Practical Network Acceleration with Tiny Sets

Guo-Hua Wang, Jianxin Wu

State Key Laboratory for Novel Software Technology, Nanjing University

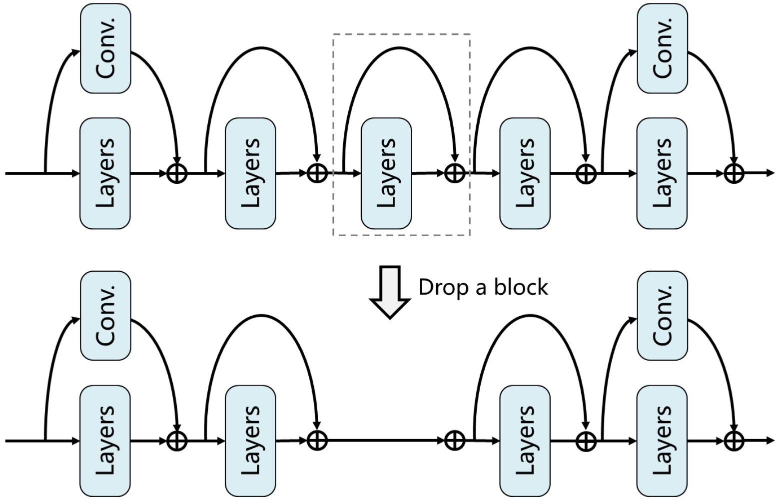
Presentation Date: June 22, 2023

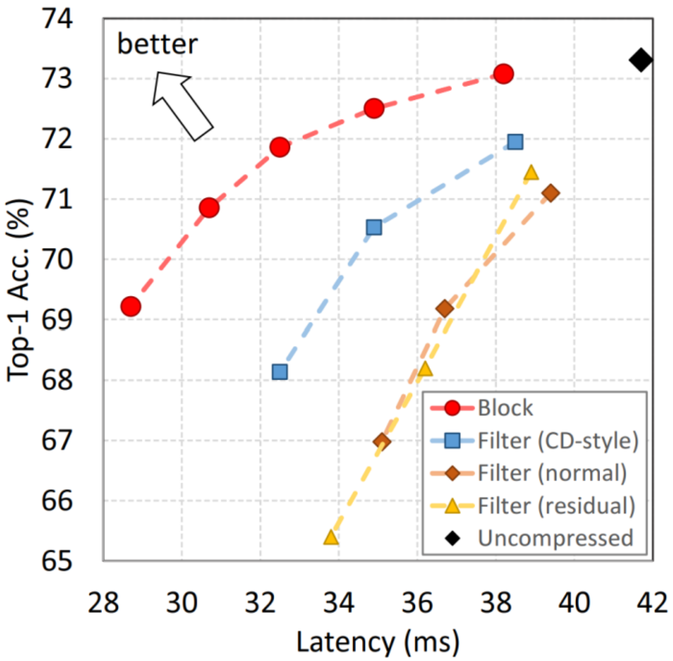
Poster Number: 366

Tag: THU-AM-366

Summary

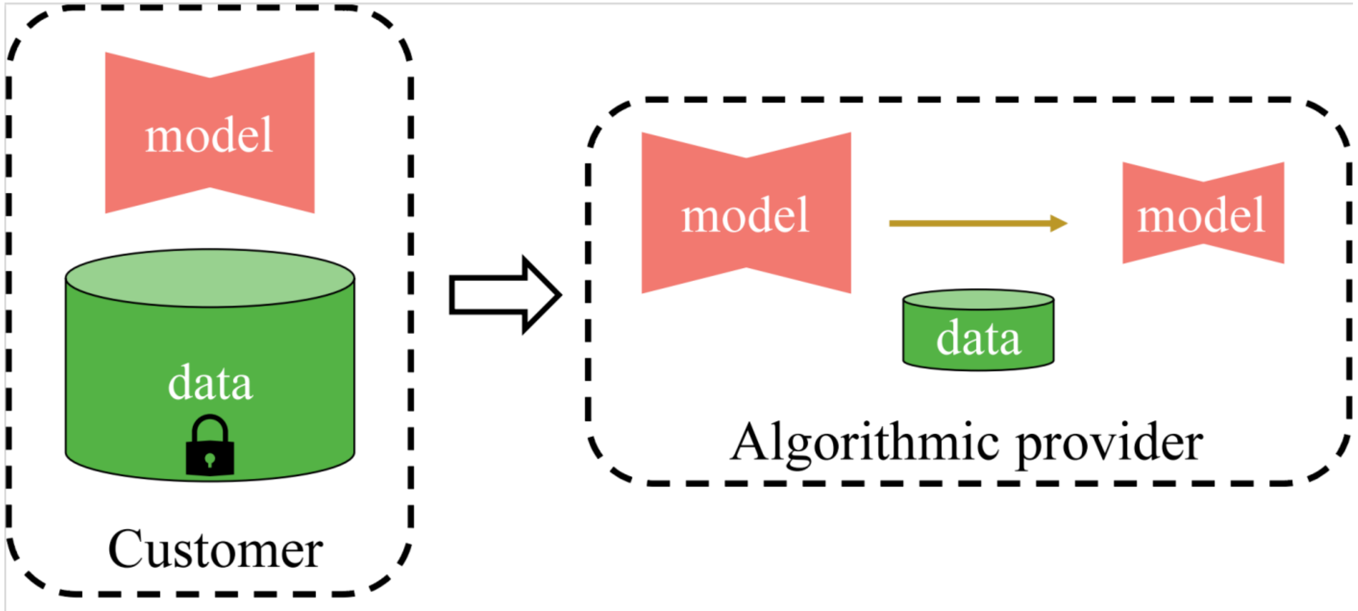
- Problem
  - How to accelerate deep networks (CNNs) with a tiny training set (50~1000 images)?
- The proposed method
