Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks

Jie Hu **Li Shen** **Samuel Albanie** **Gang Sun** **Andrea Vedaldi**

Presenters:

Siba Smarak Panigrahi and Sohan Patnaik

Reading Session IV
Kharagpur Data Analytics Group
IIT Kharagpur

March 28, 2021

Important Notes

- We expect you all to have certain queries regarding the presentation, certain intricate doubts maybe... **Please put them in the chat of this meeting**. We will address them all at the end!
- To ensure smooth flow of the presentation, we need you all to **keep your microphones and video turned off**.
- Finally, a **Google Form** [Link] will be released for feedback but most importantly, we ask you to put up name of any AI-related paper to be presented in the upcoming sessions!

NOTE: DWConv and Class Selectivity Index can be accessed from the link available on README of KDAG Reading session repo!

Outline

- Problem CNNs face
- Let’s think what we can do
- Gather & Excite operators
- Different GE-pairs
- Results & Discussion
- Comparing ResNet-50 and ResNet-50 with GE-θ
  - Class selectivity index
  - Optimization Curves
  - Feature Importance
- Increasing Extent Ratio
- Conclusion

Problem CNNs face

To improve visual representation, focus on feature context in deep networks

- receptive fields of feature extractors is supposed to be large but effective size is smaller in practice
- augment functions that perform local decisions with functions that operate on a larger context
- bottom-up local operators prevent CNN from capturing contextual long-range feature interactions
- deeper layers
  - achieve greater abstraction
  - reduce resolution, increase receptive field size & number of feature channels

Let's think what we can do..

- Squeeze-and-Excitation network:
  - reweighting feature channels as a function of features from the full extent of input
  - squeeze operator - a lightweight context aggregator
  - resulting embeddings - passed to reweighting function to exploit information beyond the local receptive fields of each filter

- Use gather and excite operator:
  - allow CNN exploit contextual information of feature responses

Gather & Excite operators

A. **Gather operator:** $\zeta_G : \mathbb{R}^{\mathrm{HxWxC}} \rightarrow \mathbb{R}^{\mathrm{H'xW'xC}}$
   a. Operator on the space of feature maps

B. **Excite operator:** $\zeta_E = x \odot f(\hat{x})$

$$f : \mathbb{R}^{\mathrm{H'xW'xC}} \rightarrow [0,1]^{\mathrm{HxWxC}}$$

   a. Operator on the both the input feature space and outputs of gather operator

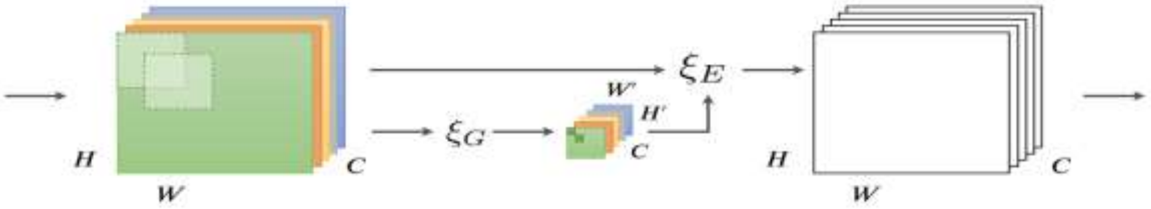
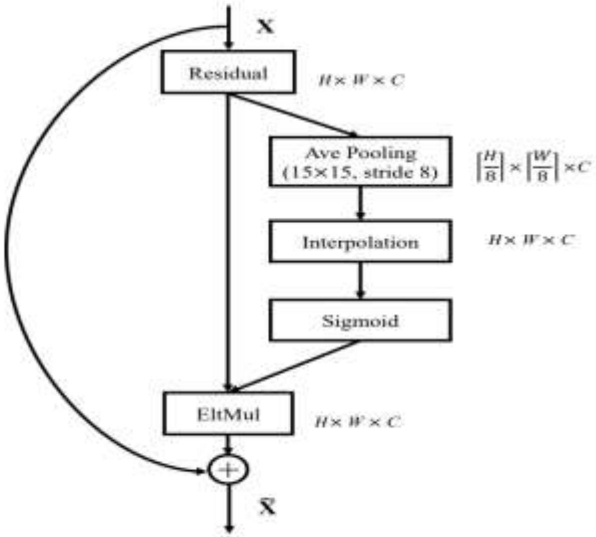


