ResNeSt: Split-Attention Networks

Hang Zhang, Chongruo Wu*, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola

Amazon, University of California, Davis*
{hzaws,chongrwu,zhongyue,yzaws,haibilin,zhiz,ysunmzn, htong,jonasmue,manmatha,mli,smola}@amazon.com

Presented by: Hengtao Guo
06/10/2020

Motivations

1. ResNet models are originally designed for classification tasks, and may not be suitable for other downstream tasks (object detection, segmentation etc)
2. Boosting the performance requires manually “surgery” to the ResNet structure, based on different downstream tasks:
   a) Pyramid Module
   b) Long-range connections
   c) Cross-channel feature map attention

Can we create a versatile backbone with universally improved feature representations, thereby improving performance across multiple tasks at the same time?

Motivations

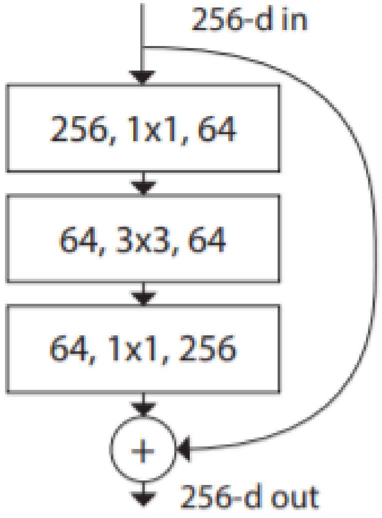
1. Cross-channel information has demonstrated success in downstream applications [56,64,65]
2. Recent image classification networks focused more on group or depth-wise convolution

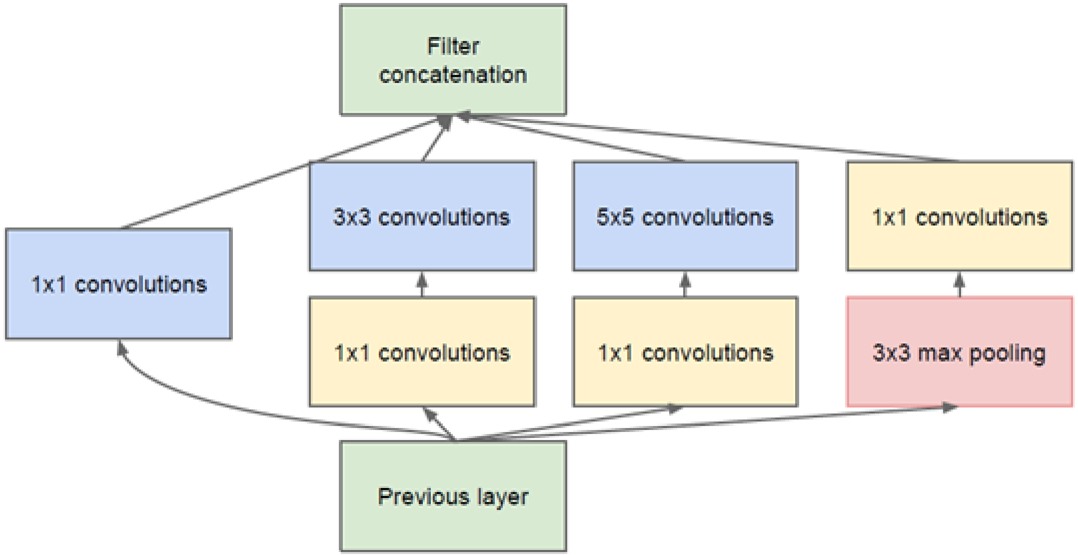
Introducing cross-channel information into the basic block of ResNet!
Benefits downstream tasks!

