Convolution with even-sized kernels and symmetric padding

Shuang Wu, Guanrui Wang, Pei Tang, Feng Chen, Luping Shi

INNOVATION CENTER FOR FUTURE CHIPS
未来芯片技术高精尖创新中心

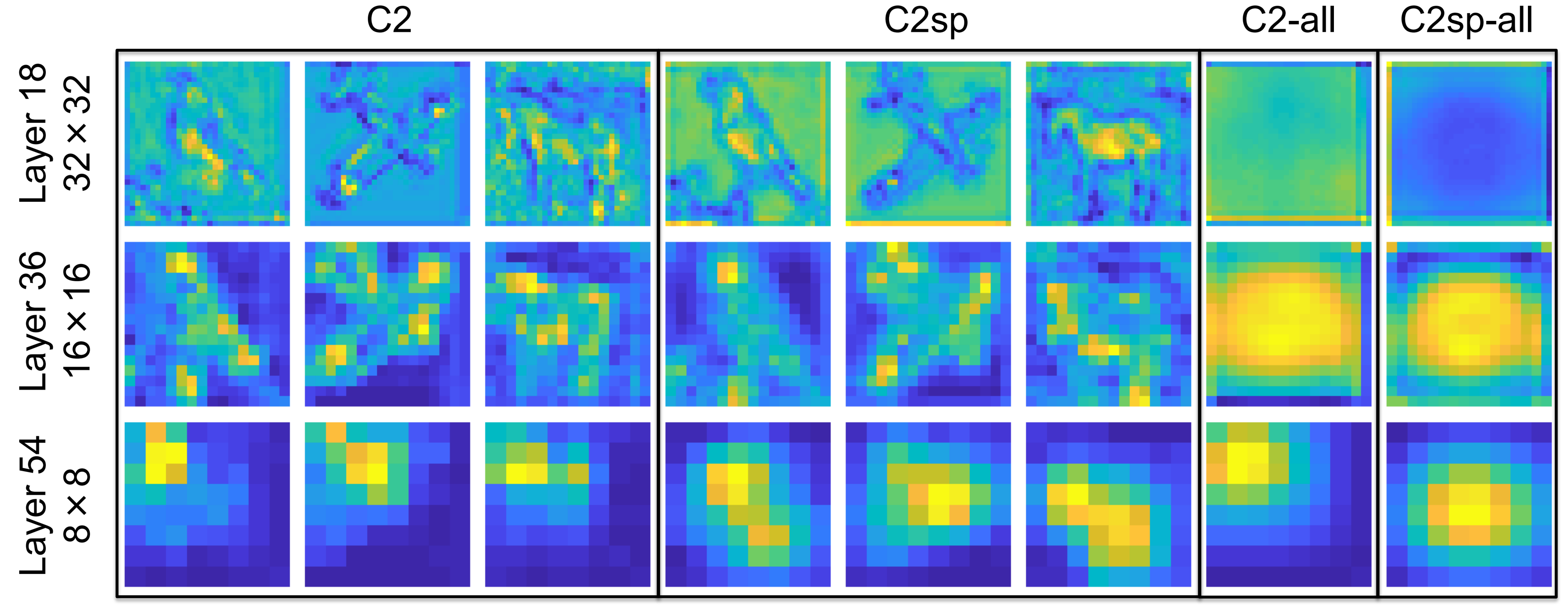
Paper Our Lab

TL;DR

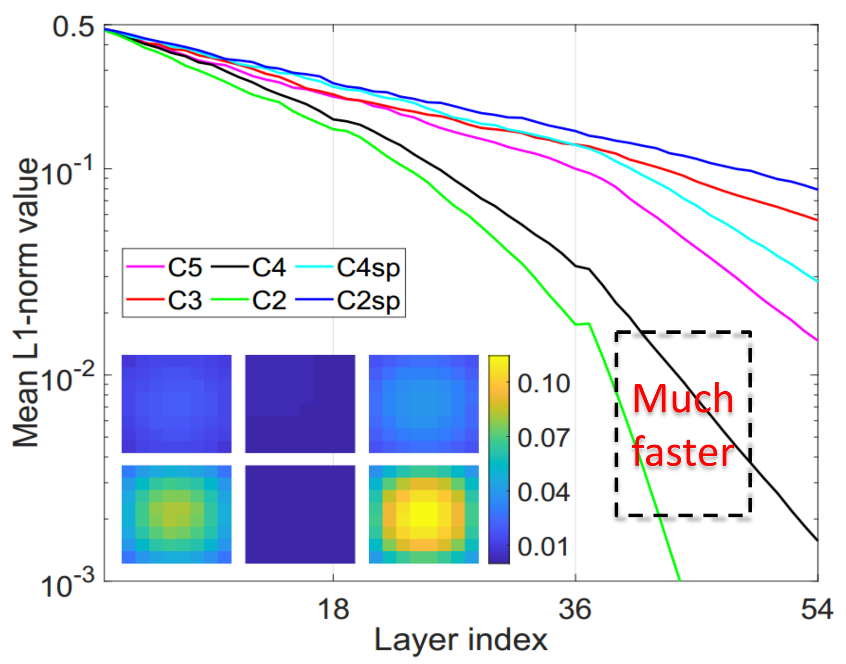
- Exploration of convolution kernel sizes (2×2, 3×3, 4×4, 5×5)
- Quantify the shift problem in even-sized kernels (2×2, 4×4)
- Proposing symmetric padding convolutions (C2sp, C4sp)
- The advantages of this approach:
  - \+ Neat and embeddable units for most CNN models
  - \+ Improved classification accuracy (CIFAR, ImageNet)
  - \+ Improved quality of GANs (CIFAR, LSUN, CelebA)
  - − Less parameter, FLOPs, training memory and time