  - Drop blocks: an embarrassingly simple but powerful few-shot compression method
  - Recoverability: measure the difficulty of recovering each block, and in determining the priority to drop blocks.
  - PRACTISE: the algorithm for accelerating networks





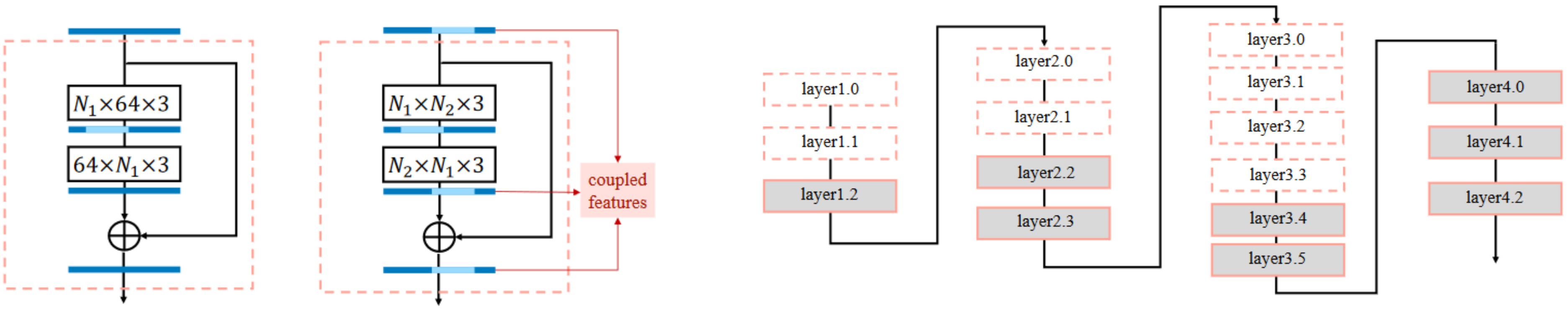
Few-shot compression

- Network compression
  - The original training set
- Few-shot compression
  - To preserve data privacy and/or to achieve fast deployment
  - A tiny training set



