ResNeSt: Split-Attention Network

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

Rishabh Tiwari

Novosibirsk State university

13 April 2021

Introduction

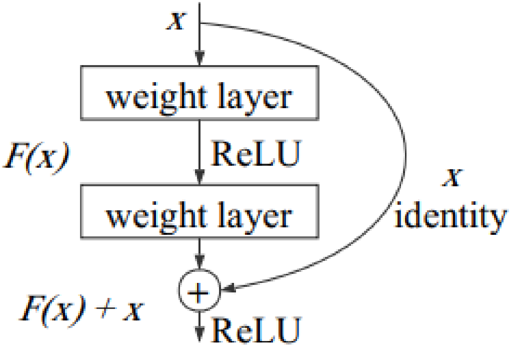
- The authors suggest a new ResNet-like network architecture that incorporates attention across groups of feature maps.
- If we Compare it to previous attention models such as SENet and SKNet, the new attention block applies the squeeze-and-attention operation separately to each of the selected groups, which is done in a computationally efficient way and implemented in a simple modular structure.

Motivation

- ResNet models were originally designed for image classification, they may not be suitable for various downstream applications because of the limited size of the receiving field and the lack of cross-channel interaction.
- To improve the performance of specific computer vision tasks requires network surgery to modify ResNet to make it more effective for specific tasks.
- Recent work in image classification is improved using neural archive structure search (NAS).
- Due to excessive memory consumption, some larger versions of these models cannot even be trained on the GPU with an appropriate batch size of 2 per device.
- This limits the use of nas-derived models in special tasks that involve intensive predictions such as segmentation.

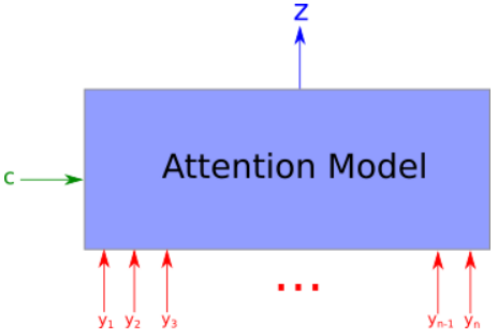
Residual And Attention Networks

- A residual network consists of residual units or blocks which have skip connections, also called identity connections.



Residual And Attention Networks

- focusing on specific parts of the input- has been applied in Deep Learning, for speech recognition, translation, reasoning, and visual identification of objects.



Architecture Of ResNest

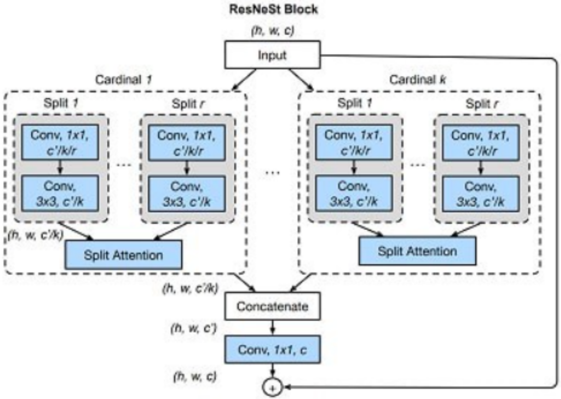


Image from the original paper

Architecture Of ResNest

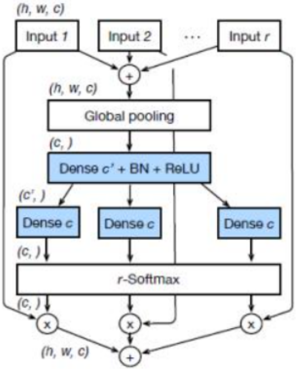
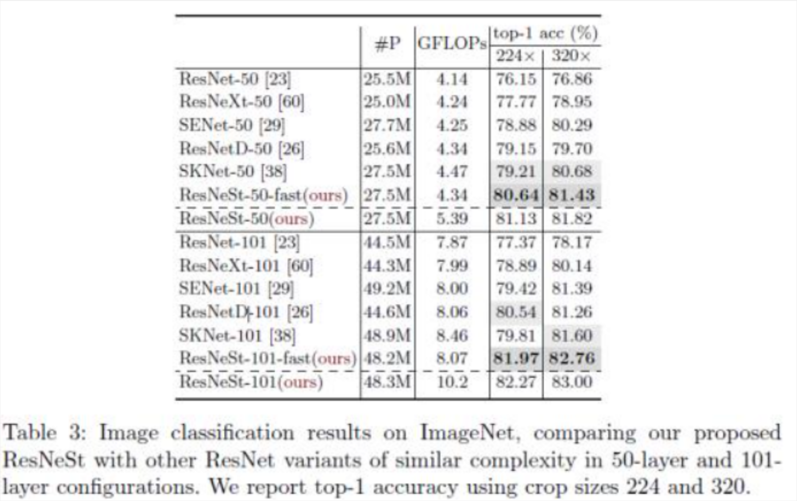