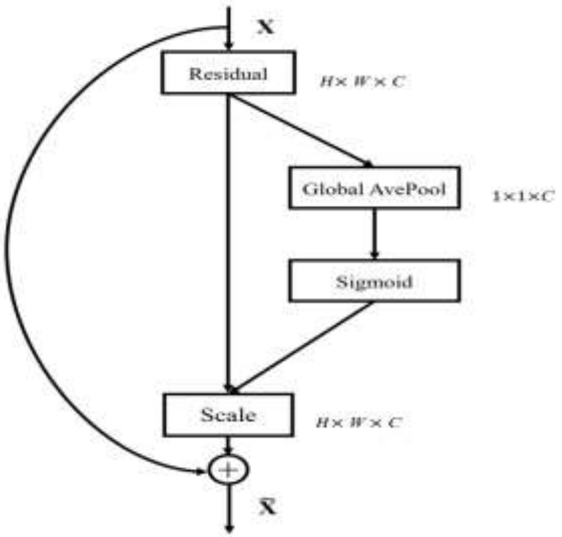
Figure 1: The interaction of a *gather-excite* operator pair, (ξ_G, ξ_E). The gather operator ξ_G first aggregates feature responses across spatial neighbourhoods. The resulting aggregates are then passed, together with the original input tensor, to an excite operator ξ_E that produces an output that matches the dimensions of the input.

Different GE-pairs

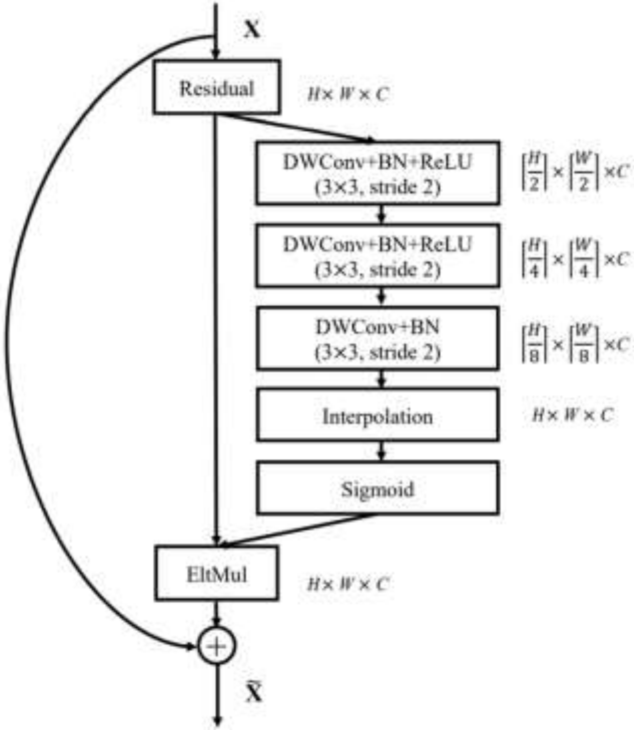
A GE-pair is a combination of gather operator and excite operator!

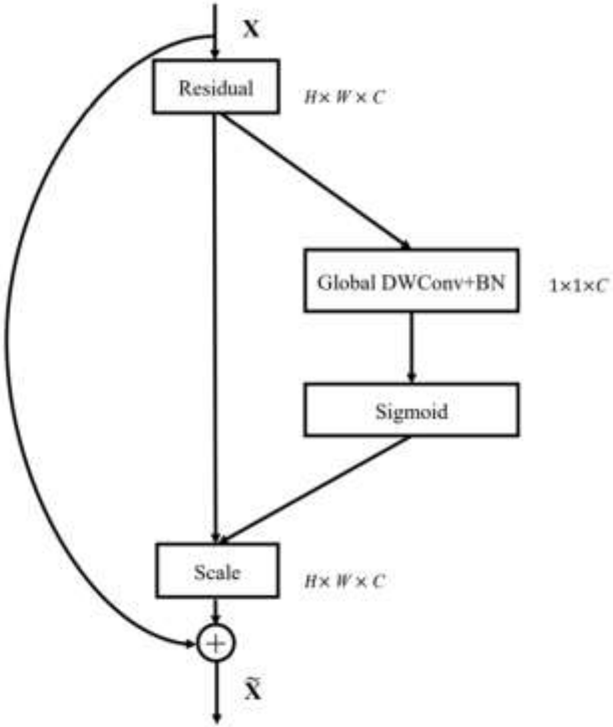
- GE-θ^-
  - Channel output after GE-pair: $y^c = x \odot \sigma(interp(\zeta_G(x)^c))$
  - Parameter-free
  - $\xi_G(x)$ is simple max-pooling OR average-pooling