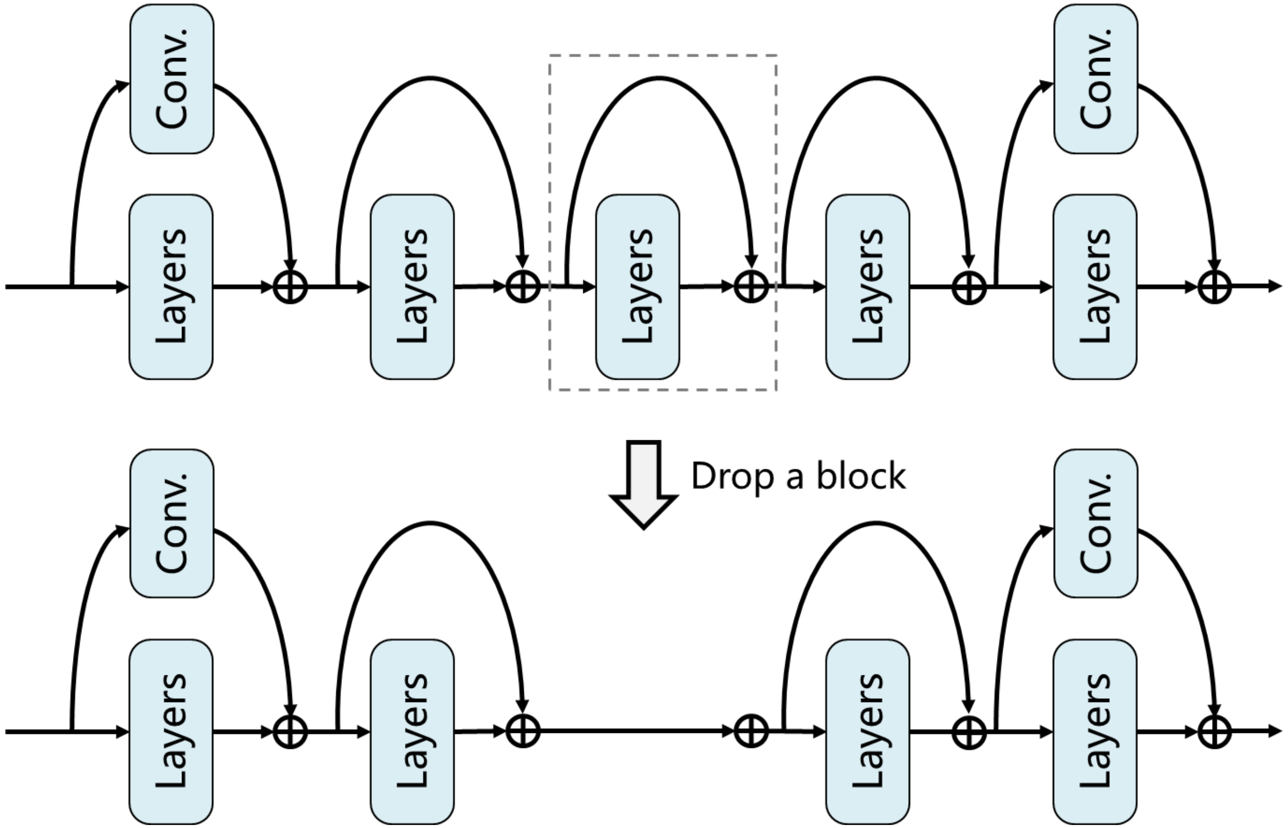
Related works and their problems

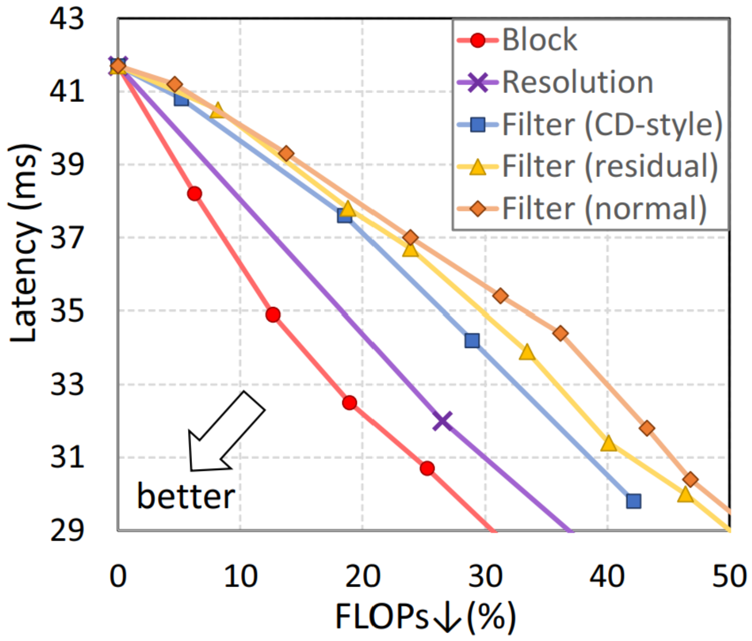
- Pruning filters
  - Suffer from a low acceleration ratio
  - Need to reduce lots of FLOPs
  - Require lots of training data
- Focus on the FLOPs-accuracy tradeoff and neglect the latency-accuracy tradeoff