Fig. 2: Split-Attention within a cardinal group. For easy visualization in the figure, we use $c = C/K$ in this figure.

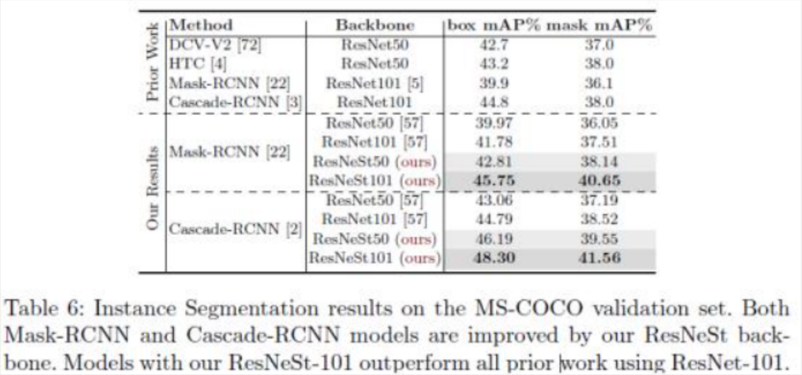
Results



	#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.

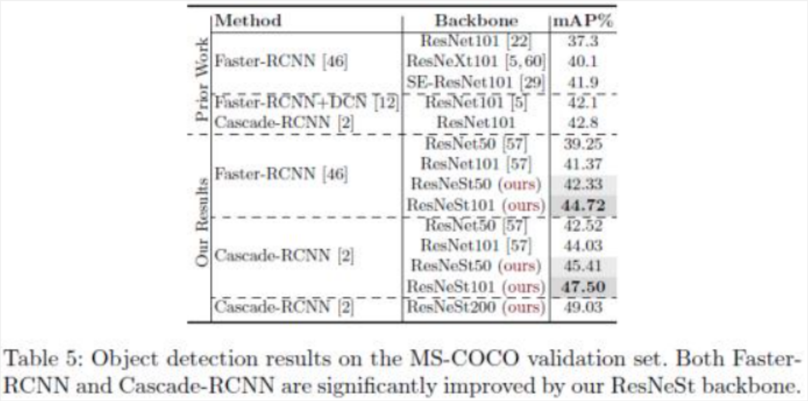
Results



	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.

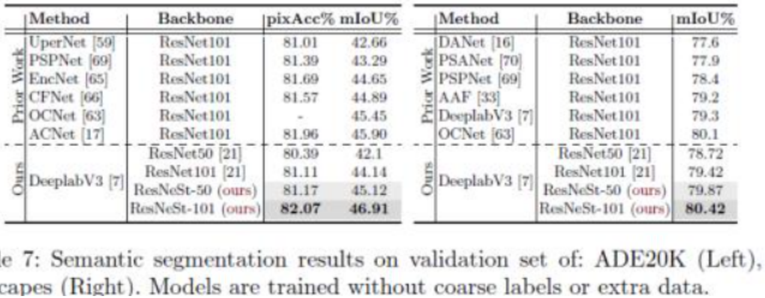
Results



	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.

Results



	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

- We explored the simple architecture modification of ResNet and put the functional diagram split attention into each network module.
- we created a ResNet-like network called ResNeSt (S stands for "split"). Our architecture only requires more calculations than existing ResNet variants and can easily be used as the basis for other visual tasks.
- The model using ResNeSt backbone can achieve optimal performance on multiple tasks, namely: image classification, object detection, instance segmentation and semantic segmentation. The proposed ResNeSt has better performance than all existing ResNet variants, the same computational efficiency, and even better speed accuracy trade-offs than the most advanced CNN model generated by neural structure search.

Thank You

THANK YOU