Problem Formulation

- Even-sized kernels: rarely discussed, performance degradation
  - No central point, resulting in 0.5 pixel shift for each convolution layer.
  - The shift is cumulative: $\mathcal{F}_n\left[\boldsymbol{p}-\left(\frac{n}{2},\frac{n}{2}\right)\right] \xleftarrow{\text{approx}} \mathcal{F}_0(\boldsymbol{p})$

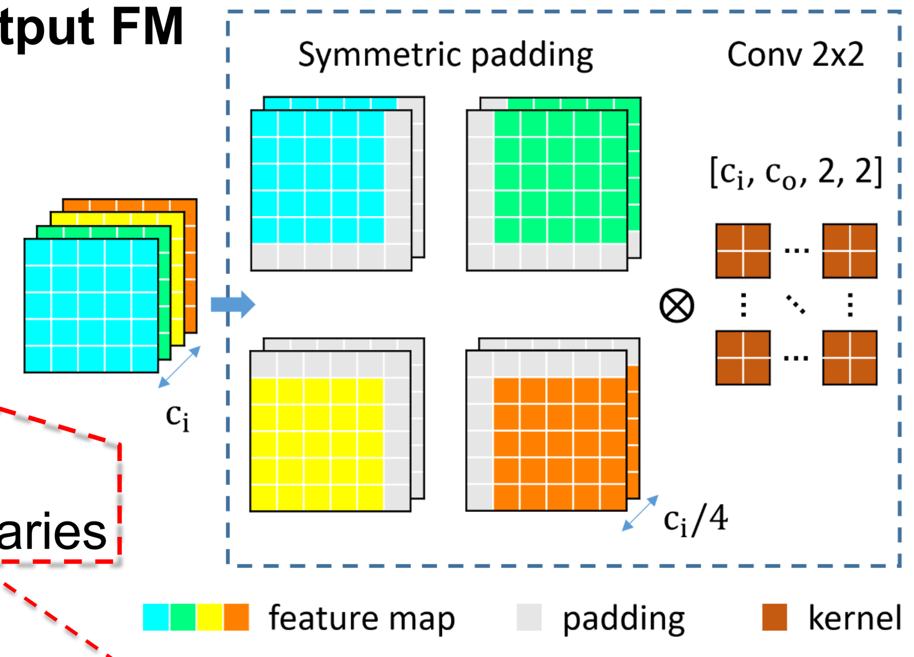


- The information erosion of padding
  - Like an ice chip melting in the water
  - zero-padding erode the information
  - $Q_n = \frac{1}{hw}\sum_{\boldsymbol{p}\in h\times w}|\mathcal{F}_n(\boldsymbol{p})|$, $Q_n < Q_{n-1}$
  - **Asymmetry accelerates the erosion**
  - Provides explanations for degradation in very deep neural networks (ResNet paper)