b) Normal c) Residual d) CD-style

Drop blocks

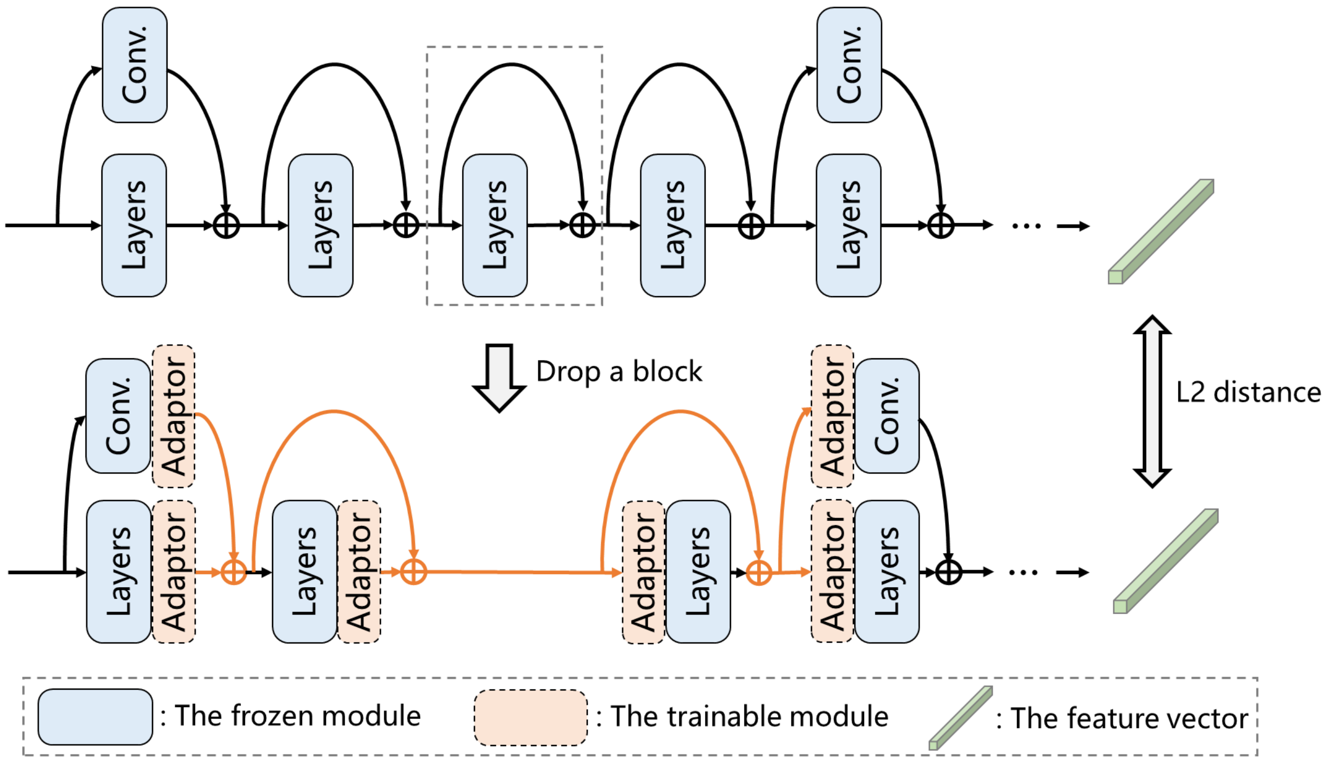


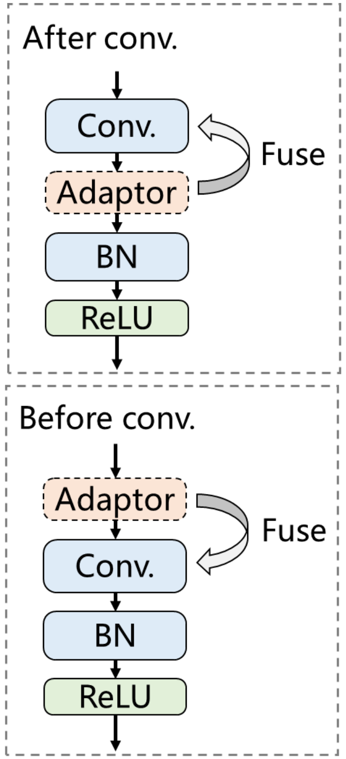


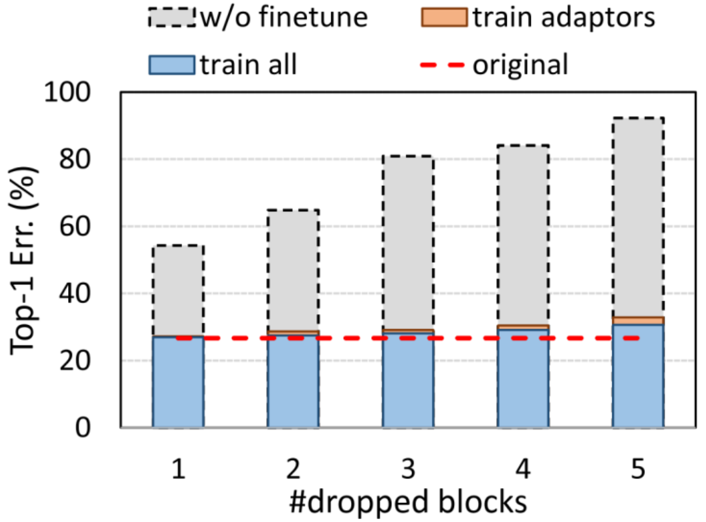
KD [10]	FSKD [12]	CD [1]	MiR [30]	BP (blocks)
44.5	45.3	56.2	64.1	**66.5**

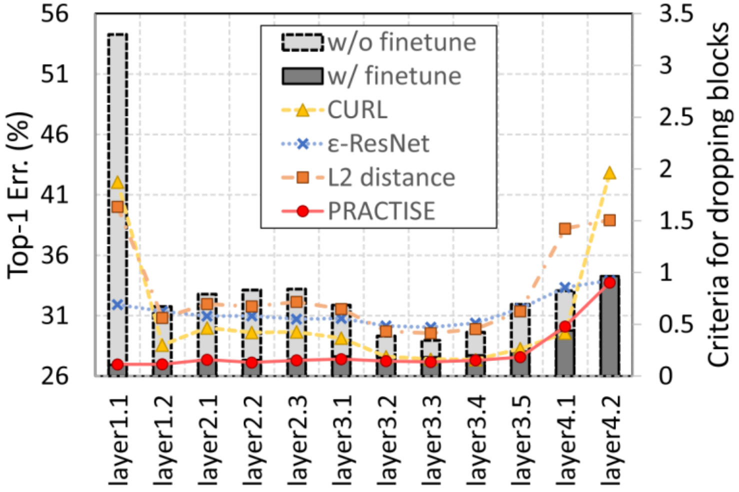
- Accelerate networks by dropping blocks
  - High acceleration ratio
  - High accuracy

The recoverability









- The framework to compute the recoverability
  - Inserting adaptors to recover the performance
  - Consistent with the finetuned accuracy

PRACTISE

- PRACTISE: Practical network acceleration with tiny sets of images

---

Algorithm 1: PRACTISE

Input: The original model $\mathcal{M}_O$, the number of dropped blocks k, the tiny training data $\mathcal{D}_\mathcal{T}$

Test the latency of $\mathcal{M}_O$;

for *each block* $\mathcal{B}_i$ **do**

  Drop $\mathcal{B}_i$ to obtain the pruned model $\mathcal{M}_{P(\mathcal{B}_i)}$;

  Test latency of $\mathcal{M}_{P(\mathcal{B}_i)}$ and find $\tau(\mathcal{B}_i)$ (Eq. 2);

  Insert adaptors;

  Compute $\mathcal{R}(\mathcal{B}_i)$ with $\mathcal{D}_\mathcal{T}$ (Eq. 1);

