Systematic Framework for Opportunistic Pruning of Deep Neural Networks on Edge Devices

Robert Viramontes, Cheng-En Wu, Azadeh Davoodi, Yu Hen Hu

University of Wisconsin - Madison

Introduction

In recent years, there has been interest in deploying deep neural networks (DNNs) to edge devices to reduce latency and improve privacy. However, the computational complexity of these DNNs can limit the practical ability to store and execute them on edge devices.

We also observe that SOTA models tend to be designed and trained on large, complex tasks such as 1000-class image classification. Meanwhile, many edge applications may be much more focused, such as distinguishing between dog breeds.

Various dog breeds from ImageWoof (Howard, 2019)

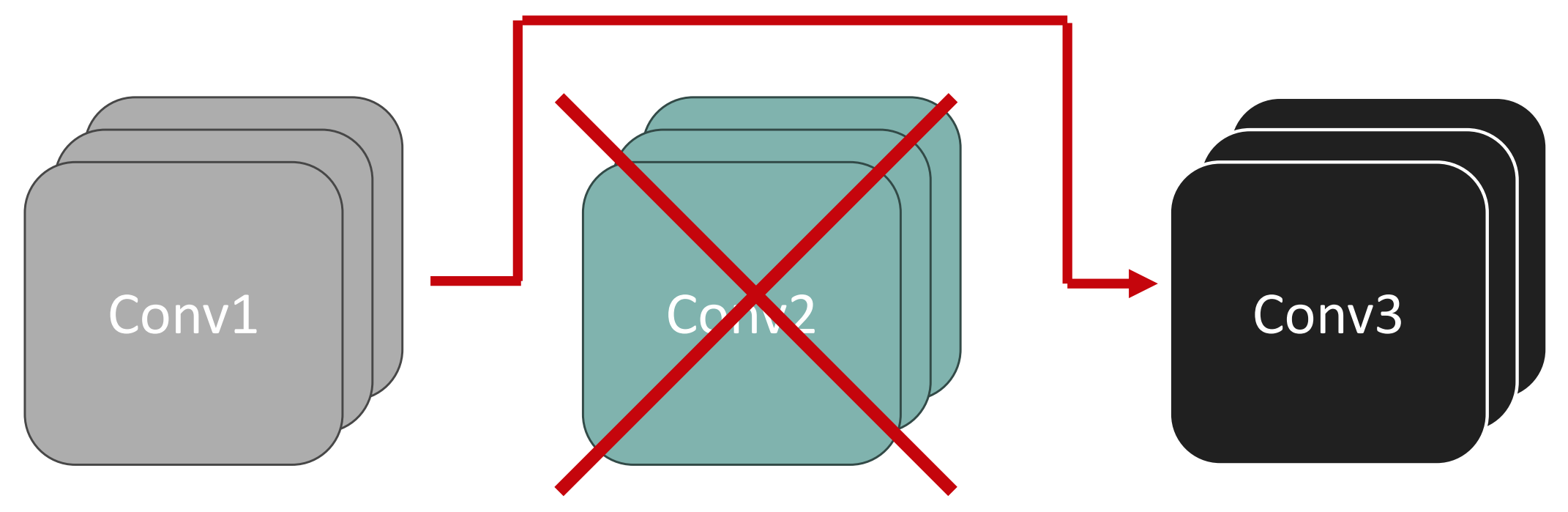
We propose a systematic, hierarchical framework `OppPrune` for opportunistic structural pruning that:

1. Allows for DNN specialization
2. Does not rely on hardware-specific optimizations such as sparsity or quantization
3. Is the first to explore iterative combination of separate structural pruning techniques