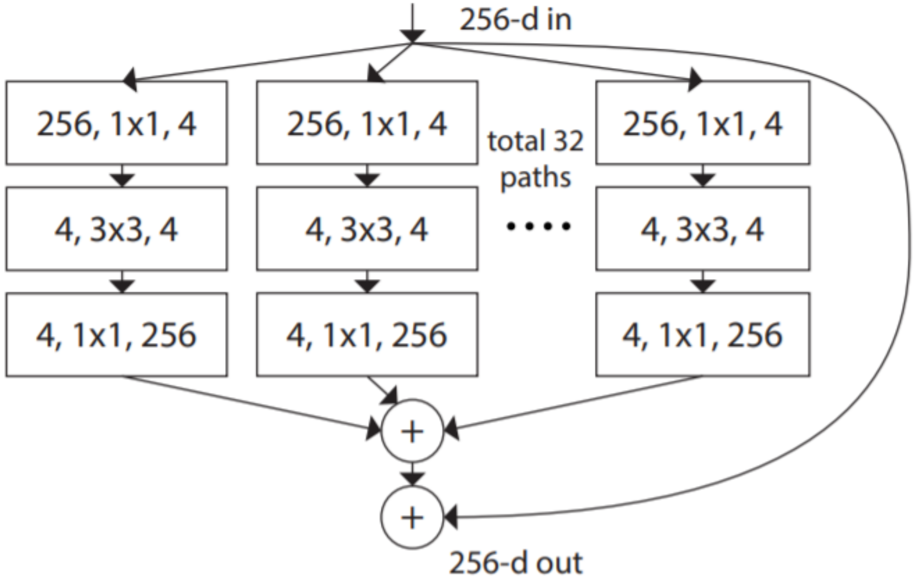
Contributions

1. By stacking several Split-Attention blocks, we create a ResNet-like network called *ResNeSt* (S stands for \split").
2. Large scale benchmarks on image classification and transfer learning applications.

Generally, this work is the combination of:
ResNeXt + Split Attention







ResNet

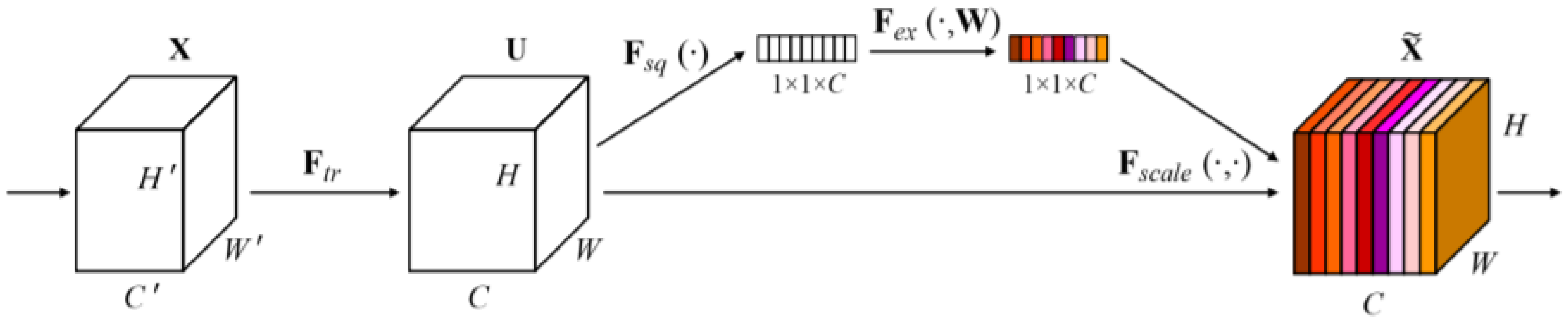
Manually design
Add width

Inception-Net
(GoogleNet)

