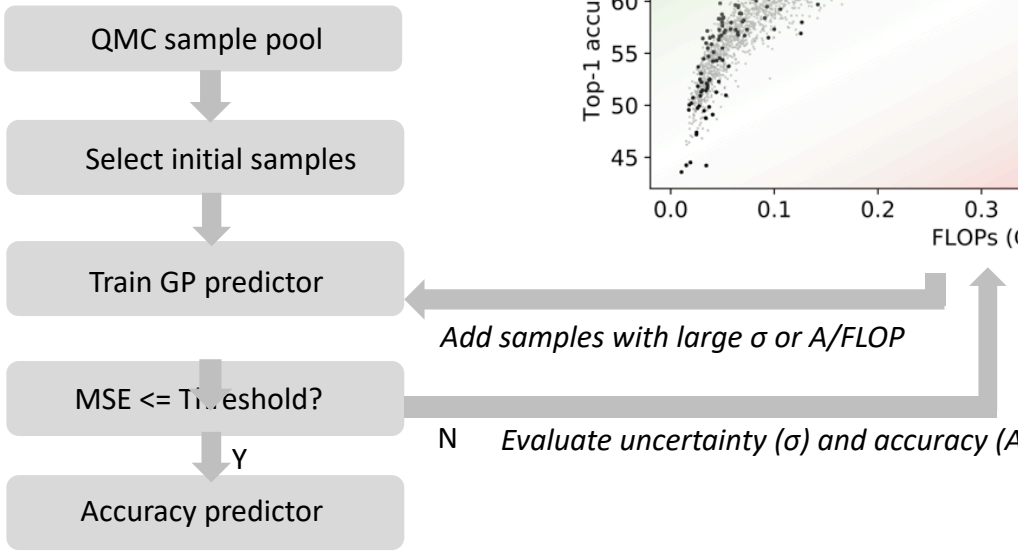
[ChamNet: Towards Efficient Network Design through Platform-Aware Model Adaptation; Dai et.al]