- GE-θ
  - Parameterised gather operator: strided depthwise convolution

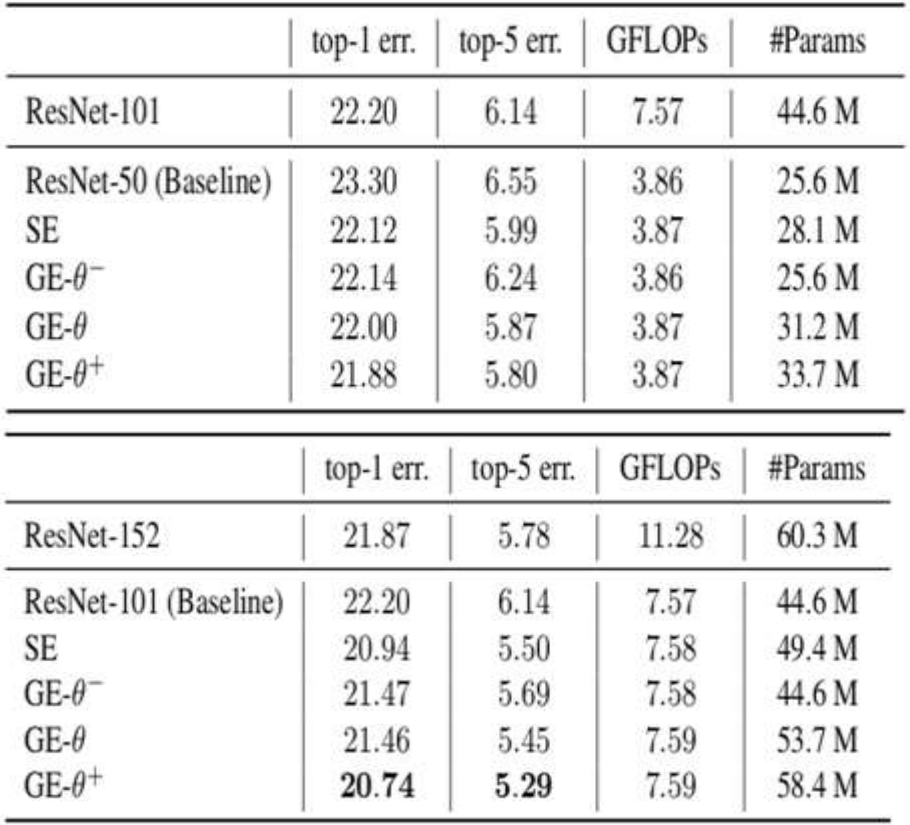

X
Residual
H× W × C
DWConv+BN+ReLU (3×3, stride 2)
$\lceil \frac{H}{2} \rceil \times \lceil \frac{W}{2} \rceil \times C$
DWConv+BN+ReLU (3×3, stride 2)
$\lceil \frac{H}{4} \rceil \times \lceil \frac{W}{4} \rceil \times C$
DWConv+BN (3×3, stride 2)
$\lceil \frac{H}{8} \rceil \times \lceil \frac{W}{8} \rceil \times C$
Interpolation
H× W × C
Sigmoid
EltMul
H× W × C
$\tilde{X}$