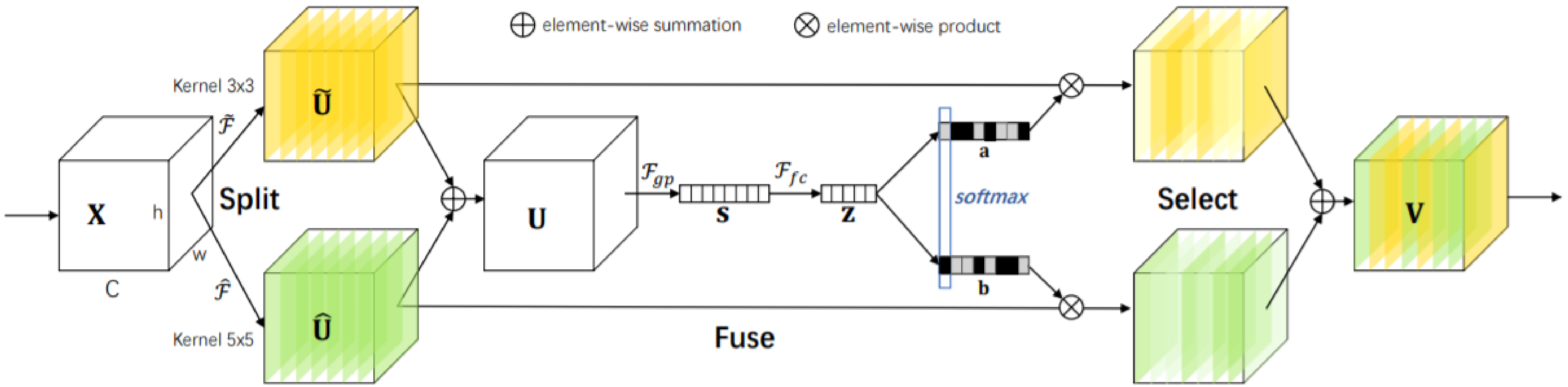
All pathways share the same design

ResNeXt

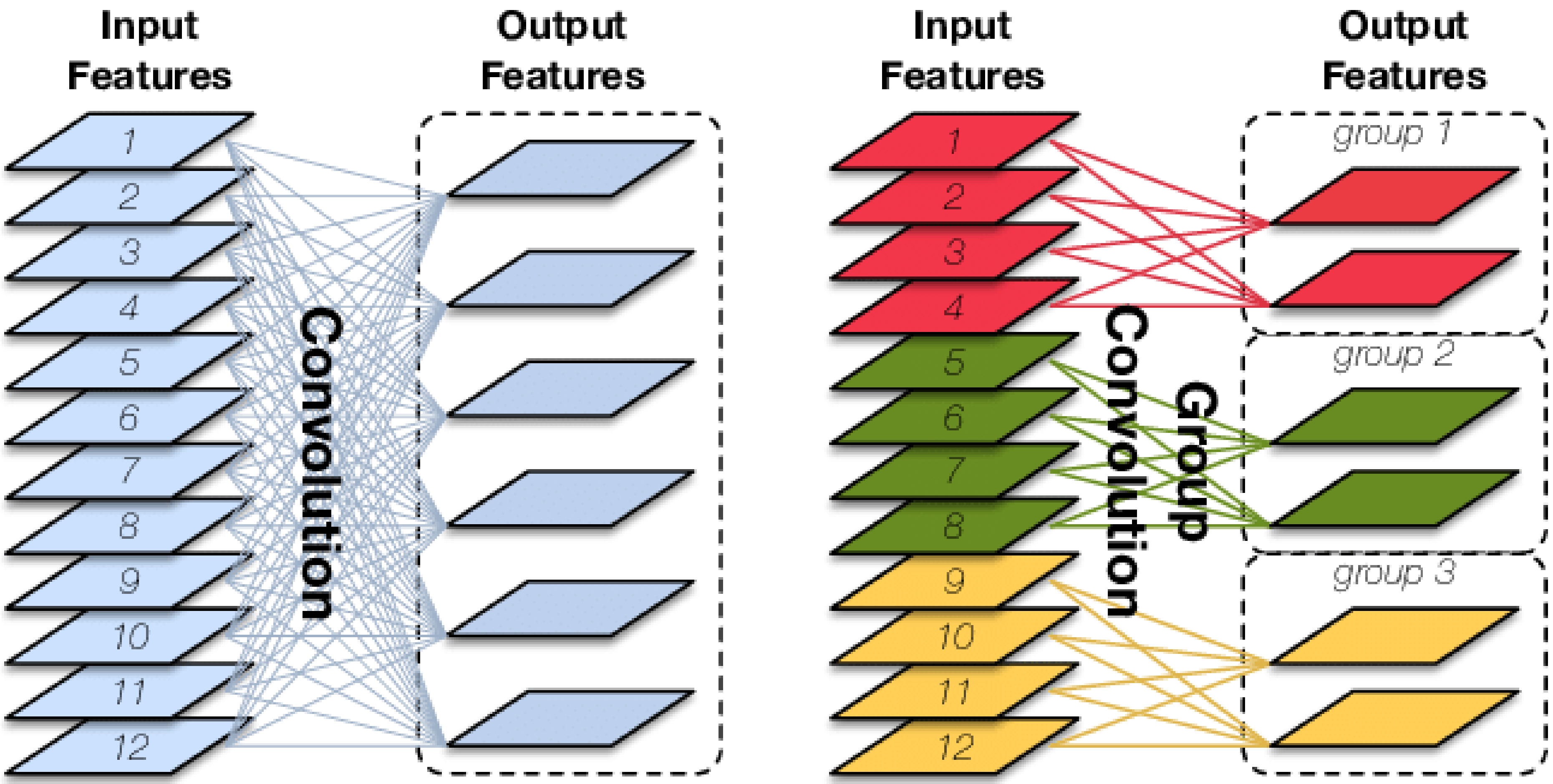
SE-Net (squeezing & excitation)