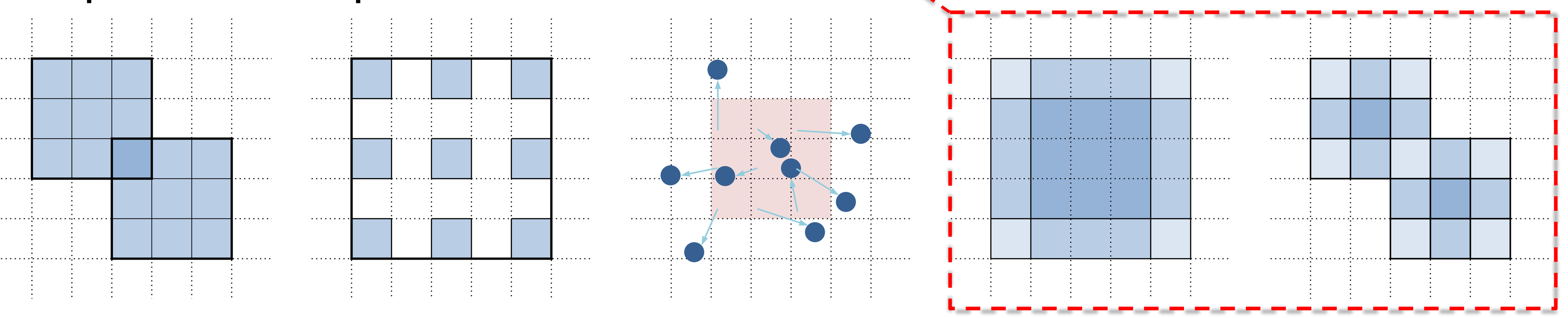


Symmetric Padding

- Introducing symmetry by group-padding strategy
  - **Asymmetry input FM → Symmetry output FM**
  - $\sum_{\delta\in\mathcal{R}}\boldsymbol{\delta} \neq (0,0) \rightarrow \sum_{i=1}^{c_i}\sum_{\delta\in\pi(i)}\boldsymbol{\delta} = (0,0)$
  - $\pi(i)$: padding policy for input channel i
  - $\mathcal{R} \rightarrow \mathcal{R}_+ = \{\mathcal{R}_{\mathrm{LT}}, \mathcal{R}_{\mathrm{LB}}, \mathcal{R}_{\mathrm{RT}}, \mathcal{R}_{\mathrm{RB}}\}$
  - \+ None Params/FLOPs
  - \+ Highly extendable (other kernel size)
  - \+ Neat implementation in computation libraries