- GE-θ^{+}
  - Parameterised gather and excite operator
  - Add 1 x 1 convolution operation in excite operator to GE-θ framework
  - Make the excite operator trainable with parameters θ
    - Similar to Squeeze-and-excitation architecture
    - Channel output after GE-pair:

$$y^c = \zeta_E(x, \hat{x}) = x \odot \sigma(interp(f(\hat{x}|\theta)))$$

Results & Discussion



	top-1 err.	top-5 err.	GFLOPs	#Params
ResNet-101	22.20	6.14	7.57	44.6 M
ResNet-50 (Baseline)	23.30	6.55	3.86	25.6 M
SE	22.12	5.99	3.87	28.1 M
GE-θ^-	22.14	6.24	3.86	25.6 M
GE-θ	22.00	5.87	3.87	31.2 M
GE-θ^+	21.88	5.80	3.87	33.7 M

	top-1 err.	top-5 err.	GFLOPs	#Params
ResNet-152	21.87	5.78	11.28	60.3 M
ResNet-101 (Baseline)	22.20	6.14	7.57	44.6 M
SE	20.94	5.50	7.58	49.4 M
GE-θ^-	21.47	5.69	7.58	44.6 M
GE-θ	21.46	5.45	7.59	53.7 M
GE-θ^+	**20.74**	**5.29**	7.59	58.4 M

Comparing various GE configurations with ResNet-50 and ResNet-101 as baseline on the ImageNet validation set

[1.2 million training + 50k validation]

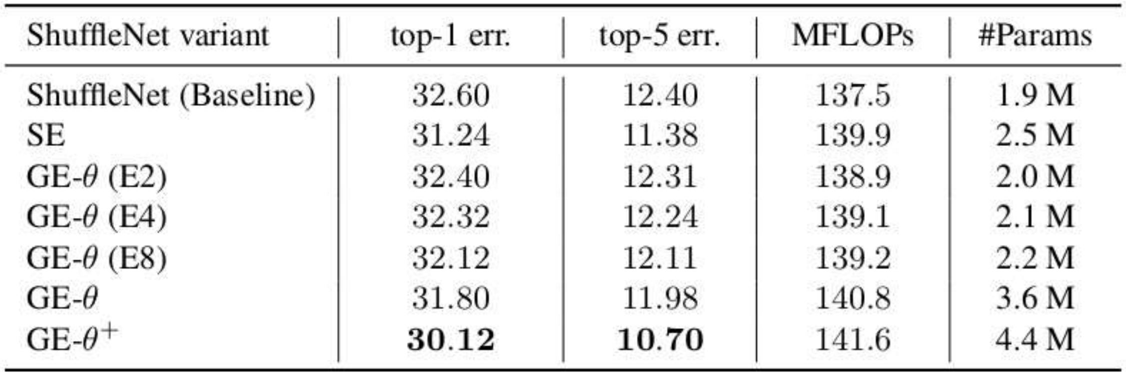
[224*224 pixel images]

- **Addition of GE-pair into baseline-architectures**
  - can improve the performance, even better than deeper architectures!
  - And that too, at a comparatively lower computational cost

But, GE operators themselves add layers …

- Extremely lightweight !!