SK-Net (Selective Kernel Convolution)



Group Convolution



Split-Attention

1. Global Average Pooling

$$s_c^k = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \hat{U}_c^k(i,j).$$

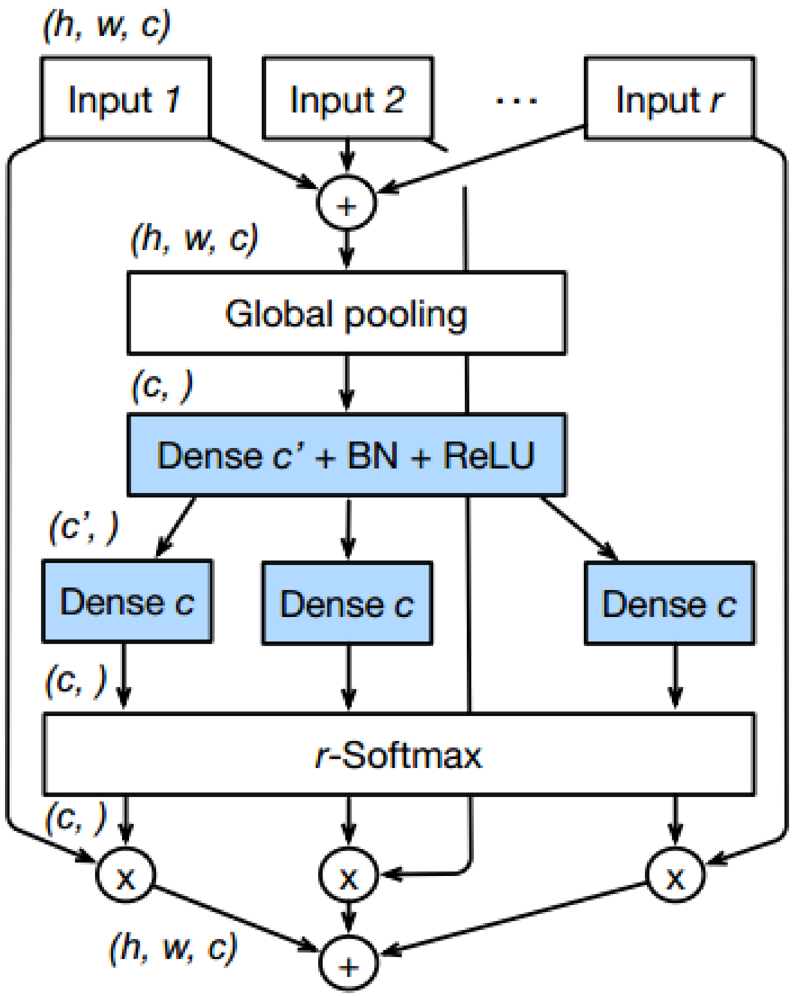
2. Softmax

$$a_i^k(c) = \begin{cases} \frac{exp(\mathcal{G}_i^c(s^k))}{\sum_{j=0}^{R} exp(\mathcal{G}_j^c(s^k))} & \text{if } R > 1, \\ \frac{1}{1+exp(-\mathcal{G}_i^c(s^k))} & \text{if } R = 1, \end{cases}$$