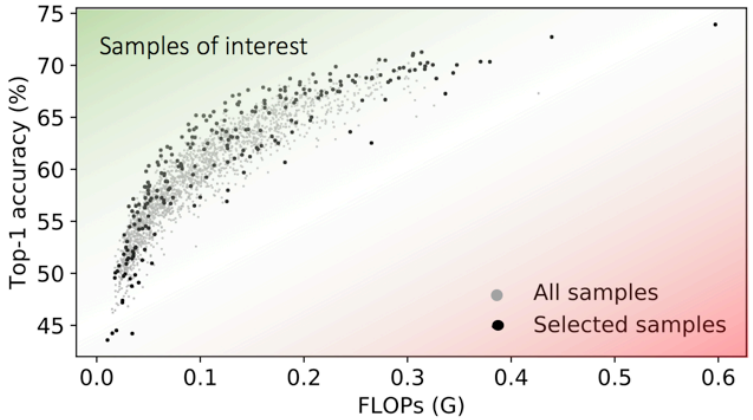
Accuracy predictor

Gaussian process (GP) accuracy predictor

Bayesian optimization for sample selection

- Exploration and exploitation
- Improved sample efficiency



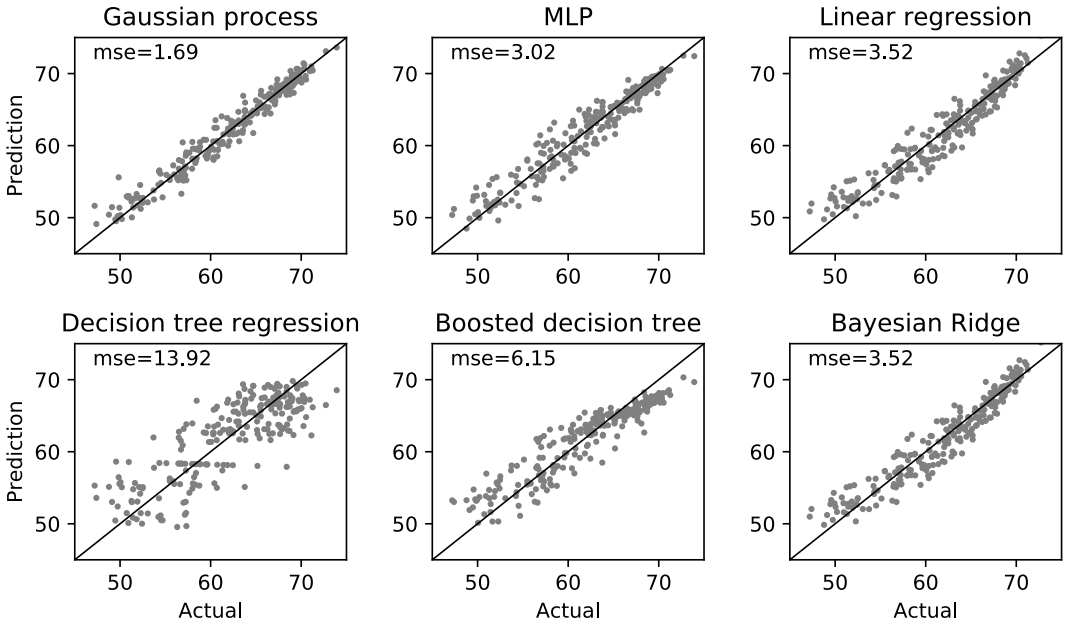


Steps involved to build an accuracy predictor

Accuracy predictor

Accuracy predictor

- Very efficient model evaluation
- No training during search



Performance comparison of different accuracy prediction models.

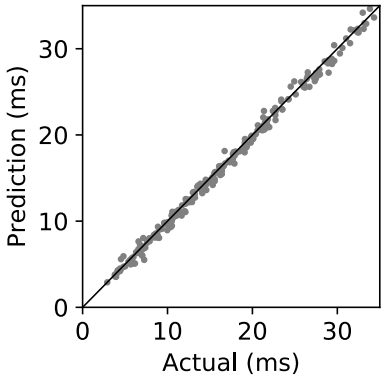
Latency and energy predictor

CNN latency look-up table (LUT)

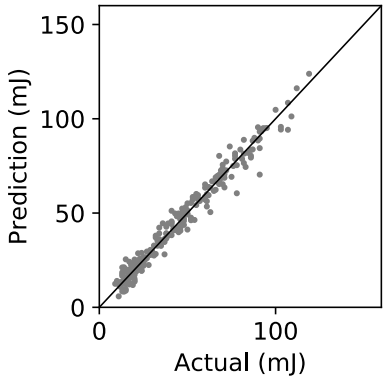
- Fast and reliable latency estimation
- Network latency = Σ operator latency