Results & Discussion

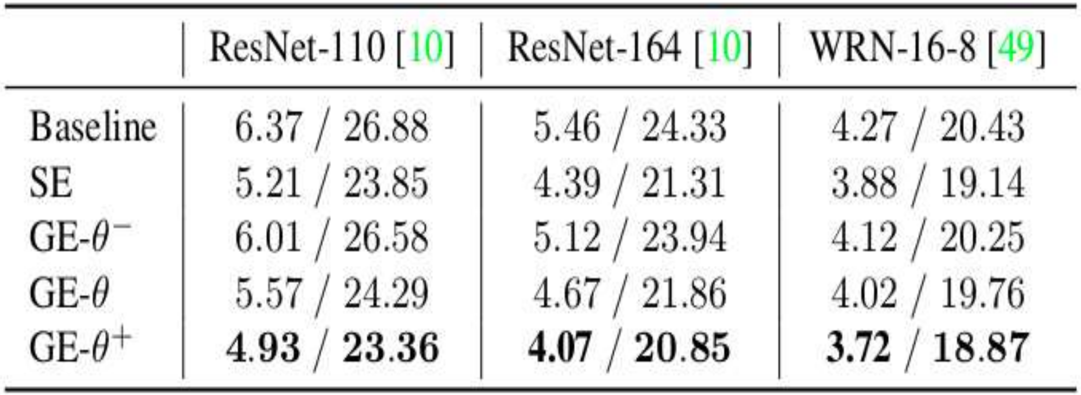


ShuffleNet variant	top-1 err.	top-5 err.	MFLOPs	#Params
ShuffleNet (Baseline)	32.60	12.40	137.5	1.9 M
SE	31.24	11.38	139.9	2.5 M
GE-θ (E2)	32.40	12.31	138.9	2.0 M
GE-θ (E4)	32.32	12.24	139.1	2.1 M
GE-θ (E8)	32.12	12.11	139.2	2.2 M
GE-θ	31.80	11.98	140.8	3.6 M
GE-θ^+	**30.12**	**10.70**	141.6	4.4 M

Comparing various GE configurations with ShuffleNet baseline on ImageNet validation dataset

- Addition of GE-pair to ShuffleNet leads to longer training time
- more number of parameters to produce same baseline
- longer epochs to optimise and reproduce baseline performances
- but the performance improves!

Results & Discussion



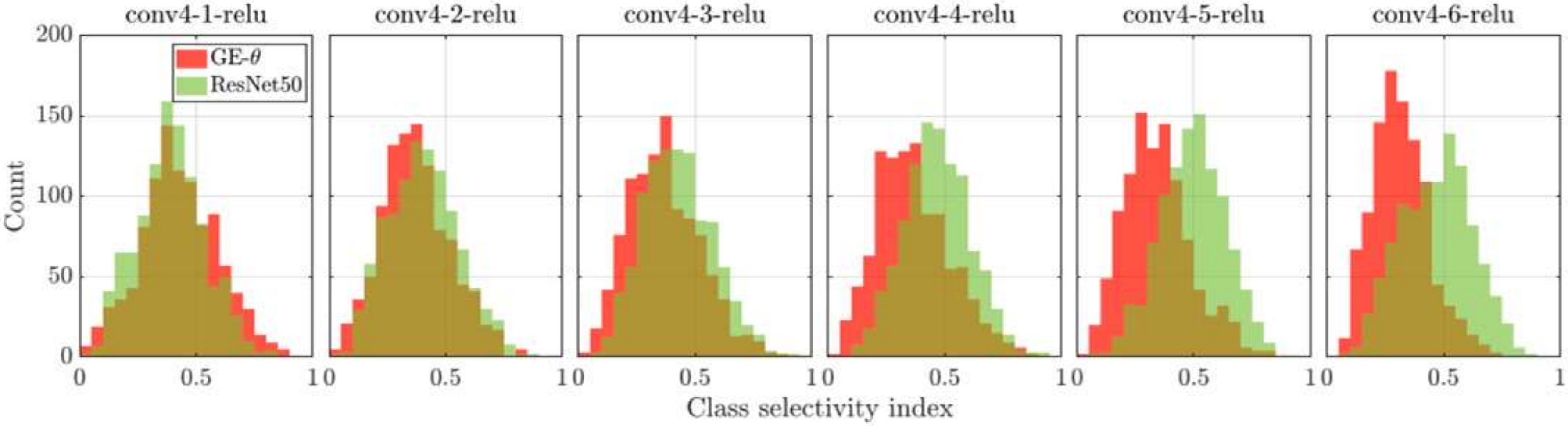
	ResNet-110 [10]	ResNet-164 [10]	WRN-16-8 [49]
Baseline	6.37 / 26.88	5.46 / 24.33	4.27 / 20.43
SE	5.21 / 23.85	4.39 / 21.31	3.88 / 19.14
GE-θ^-	6.01 / 26.58	5.12 / 23.94	4.12 / 20.25
GE-θ	5.57 / 24.29	4.67 / 21.86	4.02 / 19.76
GE-θ^+	**4.93** / **23.36**	**4.07** / **20.85**	**3.72** / **18.87**

Comparing various GE configurations on CIFAR-10/100 with standard data augmentation

- GE-pair works well with different characteristic images than ImageNet!
  - CIFAR 10/100
  - 50k training + 10k testing
  - 32*32 pixel color images