  Compute the score $s(\mathcal{B}_i)$ (Eq. 3);

  Add $\mathcal{B}_i$ back and remove all adaptors;

Choose the top k blocks with the minimum scores;

Drop these k blocks to obtain $\mathcal{M}_P$;

Finetune $\mathcal{M}_P$ with $\mathcal{D}_\mathcal{T}$ by minimizing $\mathcal{L}$ (Eq. 4);

return The pruned model $\mathcal{M}_P$

$$\tau(\mathcal{B}_i) = \frac{lat_{\mathcal{M}_O} - lat_{\mathcal{M}_{P(\mathcal{B}_i)}}}{lat_{\mathcal{M}_O}}\,, \tag{2}$$

$$\mathcal{R}(\mathcal{B}_i) = \min_{\alpha} \mathbb{E}_{x\sim p(x)} \|\mathcal{M}_O(x;\theta) - \mathcal{M}_{P(\mathcal{B}_i)}(x;\theta\backslash b_i,\alpha)\|_F^2\,, \tag{1}$$

$$s(\mathcal{B}_i) = \frac{\mathcal{R}(\mathcal{B}_i)}{\tau(\mathcal{B}_i)}\,. \tag{3}$$

$$\mathcal{L} = \|\mathcal{M}_O(x;\theta_O) - \mathcal{M}_P(x;\theta_P)\|_F^2\,, \tag{4}$$

Experiments

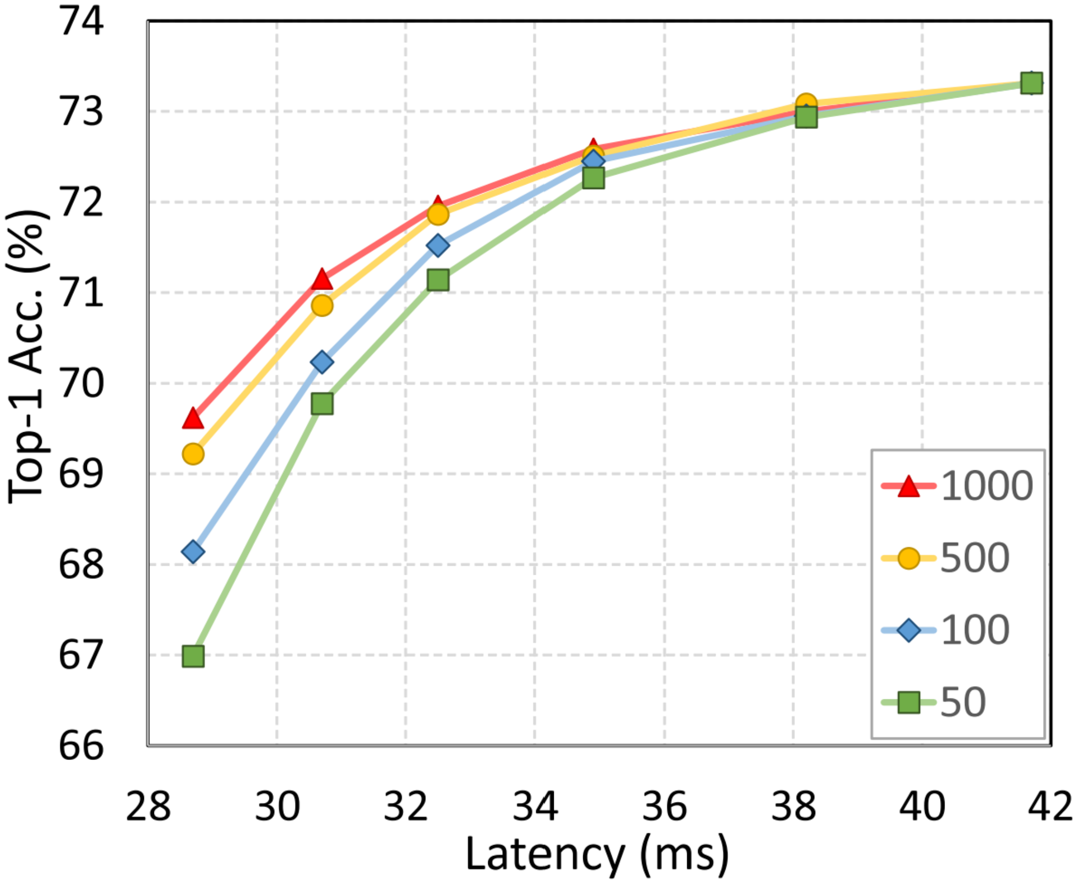
- Accelerate ResNet-34 on ImageNet-1k with tiny sets

