Dynamic Channel Execution: on-device Learning Method for Finding Compact Networks

UNIVERSITY OF CAMBRIDGE

EPSRC Engineering and Physical Sciences Research Council

Simeon E. Spasov and Pietro Liò

EPSRC Centre for Doctoral Training in Sensor Technologies & Applications

Motivation

- CNN architectures are *becoming deeper & more complex* → *higher parameter* count & floating point *operations (FLOPs).*
- Existing pruning methods *focus on* reducing computational burden during *inference* only. Pruning is a post-training technique.
- We make *training compact CNNs from scratch* feasible. Our method increases efficiency during *training* and *inference.*
- We aim to enable *training compact CNNs* on computationally and memory-constrained devices.

Overview

A. At each training iteration:
   1. Sample a training batch
   2. Select a pre-defined number of convolutional channels to activate.
   3. Run forward pass on compact model comprising active channels only.
   4. Observe utility (saliency metric) and update weights of active channels.

B. Select most salient channels

C. Fine-tune compact model comprising most salient channels

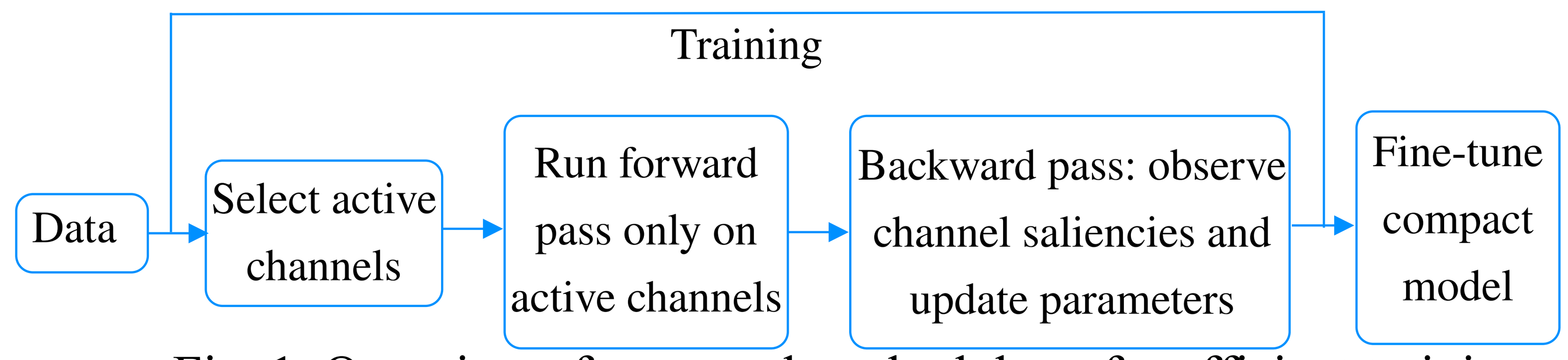

Fig. 1. Overview of proposed methodology for efficient training.