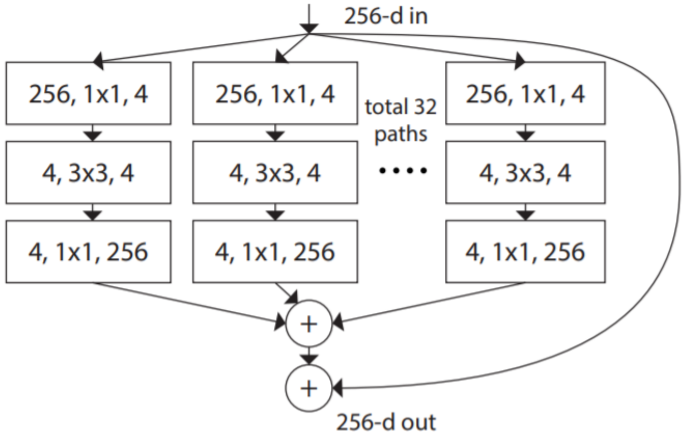
3. Feature-map channel attention

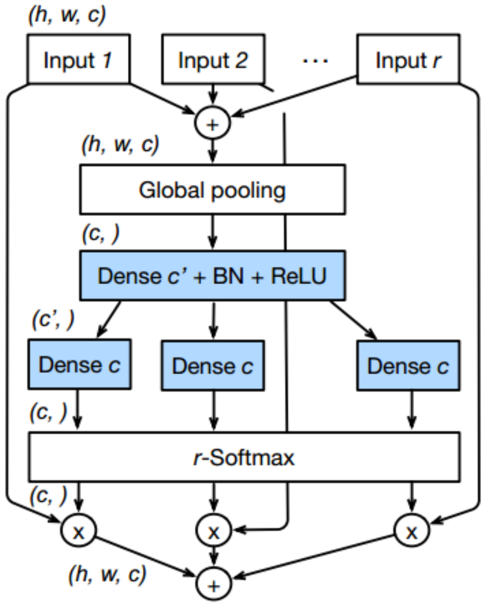
$$V_c^k = \sum_{i=1}^{R} a_i^k(c) U_{R(k-1)+i},$$

Fig. 2: Split-Attention within a cardinal group. For easy visualization in the figure, we use $c = C/K$ in this figure.