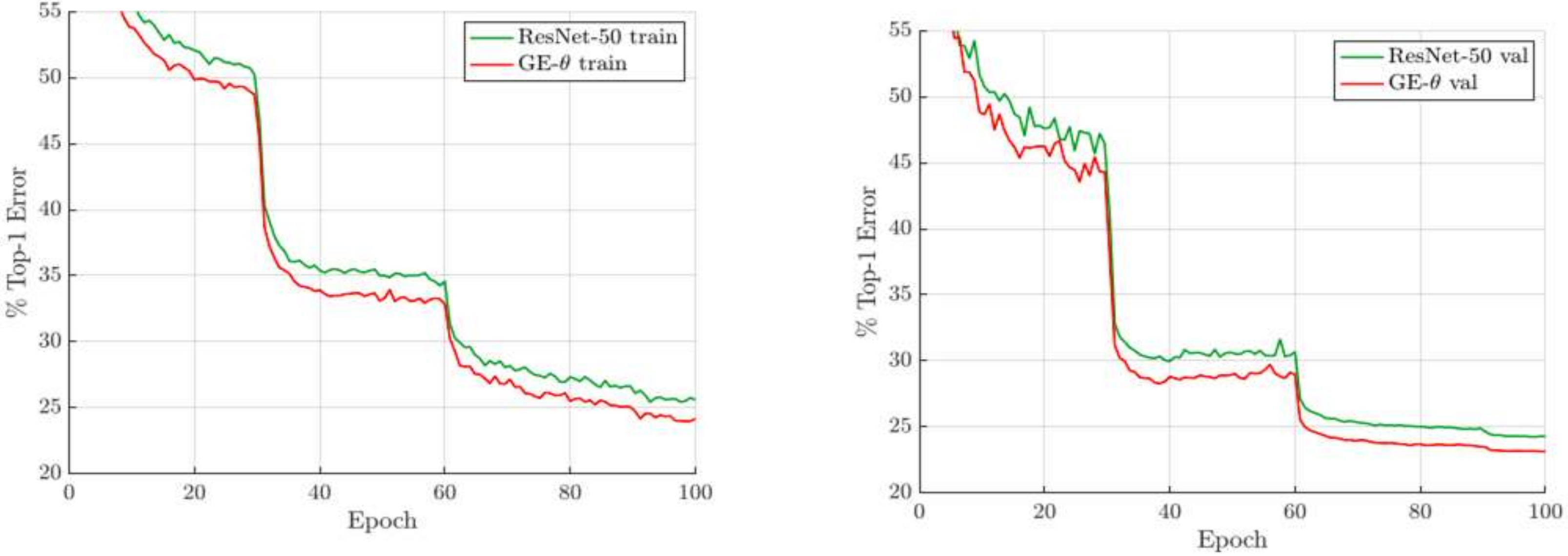
- GE-pair addition to ResNet-50 backbone object detector improves baseline mAP from 27.3% to 28.6% on MS-COCO dataset!

Comparing ResNet-50 and ResNet-50 with GE-θ (Class Selectivity Index)