Methodology

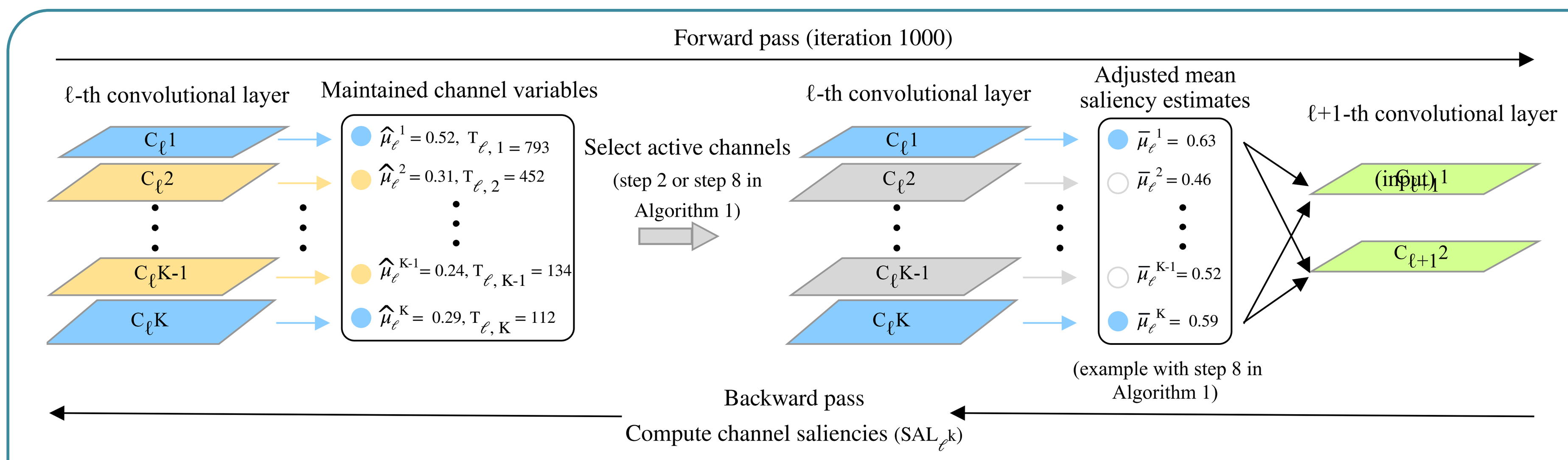

Fig. 2. The CUCB algorithm we use to drive the channel selection procedure. Visualization inspired by [3].

Estimating Channel Saliency [1]:

- Calculate *change in loss if channel is removed:*

  high change if channel removed → channel is important (highly salient)
- Use *Taylor expansion* around point where channel parameters = 0
- If we have channel C_ℓ^k, comprising M weights, which produces feature map h_ℓ^k :
- $\Delta\text{Loss} = SAL_\ell^k = \left|\frac{1}{M}\sum_{m=1}^{M}\frac{\delta Loss}{\delta h_{\ell,m}^k} h_{\ell,m}^k\right|$
- *No overhead*: Only requires gradient, which is calculated during backpropagation

Channel selection procedure

1. For each channel k in each layer ℓ maintain:

   $\mathbf{T}_{\ell,k}$ as the total number of *times the channel has been activated* so far;

   $\hat{\boldsymbol{\mu}}_\ell^k$ as the *mean of all saliency estimates* observed so far.
2. Randomly select and activate channels for τ training steps.
3. t ← τ
4. **for** training iteration j = 1…J **do:**
5. **for** batch in dataset **do:**
6. t ← t + 1
7. For each channel C_ℓ^k , set $\bar{\mu}_\ell^k = \hat{\mu}_\ell^k + \sqrt{\frac{3\ln(t)}{2T_\ell^k}}$
8. S = select top percentile of channels to activate according to $\bar{\mu}_\ell^k$
9. Run forward and backward passes through network
10. Update all $T_{\ell,k}$ and $\hat{\mu}_\ell^k$

Algorithm 1: Combinatorial Upper Confidence Bound (CUCB) algorithm [2]