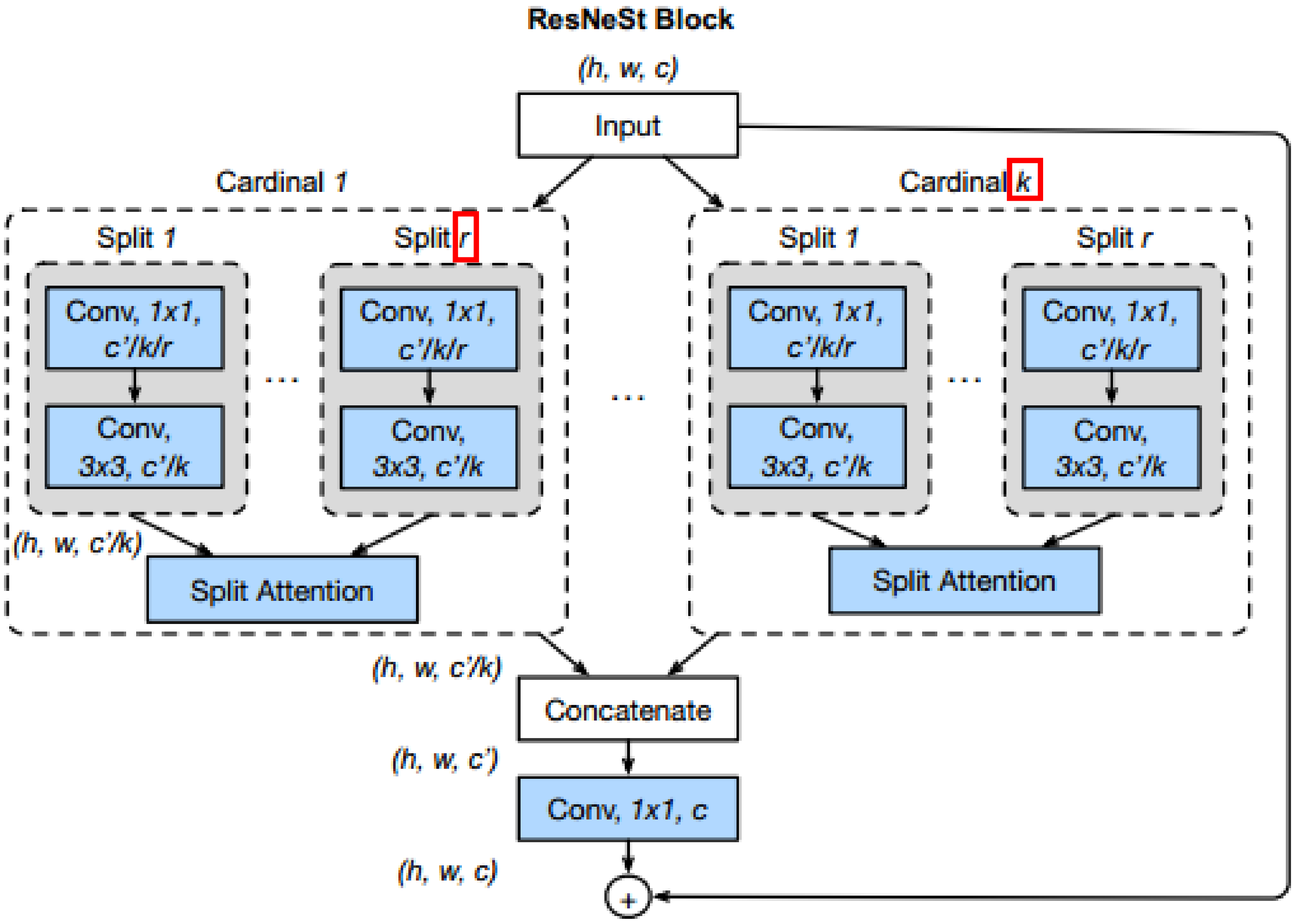


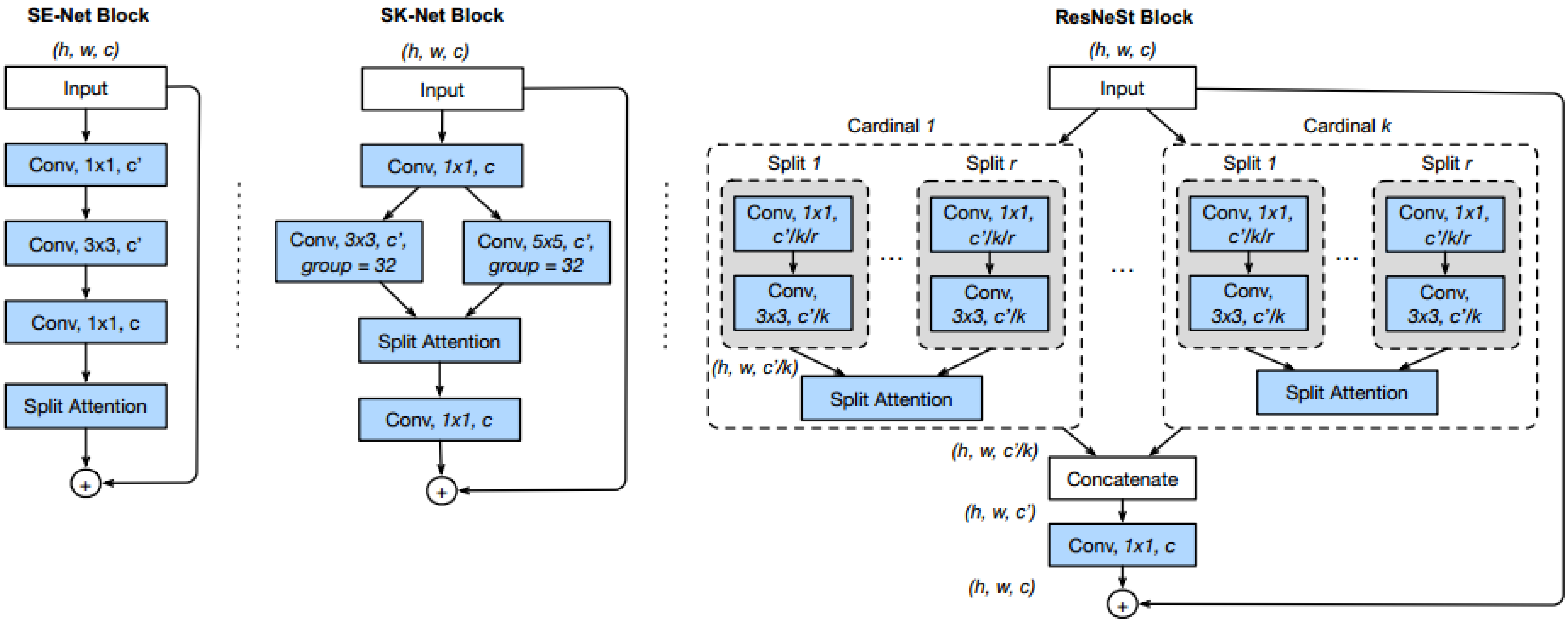
ResNeXt





Split Attention




SE-Net Block
(h, w, c)
Input
Conv, 1x1, c'
Conv, 3x3, c'
Conv, 1x1, c
Split Attention
SK-Net Block
(h, w, c)
Input
Conv, 1x1, c
Conv, 3x3, c', group = 32
Conv, 5x5, c', group = 32
Split Attention
Conv, 1x1, c
ResNeSt Block
(h, w, c)
Input
Cardinal 1
Cardinal k
Split 1
Split r
Conv, 1x1, c'/k/r
Conv, 3x3, c'/k
(h, w, c'/k)
Split Attention
(h, w, c'/k)
Concatenate
(h, w, c')
Conv, 1x1, c
(h, w, c)

Experiments

1. Classification tasks
2. Downstream tasks
   a) Object Detection
   b) Instance Segmentation
   c) Semantic Segmentation

ImageNet Classification

	#P	GFLOPs	top-1 acc (%)	
			224×	320×
ResNet-50 [23]	25.5M	4.14	76.15	76.86
ResNeXt-50 [60]	25.0M	4.24	77.77	78.95
SENet-50 [29]	27.7M	4.25	78.88	80.29
ResNetD-50 [26]	25.6M	4.34	79.15	79.70
SKNet-50 [38]	27.5M	4.47	79.21	80.68
ResNeSt-50-fast(ours)	27.5M	4.34	**80.64**	**81.43**
ResNeSt-50(ours)	27.5M	5.39	81.13	81.82
ResNet-101 [23]	44.5M	7.87	77.37	78.17
ResNeXt-101 [60]	44.3M	7.99	78.89	80.14
SENet-101 [29]	49.2M	8.00	79.42	81.39
ResNetD-101 [26]	44.6M	8.06	80.54	81.26
SKNet-101 [38]	48.9M	8.46	79.81	81.60
ResNeSt-101-fast(ours)	48.2M	8.07	**81.97**	**82.76**
ResNeSt-101(ours)	48.3M	10.2	82.27	83.00

Table 3: Image classification results on ImageNet, comparing our proposed ResNeSt with other ResNet variants of similar complexity in 50-layer and 101-layer configurations. We report top-1 accuracy using crop sizes 224 and 320.

GFLOPS:
Giga **Fl**oating-point **O**perations **P**er **S**econd

Top-1 accuracy is the conventional accuracy, which means that the model answer (the one with the highest probability) must be exactly the expected answer.

Top-5 accuracy means that *any* of your model that gives 5 highest probability answers that must match the expected answer.