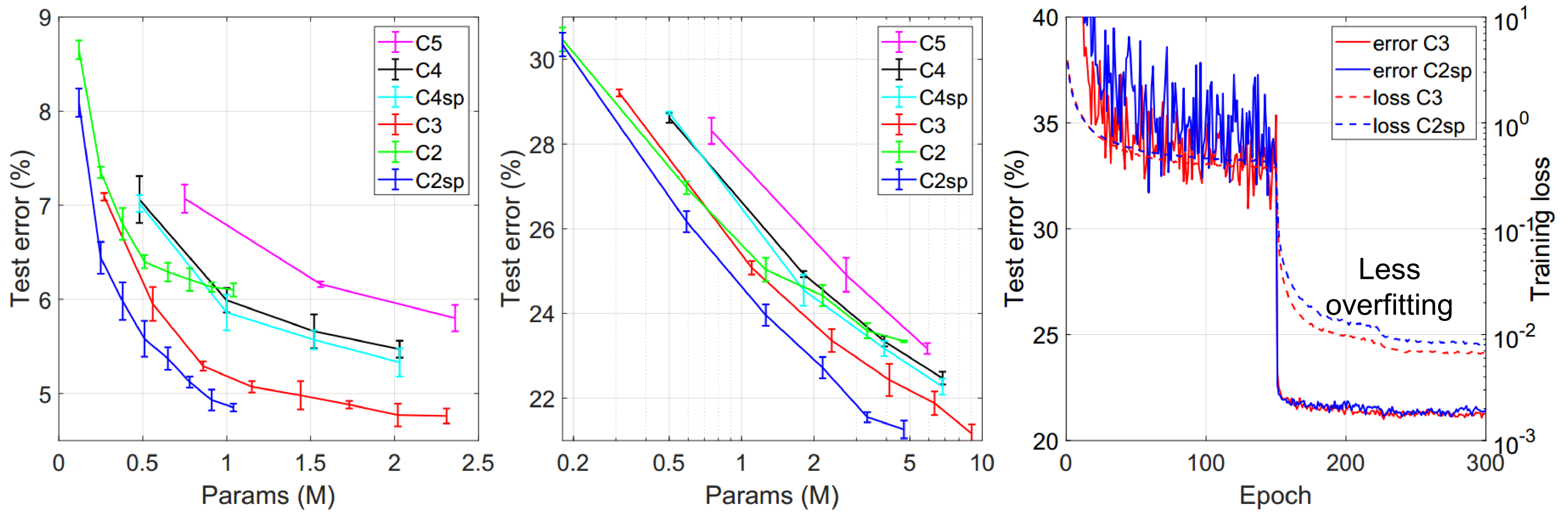


- Expanded reception fields



Exploration of Kernel Sizes

- Model: ResNet & DenseNet, Dataset: CIFAR10/100
  - Only replace each C3 by a C2, C4, C5, C2sp or C4sp
  - C2 is inferior to C3 and saturate much faster as the network deepens
  - **C2sp save 30%-50% parameters and FLOPs compared with C3**
  - Edge effect dominate the information erosion (C4, C5)

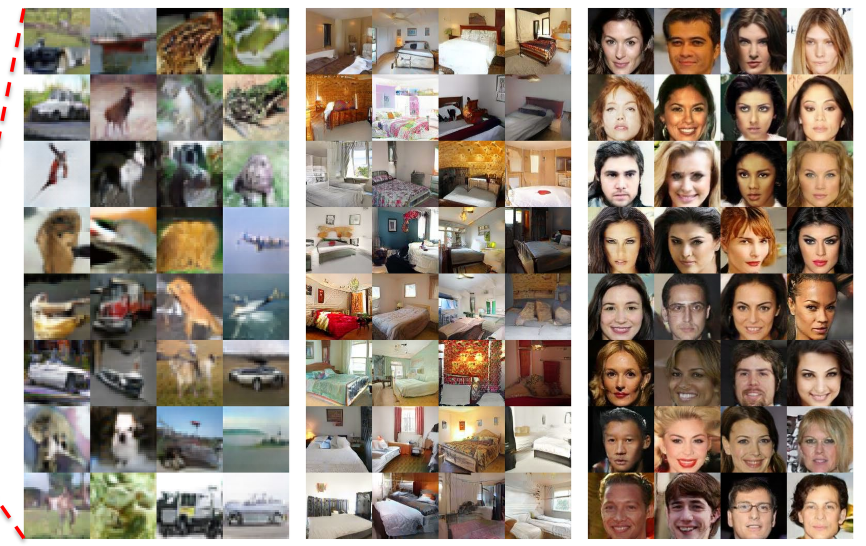


Improved Performance

- Classification results:

Model (ImageNet)	Error (%)	Params (M)	FLOPs (M)
ResNet-50 C3	**23.8**	25.5	4089
ResNet-50 C2sp	24.0	19.3	3062
DenseNet-121 C3	24.6	8.0	2834
DenseNet-121 C2sp	25.1	**6.7**	**2143**
ResNet-50 0.5x C3	**26.8**	6.9	1127
ResNet-50 0.5x C2	29.8	5.3	870
ResNet-50 0.5x C2sp	27.3	**5.3**	**870**
ResNet-50 0.5x C2sp optim	26.8	**5.8**	**573**
ShuffleNet v2 2.0x	26.6	7.4	591
MoibleNet v2 1.4x	**25.8**	6.1	582

Model (CIFAR10)	Error (%)	Params (M)
NASNet-A	3.41	3.3
PNASNet-5	3.41	3.2
AmoebaNet-A	**3.34**	3.2
Wide-DenseNet C3	3.81	3.4
Wide-DenseNet C2sp	3.54	3.2
NASNet-A + *cutout*	2.65	3.3
Wide-DenseNet C2sp + *cutout* + *mixup*	**2.44**	**3.2**