Method	Latency (ms)	50	100	500	1000
BP (filter)	35.1 (15.8%↓)	$39.0_{\pm1.41}/68.9_{\pm1.17}$	$41.0_{\pm0.33}/70.5_{\pm0.66}$	$51.8_{\pm0.30}/78.1_{\pm0.38}$	$57.8_{\pm0.30}/81.5_{\pm0.18}$
BP (block)	**34.9 (16.3%↓)**	$\mathbf{66.5}_{\pm0.81}/\mathbf{78.4}_{\pm0.44}$	$\mathbf{66.8}_{\pm0.23}/\mathbf{87.7}_{\pm0.23}$	$\mathbf{68.6}_{\pm0.18}/\mathbf{88.8}_{\pm0.09}$	$\mathbf{69.8}_{\pm0.12}/\mathbf{89.3}_{\pm0.07}$
KD [10]	35.1 (15.8%↓)	$44.5_{\pm1.20}/72.3_{\pm0.87}$	$46.4_{\pm0.34}/74.0_{\pm0.58}$	$54.7_{\pm0.26}/79.7_{\pm0.19}$	$57.9_{\pm0.21}/81.6_{\pm0.12}$
FSKD [12]	35.1 (15.8%↓)	$45.3_{\pm0.77}/71.5_{\pm0.62}$	$51.2_{\pm0.30}/76.8_{\pm0.23}$	$57.6_{\pm0.21}/81.6_{\pm0.15}$	$59.4_{\pm0.13}/82.7_{\pm0.06}$
CD [1]	35.1 (15.8%↓)	$56.2_{\pm0.37}/80.8_{\pm0.31}$	$59.1_{\pm0.22}/82.8_{\pm0.11}$	$63.7_{\pm0.18}/86.0_{\pm0.05}$	$64.4_{\pm0.03}/86.3_{\pm0.07}$
MiR [30]	35.1 (15.8%↓)	$64.1_{\pm0.10}/86.3_{\pm0.11}$	$65.1_{\pm0.19}/87.0_{\pm0.11}$	$67.0_{\pm0.09}/88.1_{\pm0.07}$	$67.8_{\pm0.06}/88.5_{\pm0.02}$
PRACTISE	**34.9 (16.3%↓)**	$\mathbf{70.3}_{\pm0.16}/\mathbf{89.6}_{\pm0.06}$	$\mathbf{71.5}_{\pm0.74}/\mathbf{90.3}_{\pm0.37}$	$\mathbf{72.5}_{\pm0.04}/\mathbf{90.9}_{\pm0.03}$	$\mathbf{72.5}_{\pm0.05}/\mathbf{91.0}_{\pm0.02}$