	Method	Backbone	mAP%
Prior Work	Faster-RCNN [46]	ResNet101 [22]	37.3
		ResNeXt101 [5, 60]	40.1
		SE-ResNet101 [29]	41.9
	Faster-RCNN+DCN [12]	ResNet101 [5]	42.1
	Cascade-RCNN [2]	ResNet101	42.8
Our Results	Faster-RCNN [46]	ResNet50 [57]	39.25
		ResNet101 [57]	41.37
		ResNeSt50 (ours)	42.33
		ResNeSt101 (ours)	**44.72**
	Cascade-RCNN [2]	ResNet50 [57]	42.52
		ResNet101 [57]	44.03
		ResNeSt50 (ours)	45.41
		ResNeSt101 (ours)	**47.50**
	Cascade-RCNN [2]	ResNeSt200 (ours)	49.03

Table 5: Object detection results on the MS-COCO validation set. Both Faster-RCNN and Cascade-RCNN are significantly improved by our ResNeSt backbone.

Object Detection

Compared to the baselines using standard ResNet, Our backbone is able to boost mean average precision by around 3% on both Faster-RCNNs and Cascade-RCNNs. The result demonstrates our backbone has good generalization ability and can be easily transferred to the downstream task. Notably, our ResNeSt50 outperforms ResNet101 on both Faster-RCNN and Cascade-RCNN detection models, using significantly fewer parameters. Detailed results in Table 10. We evaluate our Cascade-RCNN with ResNeSt101 deformable, that is trained using 1x learning rate schedule on COCO test-dev set as well. It yields a box mAP of 49.2 using single scale inference.

	Method	Backbone	box mAP%	mask mAP%
Prior Work	DCV-V2 [72]	ResNet50	42.7	37.0
	HTC [4]	ResNet50	43.2	38.0
	Mask-RCNN [22]	ResNet101 [5]	39.9	36.1
	Cascade-RCNN [3]	ResNet101	44.8	38.0
Our Results	Mask-RCNN [22]	ResNet50 [57]	39.97	36.05
		ResNet101 [57]	41.78	37.51
		ResNeSt50 (ours)	42.81	38.14
		ResNeSt101 (ours)	**45.75**	**40.65**
	Cascade-RCNN [2]	ResNet50 [57]	43.06	37.19
		ResNet101 [57]	44.79	38.52
		ResNeSt50 (ours)	46.19	39.55
		ResNeSt101 (ours)	**48.30**	**41.56**

Table 6: Instance Segmentation results on the MS-COCO validation set. Both Mask-RCNN and Cascade-RCNN models are improved by our ResNeSt backbone. Models with our ResNeSt-101 outperform all prior work using ResNet-101.

	Method	Backbone	pixAcc%	mIoU%
Prior Work	UperNet [59]	ResNet101	81.01	42.66
	PSPNet [69]	ResNet101	81.39	43.29
	EncNet [65]	ResNet101	81.69	44.65
	CFNet [66]	ResNet101	81.57	44.89
	OCNet [63]	ResNet101	-	45.45
	ACNet [17]	ResNet101	81.96	45.90
Ours	DeeplabV3 [7]	ResNet50 [21]	80.39	42.1
		ResNet101 [21]	81.11	44.14
		ResNeSt-50 (ours)	81.17	45.12
		ResNeSt-101 (ours)	**82.07**	**46.91**

	Method	Backbone	mIoU%
Prior Work	DANet [16]	ResNet101	77.6
	PSANet [70]	ResNet101	77.9
	PSPNet [69]	ResNet101	78.4
	AAF [33]	ResNet101	79.2
	DeeplabV3 [7]	ResNet101	79.3
	OCNet [63]	ResNet101	80.1
Ours	DeeplabV3 [7]	ResNet50 [21]	78.72
		ResNet101 [21]	79.42
		ResNeSt-50 (ours)	79.87
		ResNeSt-101 (ours)	**80.42**

Table 7: Semantic segmentation results on validation set of: ADE20K (Left), Citscapes (Right). Models are trained without coarse labels or extra data.

Conclusion

1. This work proposed the ResNeSt architecture with a novel Split-Attention block that universally improves the learned feature representations to boost performance across image classification, object detection, instance segmentation and semantic segmentation.
2. In the latter downstream tasks, the empirical improvement produced by simply switching the backbone network to our ResNeSt is substantially better than task-specific modifications applied to a standard backbone such as ResNet.
3. Our Split-Attention block is easy to work with and computationally efficient, and thus should be broadly applicable across vision tasks.

56. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018)

64. Yuhui Yuan, J.W.: Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)

65. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)