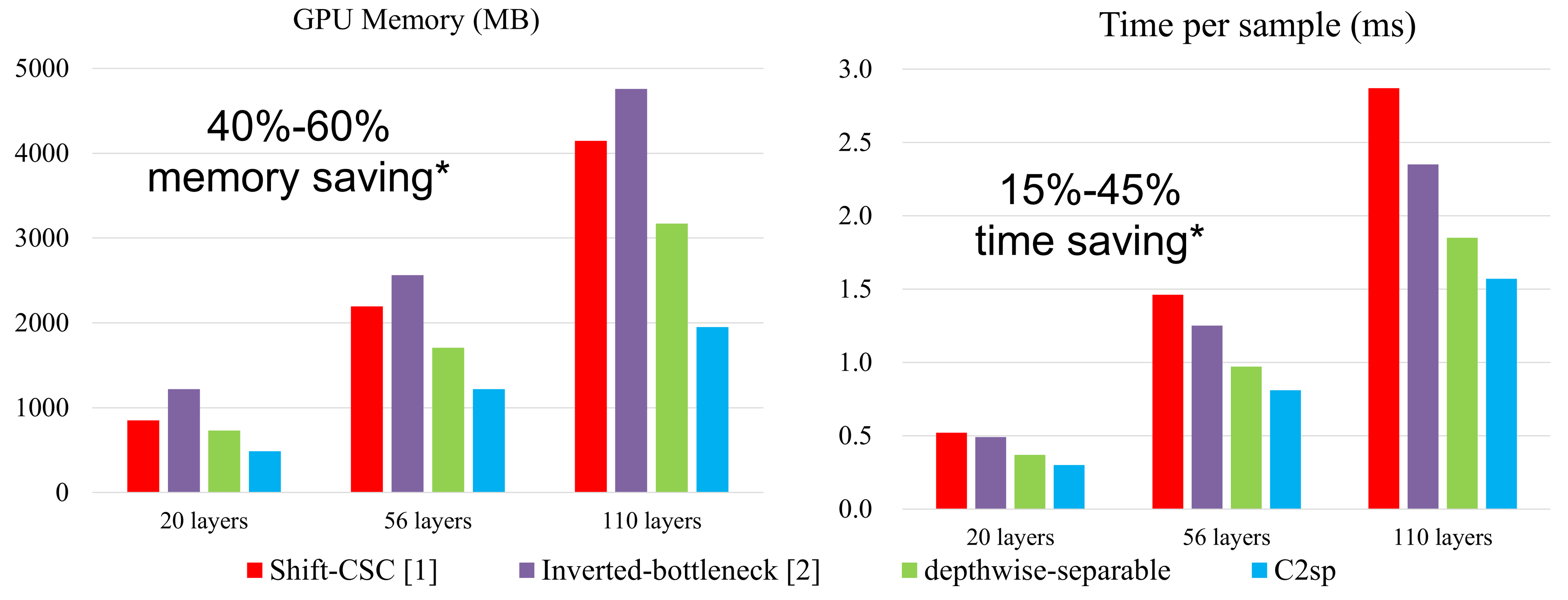


- Image generation tasks (GAN):

Model	CIFAR10	LSUN	CelebA
C5	**26.54±1.38**	30.84±1.13	33.90±3.77
C3	24.12±0.47	36.04±7.10	43.39±5.78
C4	26.45±1.50	39.17±5.52	37.93±7.54
C4sp	24.86±0.41	**24.61±2.45**	**30.22±3.02**
C2		Non-convergence	
C2sp	23.35±0.26	27.73±6.76	31.25±4.86

Reduced Overheads

- C2sp vs other compact CNN building blocks



*Naïve Tensorflow high-level implementation, can be further optimized

[1] Wu, B., et al. Shift: A zero flop, zero parameter alternative to spatial convolutions. *ICCV*'18.
[2] Sandler, M., et al. Mobilenetv2: Inverted residuals and linear bottlenecks. *CVPR*'18.
[3] He, K., Sun, J. Convolutional neural networks at constrained time cost. *CVPR*'15.