Method	Latency (ms)	50	100	500	1000
BP (filter)	33.8 (18.9% ↓)	$24.2_{\pm0.92}/52.7_{\pm1.36}$	$27.6_{\pm0.41}/56.7_{\pm0.62}$	$42.9_{\pm0.28}/70.5_{\pm0.27}$	$51.2_{\pm0.32}/76.5_{\pm0.16}$
BP (block)	**32.5 (22.1% ↓)**	$\mathbf{60.6}_{\pm0.62}/\mathbf{83.5}_{\pm0.42}$	$\mathbf{61.6}_{\pm0.31}/\mathbf{84.3}_{\pm0.36}$	$\mathbf{65.0}_{\pm0.19}/\mathbf{86.5}_{\pm0.20}$	$\mathbf{66.8}_{\pm0.18}/\mathbf{87.5}_{\pm0.13}$
KD [10]	33.8 (18.9% ↓)	$30.1_{\pm0.69}/57.7_{\pm1.10}$	$33.1_{\pm0.43}/61.0_{\pm0.53}$	$45.7_{\pm0.26}/72.2_{\pm0.25}$	$50.5_{\pm0.29}/75.9_{\pm0.23}$
FSKD [12]	33.8 (18.9% ↓)	$31.1_{\pm0.90}/56.5_{\pm1.10}$	$36.6_{\pm0.44}/63.1_{\pm0.46}$	$42.8_{\pm0.49}/69.1_{\pm0.58}$	$44.9_{\pm0.20}/70.5_{\pm0.29}$
MiR [30]	33.8 (18.9% ↓)	$59.9_{\pm0.30}/83.2_{\pm0.31}$	$62.1_{\pm0.22}/84.8_{\pm0.18}$	$65.4_{\pm0.07}/87.0_{\pm0.03}$	$66.6_{\pm0.05}/87.7_{\pm0.04}$
PRACTISE	**32.5 (22.1% ↓)**	$\mathbf{68.0}_{\pm1.36}/\mathbf{88.2}_{\pm0.77}$	$\mathbf{70.4}_{\pm0.42}/\mathbf{89.7}_{\pm0.23}$	$\mathbf{71.8}_{\pm0.07}/\mathbf{90.5}_{\pm0.02}$	$\mathbf{71.9}_{\pm0.05}/\mathbf{90.6}_{\pm0.04}$

Experiments

- The data-latency-accuracy tradeoff
- Accelerate MobileNetV2 on ImageNet-1k with tiny sets



Method	Latency (ms)	Top-1/Top-5
Original	37.6	71.9/90.3
BP (filter)	31.5 (16.2% ↓)	$45.0_{\pm0.34}/71.8_{\pm0.38}$
KD [10]	31.5 (16.2% ↓)	$48.4_{\pm0.34}/73.9_{\pm0.32}$
MiR [30]	31.5 (16.2% ↓)	$67.6_{\pm0.05}/87.9_{\pm0.04}$
PRACTISE	**30.4** (**19.1**% ↓)	$\mathbf{69.3}_{\pm0.05}/\mathbf{88.9}_{\pm0.05}$
BP (filter)	34.1 (9.3% ↓)	$55.5_{\pm0.16}/80.3_{\pm0.26}$
KD [10]	34.1 (9.3% ↓)	$59.1_{\pm0.17}/82.5_{\pm0.15}$
MiR [30]	34.1 (9.3% ↓)	$69.7_{\pm0.04}/89.2_{\pm0.03}$
PRACTISE	**31.9** (**15.2**% ↓)	$\mathbf{70.3}_{\pm0.03}/\mathbf{89.5}_{\pm0.03}$

Experiments

- Zero-shot compression
- Accelerate ResNet-34 on out-of-domain training datasets

Network	Method	Pruning	Latency	Top-1
ResNet-50	Original		83.8	76.1
	DI [33]	filter	-	72.0
	MixMix [14]	filter	-	69.8
	ADI [33]	filter	-	73.3
	ADI* [33]	filter	79.9 (4.7%↓)	73.5
	PRACTISE	block	**66.2 (21.0%↓)**	**74.8**
MobileNetV2	Original		37.6	71.9
	DI [33]	filter	-	15.3
	MixMix [14]	filter	-	42.5
	ADI* [33]	filter	30.8 (18.1%↓)	62.8
	PRACTISE	block	**30.4 (19.1%↓)**	**68.0**

Dateset	50	500	1000	5000	All
ImageNet [26]	74.22	74.58	74.58	75.14	75.24
ADI [33]	69.85	72.68	73.01	74.40	74.79
CUB [28]	72.49	73.71	73.94	74.86	74.92
Place365 [36]	**72.80**	**74.10**	**74.18**	**75.05**	**75.21**

Conclusions

- Argue that the FLOPs-accuracy tradeoff is a misleading metric for few-shot compression, and advocate that the latency-accuracy tradeoff is more crucial in practice.
- The first to reveal dropping blocks great potential in few-shot compression.
- Propose a new concept recoverability to measure the difficulty of recovering each block, and in determining the priority to drop blocks.
- Propose PRACTISE, an algorithm for accelerating networks with tiny sets of images.
- The extraordinary performance: For 22.1% latency reduction, PRACTISE surpasses the previous state-of-the-art (SOTA) method on average by 7.0%.

Thank you!

https://arxiv.org/abs/2202.07861

https://github.com/DoctorKey/Practise