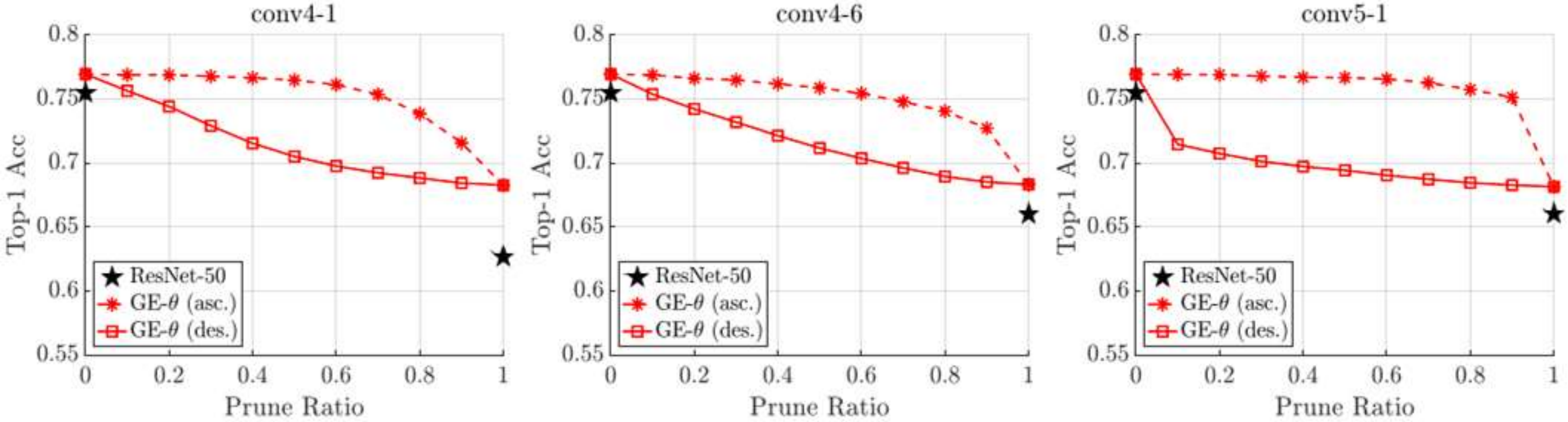
The class selectivity index decreases with increasing depth for ResNet-50 with GE-θ and in the deeper layer there is proper distinction between two models under consideration

Comparing ResNet-50 and ResNet-50 with GE-θ (Optimization Curves)



We can observe from above graphs, ResNet-50 with GE-θ model always has always lower error throughout the optimization process

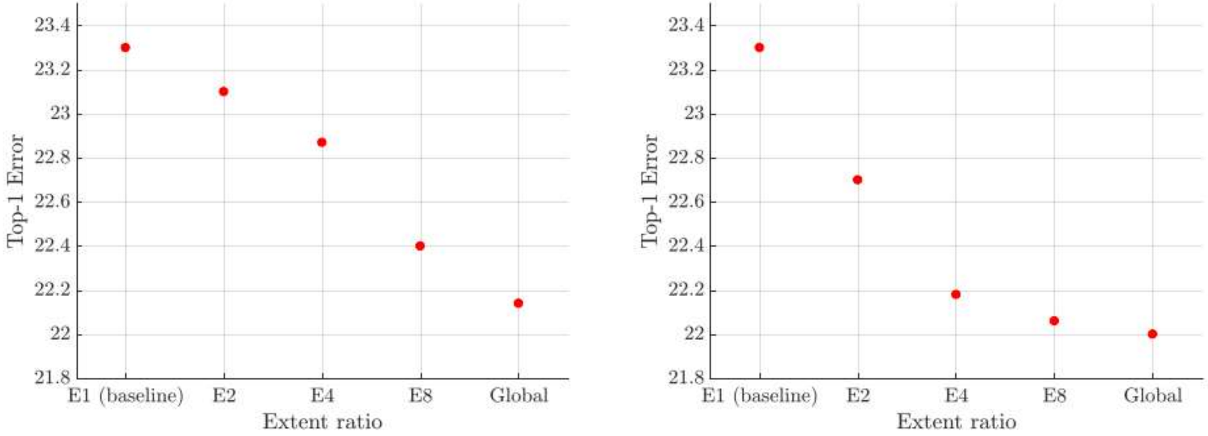
Comparing ResNet-50 and ResNet-50 with GE-θ (Feature Importance)



The above figure shows the effect of dropping off (on the basis of prune ratio) various feature maps at various indicated stages on the basis of ascending or descending order

Increasing the Extent Ratio

$$H' = \left\lceil \frac{H}{e} \right\rceil, W' = \left\lceil \frac{W}{e} \right\rceil$$



The above figure shows the effect of increasing extent ratio (e). Baseline is equivalent to e = 1. With increase in extent ratio, the performance improved with highest in case of global extent ratio.

Conclusion

- simple, lightweight approach
- use composition of two operators: gather & excite
- gather aggregates contextual information across large neighbourhoods of each feature map
- excite redistributes pooled information to local features
- operators are cheap
- few number of added parameters
- low computational complexity
- Integrate directly in existing architectures

Thank You!