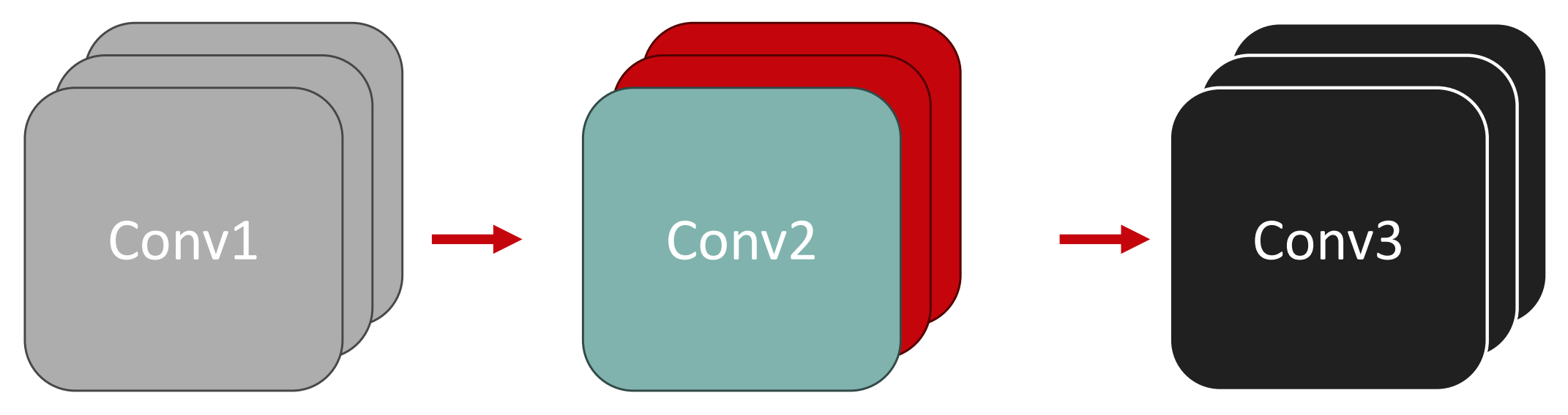
Background

We utilize two structural pruning techniques:

- Block pruning, is a coarse-grained approach that removes entire blocks



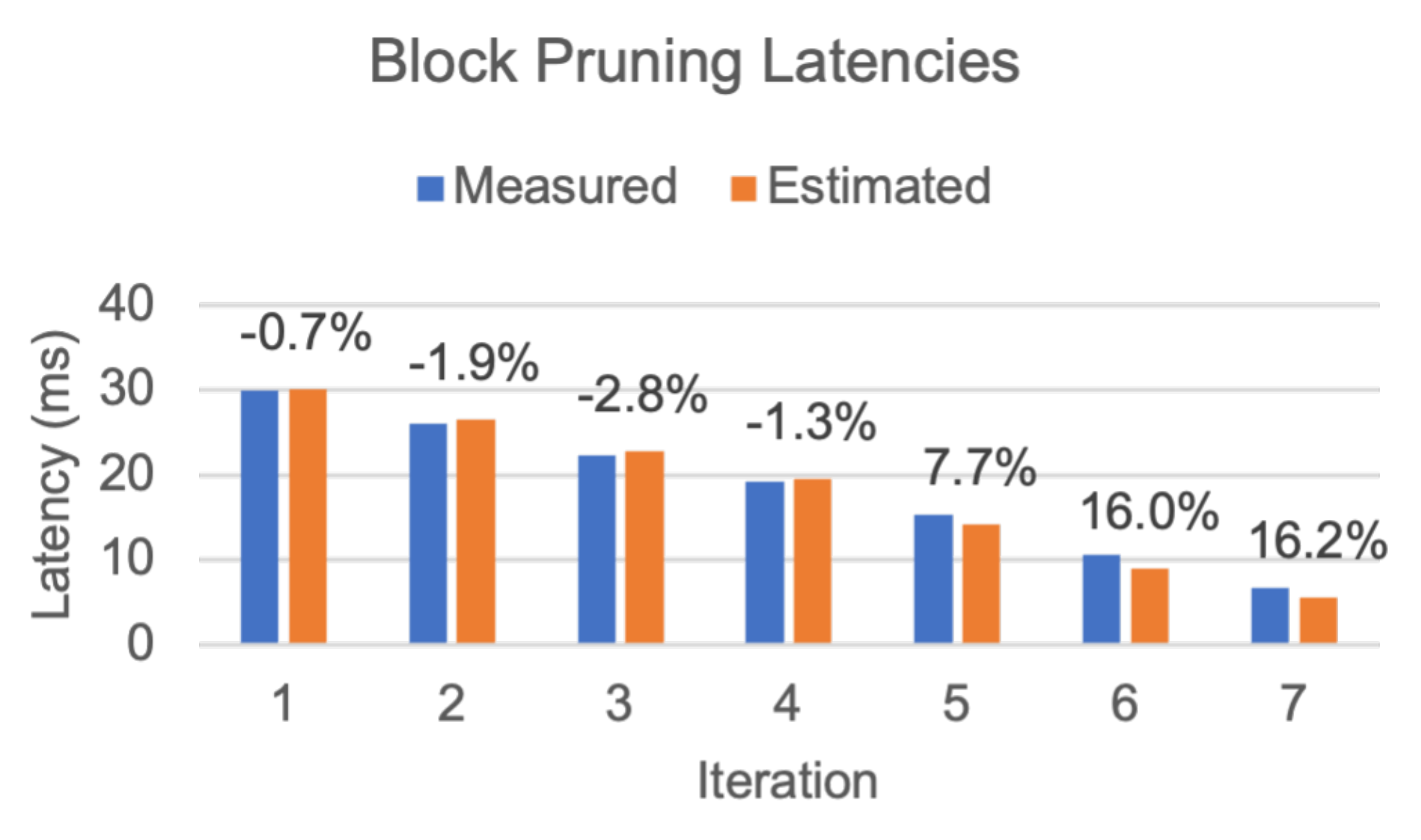
- Channel pruning, a finer-grained technique that removes particular channels from a CONV layer



Approach

Block Profiling

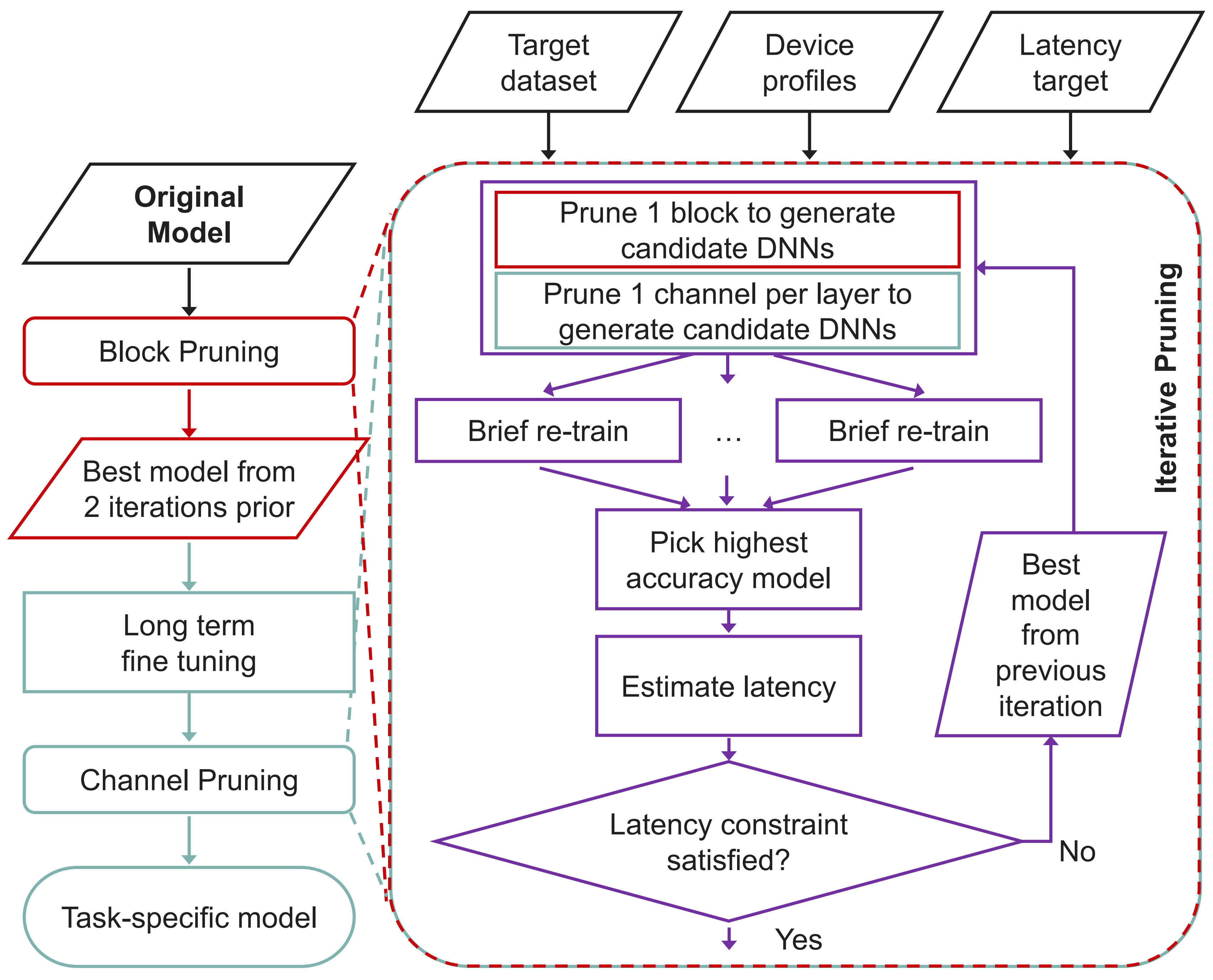
We explore the implications for latency estimation when block pruning. We find that profiling a network once, we can effectively utilize the latency of the block on the edge device. This represents a significantly lower profiling effort than for channel pruning, where we use the approach introduced by NetAdapt (*Yang et. al., 2018)* of profiling all possible configurations of layers.



`OppPrune` Iterative Framework

Our iterative framework is inspired by NetAdapt and borrows their approach to channel pruning (*Yang et. al., 2018*). We augment the framework to include our own approach to block profiling and pruning.

`OppPrune` starts with block (coarse) pruning, to rapidly move towards the latency target, removing one block at a time with the least impact on latency. Once the latency constraint is met, we recover the last still-violating network and apply channel pruning until the latency constraint is met. In this framework, we extract the best of each technique when creating a device-ready task-specific model!


Target dataset
Device profiles
Latency target
Original Model
Block Pruning
Best model from 2 iterations prior
Long term fine tuning
Channel Pruning
Task-specific model
Prune 1 block to generate candidate DNNs
Prune 1 channel per layer to generate candidate DNNs
Brief re-train
…
Brief re-train
Pick highest accuracy model
Estimate latency
Latency constraint satisfied?
No
Yes
Best model from previous iteration
Iterative Pruning

References

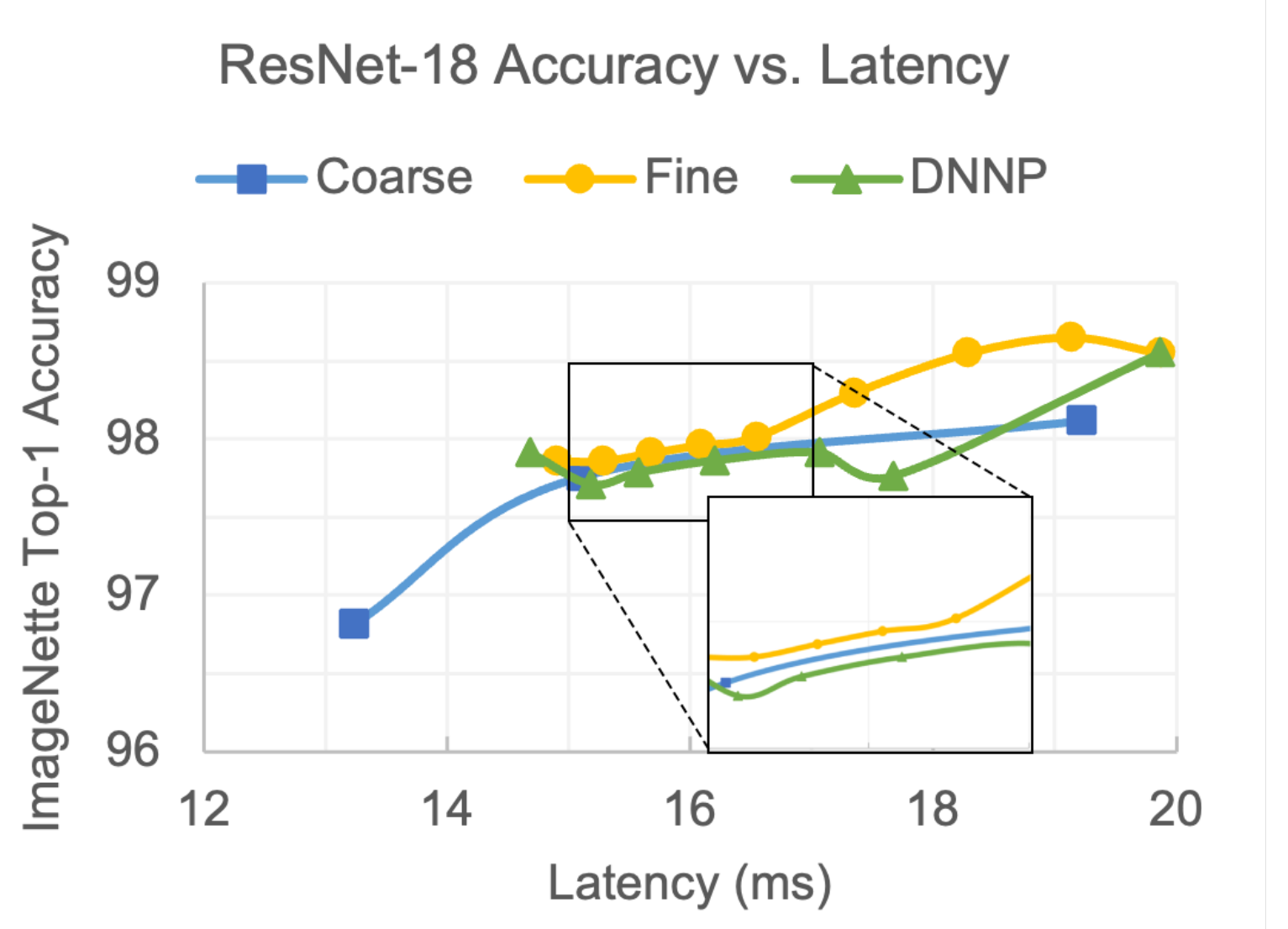
Yang, T.-J., Howard, A., Chen, B., Zhang, X., Go, A., Sandler, M., Sze, V., and Adam, H. NetAdapt: Platform-aware neural network adaptation for mobile applications. In *Proc. European Conf. on Computer Vision (ECCV)*, pp. 285–300, 2018.

Howard, J. ImageNette. Online, Dec 2019. URL https://github.com/fastai/imagenette.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.

Results



Compared to coarse-only, `OppPrune` does a better job of maintaining accuracy. Compared to fine-only, `OppPrune` is able to meet the latency target with far fewer iterations. Our framework provides a good balance between accuracy preservation and runtime efficiency for pruning a model.

	ResNext-50, ImageNette, 33ms Target		
	Latency (ms)	Top-1 Accur.	Runtime (hrs)
Original	83.94	99.95 %	n/a
Coarse	31.09	94.19 %	8.00
Fine	32.51	95.67 %	20.30
OppPrune	30.31	95.95 %	9.03

Future Directions

- Adapt profiling and pruning techniques for energy minimization targets
- Investigate impacts of device configurations such as DVFS policies on pruning
- Explore analogous systematic approaches for pruning Transformer-based networks

Conclusion

This work introduces `OppPrune`, an opportunistic DNN pruning framework that targets latency reductions for edge devices. Compared to prior works, we utilize a *coarse to fine* workflow. We reap the benefits of rapid latency reduction from block pruning with the accuracy preservation of channel pruning. Our experiments with more-specific datasets demonstrate that our approach achieves these goals, with similar accuracy to only channel pruning while keeping runtime similar to block pruning. In addition, our method does not require expert parameter tuning or development, minimizing the barrier to entry for deploying a DNN to the edge.

Acknowledgements

GitHub Copilot was enabled for a portion of code development.

This material is based upon work supported by the National Science Foundation under Grant No. 2006394.

MLSys