Results

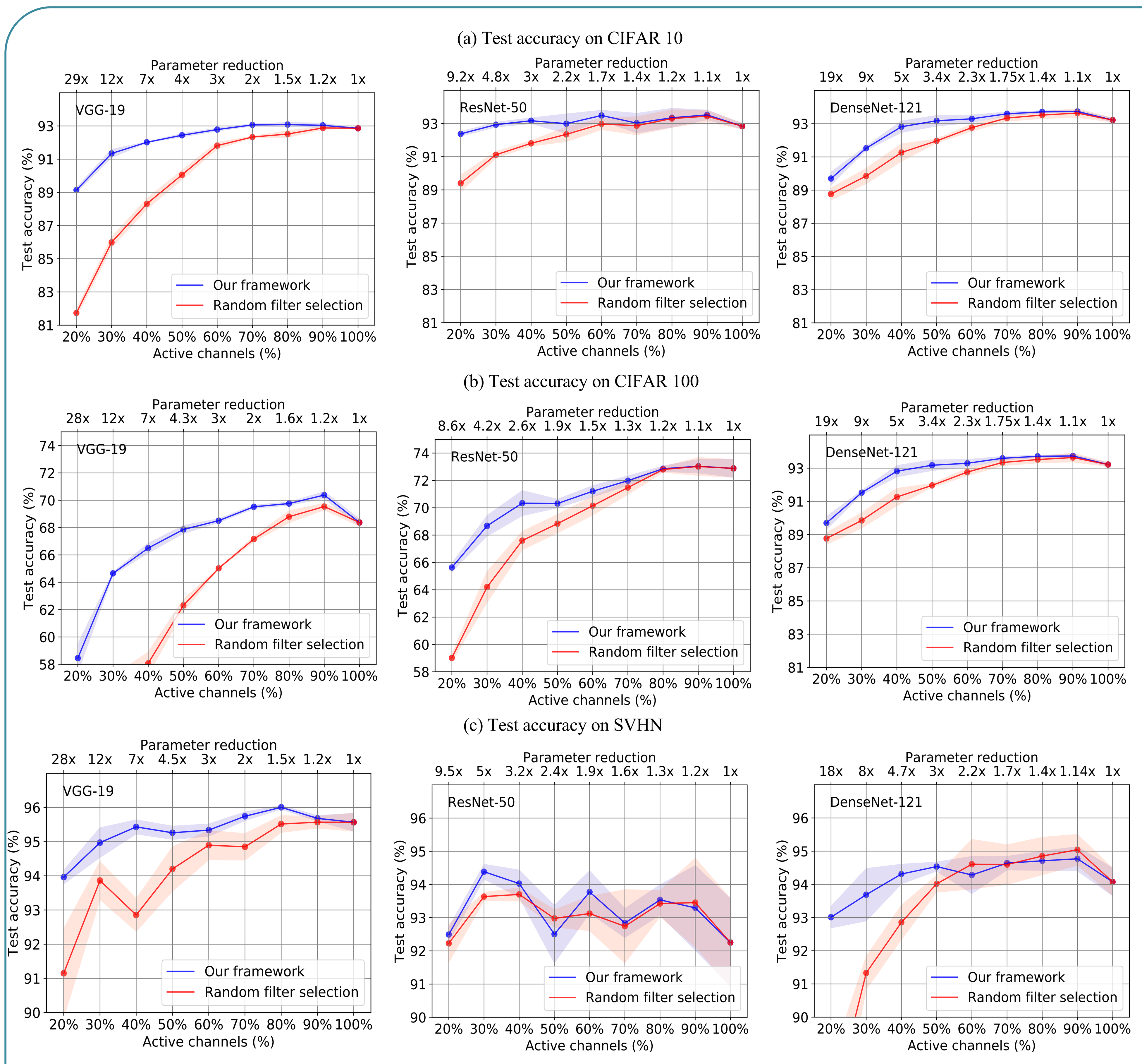

Fig. 3. Comparing proposed methodology to random channel selection.

- **Training methodology:**
  1. Train a model for 160 iterations using proposed methodology
  2. Select and activate a pre-defined number of the most salient channels
  3. For random channel selection: activate random channels for comparison
  3. Fine-tune compact network
  4. All experiments are conducted 3 times
- **Regularization effect**
  A. Peak accuracy achieved @ 70%-90% active channels
  B. 10%-50% parameter reduction @ peak accuracy
  C. 15%-30% FLOP reduction @ peak accuracy
- **Parameter and FLOP reduction**
  A. CIFAR10 & SVHN: parameter reduction 3x-7x and FLOPs reduction 2x-5x while maintaining baseline accuracy (all models).
  B. CIFAR100: 2-3% accuracy drop for compact models - high model capacity is required for 100-label classification.
- **Proposed methodology vs random channel selection**:
  A. In general our methodology outperforms random channel selection
  B. Performance difference is significant when active channels are few
- **Additional note**: Our method is based on a *channel independence assumption* & is adversely affected by skip connections → DenseNet and ResNet cannot achieve such efficiency as the sequential VGG.

1.Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, Jan Kautz. Pruning Convolutional Neural Networks for Resource Efficient Inference. In Proceedings of the 5th International Conference on Learning Representations, 2017
2.Wei Chen, Yajun Wang, Yang Yuan. Combinatorial Multi-Armed Bandit: General Framework, Results and Applications. In Proceedings of the 30th International Conference on Machine Learning (ICML), vol. 28, pp. I-151-I-159, 2013
3.Zhuang Liu et al. Learning Efficient Convolutional Networks through Network Slimming. In proceedings of the International Conference on Computer Vision (ICCV) 2017, Venice, Italy.

The 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019