Energy predictor

- GP + Bayesian optimization
- Based on real measurements on hardware



Latency predictor

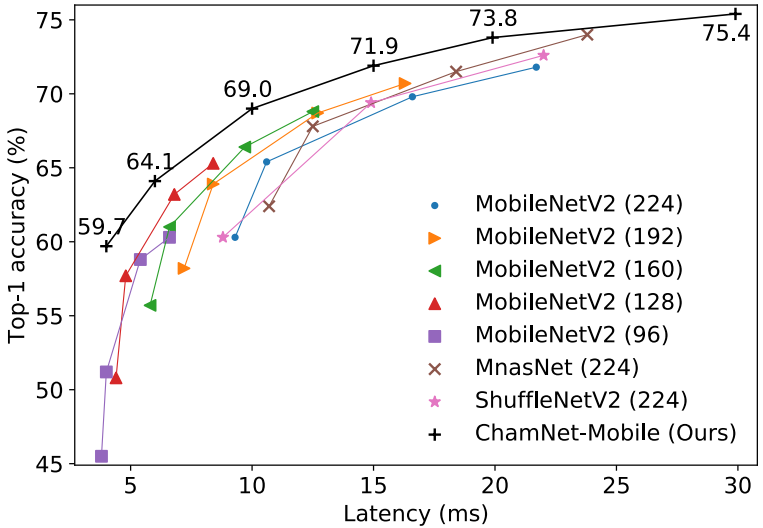


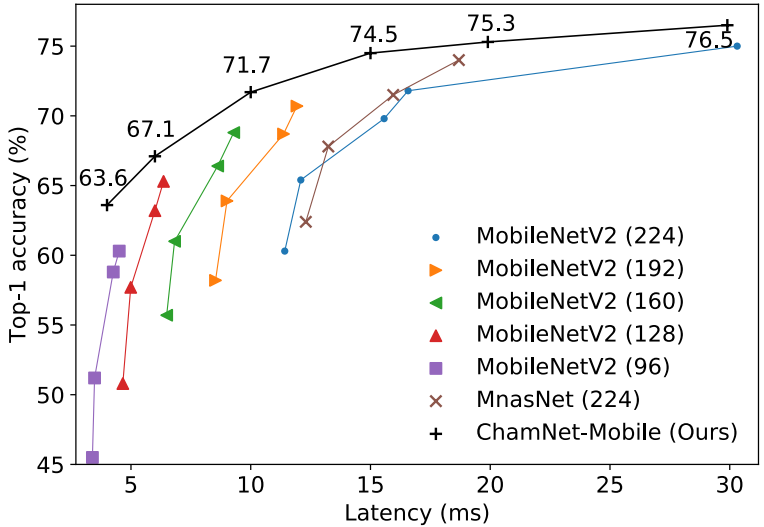
Energy predictor

Experiments on Mobile CPU and DSP

Search space: #Bottlenecks, #Filters, expansion, input resolution

- Base module: Inverted residual
- 8.5% absolute accuracy gain at 4ms compared to MobileNetV2
- 6.6% absolute accuracy gain at 10ms compared to MnasNet





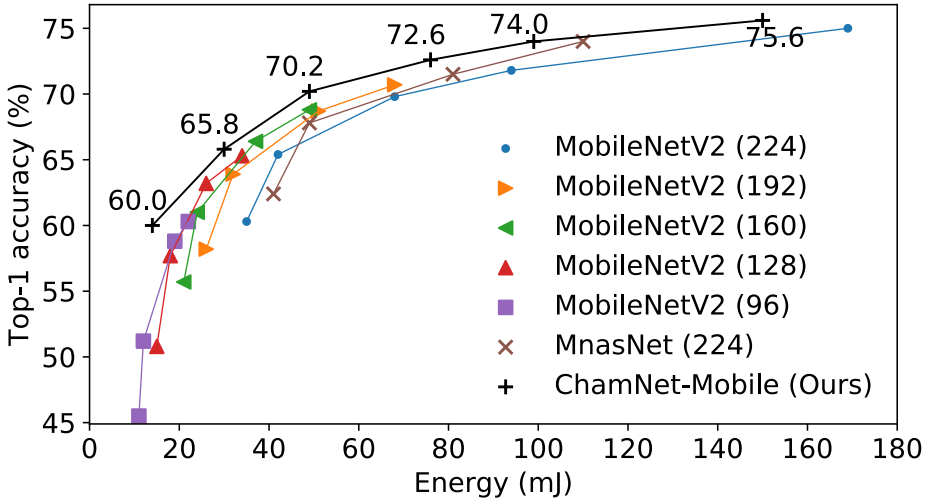
Performance of ChamNet-Mobile on (left) Snapdragon 835 CPU and (right) Hexagon v62 DSP. Numbers in parentheses indicate input image resolution.

Energy-constrained adaptation

Energy-driven adaptation (base module: Inverted residual)

Search space: #Bottlenecks, #Filters, expansion, expansion, input resolution

- Significant energy reduction
- Platform: Mobile CPU



Energy-constrained ChamNet-Mobile on a Snapdragon 835 CPU

Summary

- Challenges of efficient ConvNet design:
  - Intractable search space
  - Conditional optimality
  - Inaccurate metrics
- We proposed FBNets & ChamNet
  - Extremely fast: 8 GPUs for 24 hours, **421x** faster search
  - State-of-the-art performance: same accuracy, **1.5x faster, 2.4x smaller FLOPs**
  - **Latency and Energy based optimization** for architectures instead of FLOPs
  - Efficient architecture search for specific devices or specific tasks

Resources

Contacts

Peizhao Zhang
https://research.fb.com/people/zhang-peizhao/
stzpz@fb.com

Peter Vajda
https://research.fb.com/people/vajda-peter/
vajdap@fb.com

Apply for research intern
mv-apply@fb.com

Thank you!

References

Dai et.al

ChamNet: Towards Efficient Network Design through Platform-Aware Model Adaptation

Wu et.al

FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search

Models available: