भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

ResNeSt-Split attention Network

(IEEE-CVPR 2020.)

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

Student Mentor: -
Sai Harsha Yelleni sir

Faculty Mentor: -
Prof. C.K. Mohan sir

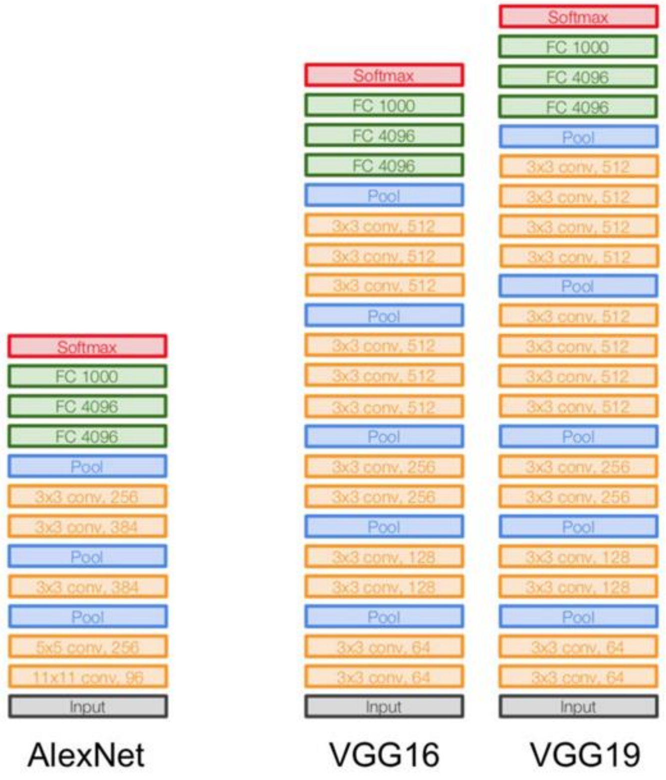
Presented By: -
Shubham Jain
(SM20MTECH12007)

Presentation Outline: -

- Motivation
- Challenges
- Previous Network Architectures
- ResNeSt: Split Attention Network
- Training Strategy
- Regularization
- Results
- Conclusion
- Future Work

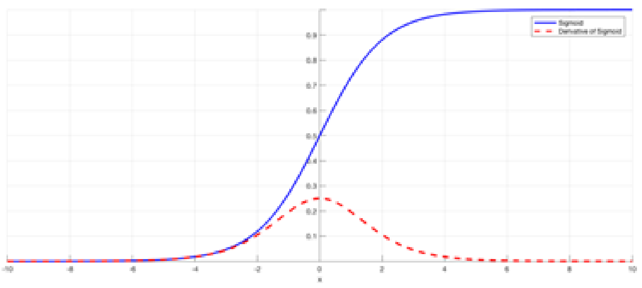
History / Motivation

- AlexNet (8 layers)
- VGGNet (11, 13, 16, 19 layers)
- GoogleNet (22 layers) also called inception (network in network) model.

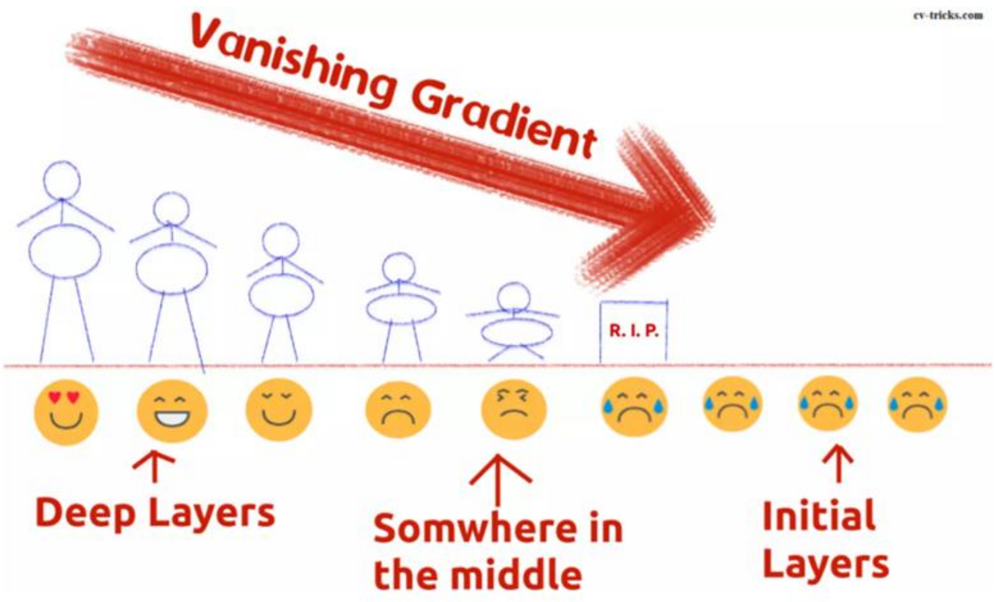


Challenges

Vanishing / exploding gradients.

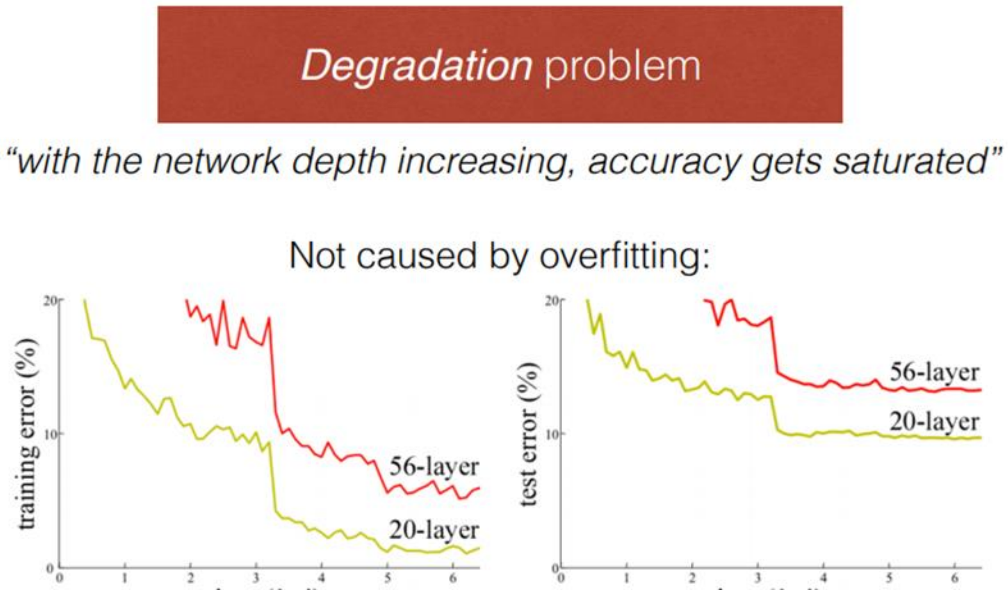


Sigmoid Function and its derivative



Challenges

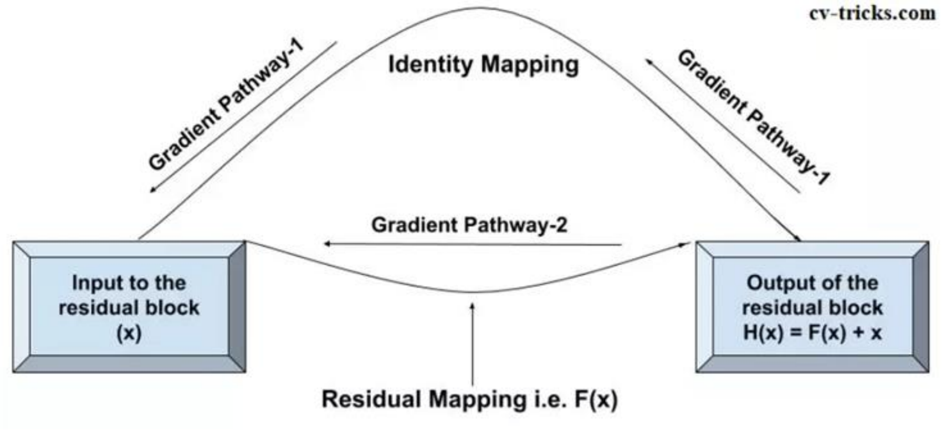
Degradation problem increasing the number of layers in the network abruptly degrades the accuracy.

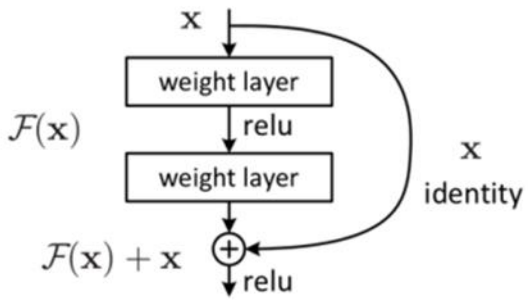


Previous Networks

- ResNet
- GoogleNet
- ResNext
- SE-Net (Squeeze and Excitation)
- SK-Net (Selective Kernel)

ResNet: Identity Connection



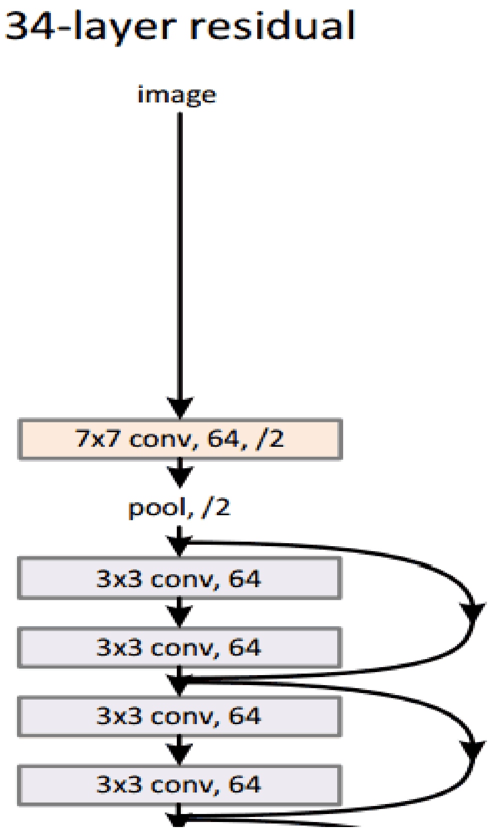


Note: -The identity connections introduce neither extra parameter nor computation complexity

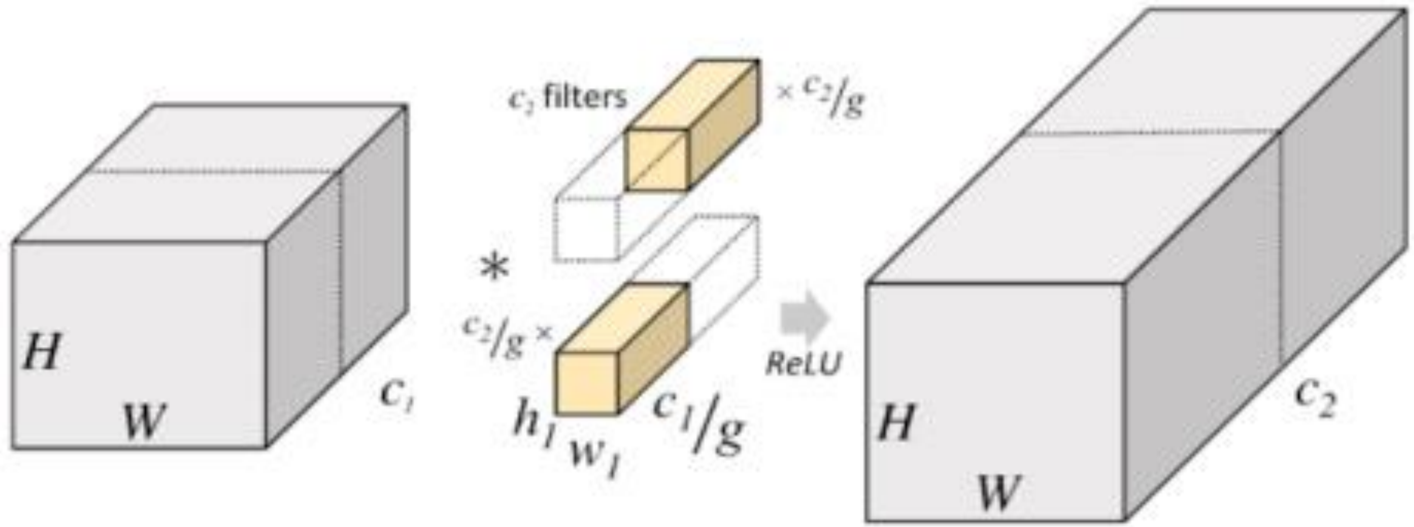
ResNet: Identity Connection

ResNet uses Batch Normalization. The problem of covariate shift is mitigated.

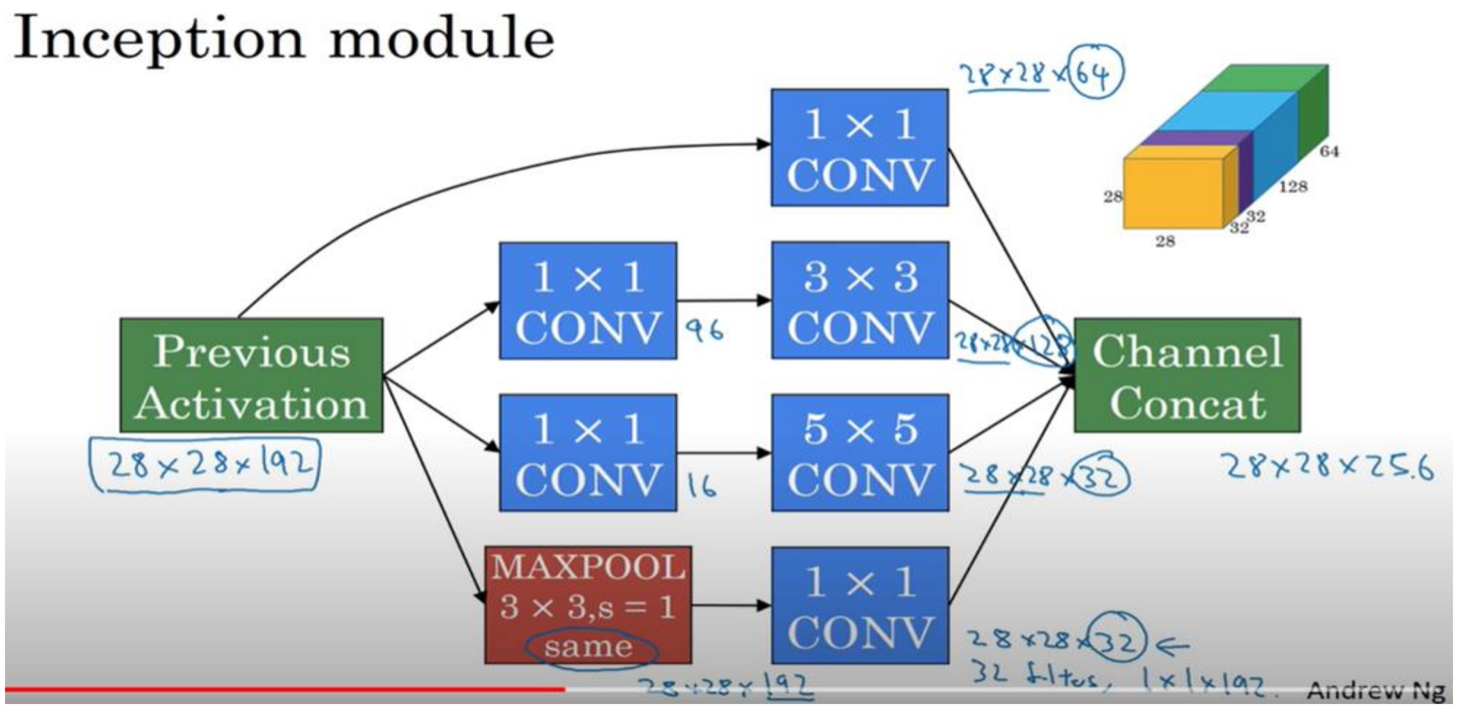
ResNet makes use of the Identity Connection, which helps to protect the network from vanishing gradient problem.



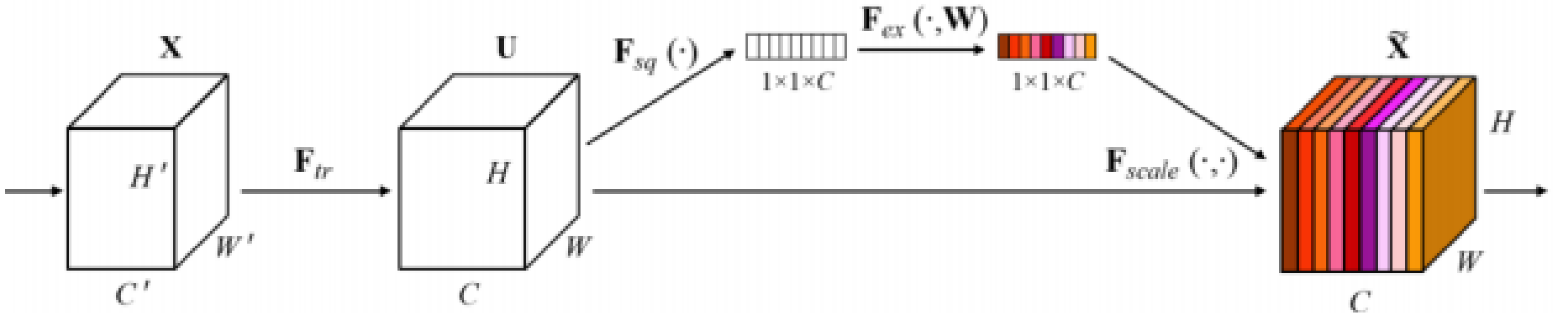
ResNext: Group Convolution



GoogleNet: Inception Network



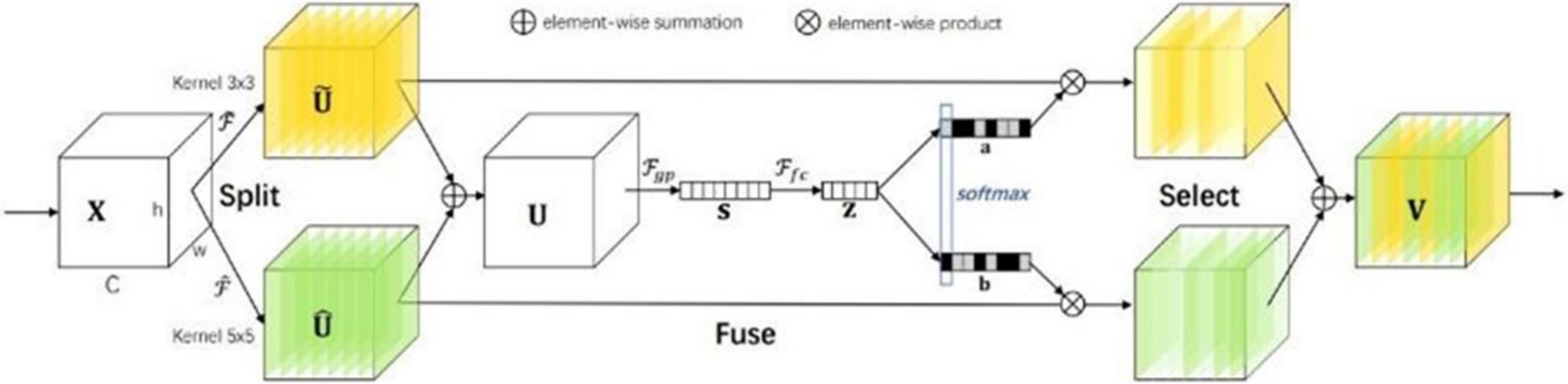
SE-Net: Squeeze and Excitation Network



The figure described detail three steps in Squeeze and Excitation network:

1) Transforming input to features map.
2) Squeezing: applying Global Average Pooling into feature maps in channel dimension
3) Excitation: applying two linear transformations on the result of Squeezing phase, and then gating it by a sigmoid function.

SK-Net: Selective Kernel Network



भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

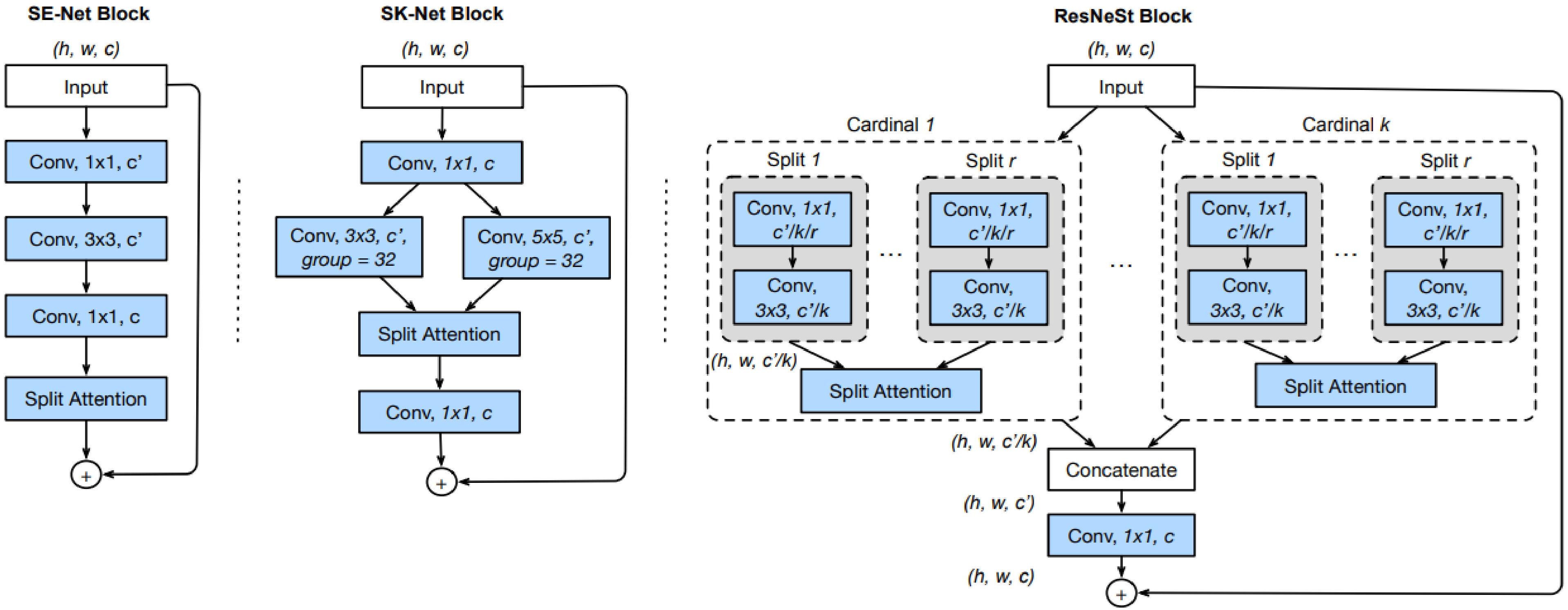
Overview

- "ResNet" introduced residual networks which showed significant improvement in accuracy by countering Vanishing Gradient problem.
- Multi-path representation has shown success in "GoogleNet"
- "ResNext" adopted group convolution in the ResNet bottle block, which converts the multi-path structure into a unified operation.
- "SE-Net" introduced a channel-attention mechanism.
- "SK-Net" brings the feature-map attention across two network branches.

ResNeSt

“Picking up all concepts that we have seen so far and using it in some way to improve performance”

ResNeSt: Split Attention Block



ResNeSt: Within Cardinal

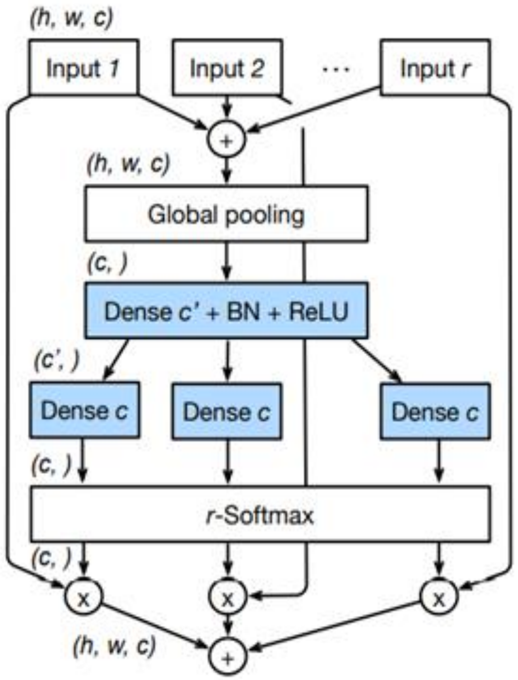
Let's choose c = C/K

$$\hat{U}^k = \sum_{j=R(k-1)+1}^{Rk} U_j$$

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

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

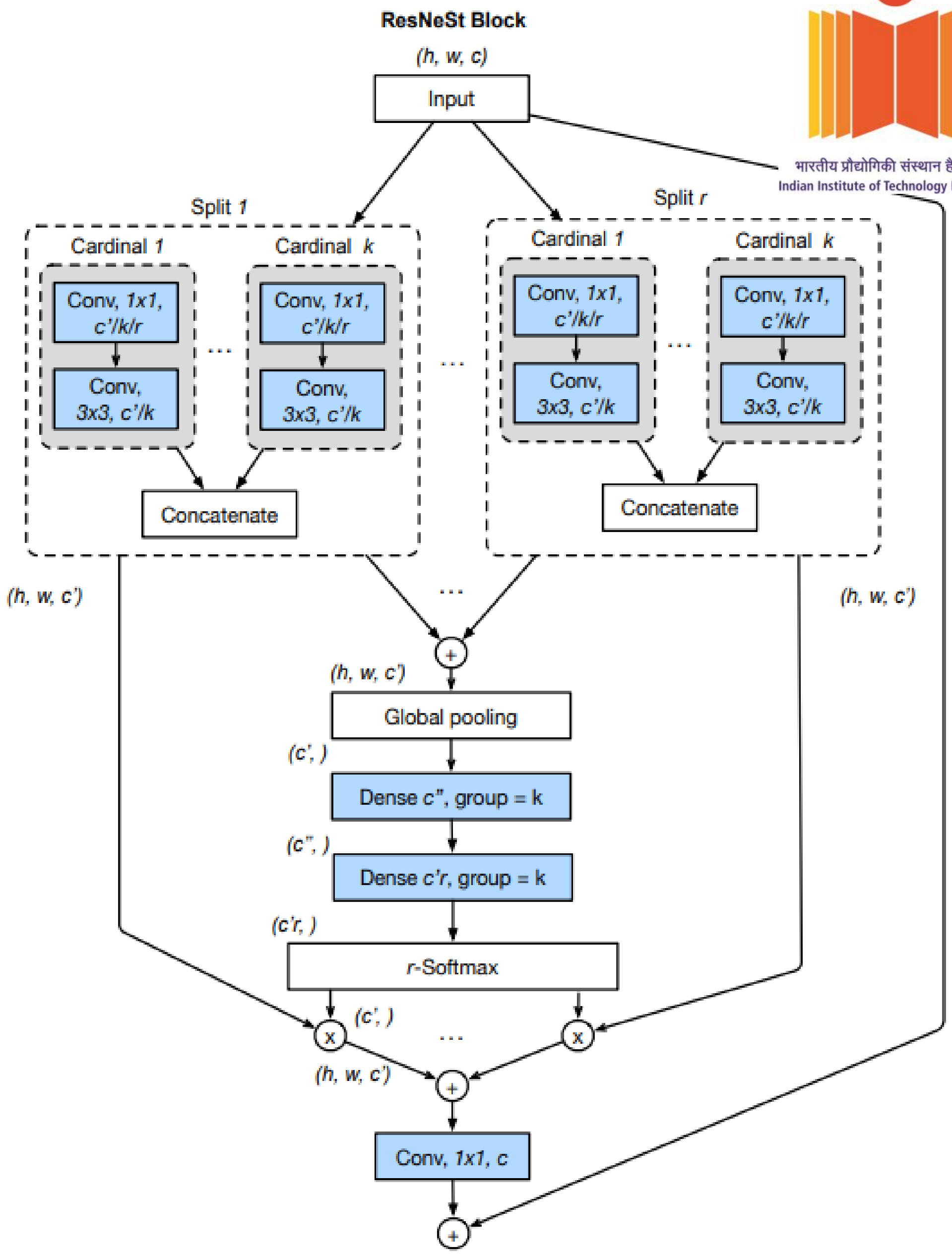
$$V_c^k = \sum_{i=1}^{R} a_i^k(c) U_{R(k-1)+i}$$



ResNeSt:

- Features are divided into several groups
  - Cardinality hyperparameter: K
  - Radix hyperparameter: R
  - Total number of feature groups: G = RK

- Element-wise sum across multiple splits
  - Feature-map groups with the same cardinality index but different radix index are fused together.



ResNeSt: Network Tweaks

- Average Downsampling.

In terms of preserving spatial information, zero padding is suboptimal. Instead of using strided convolution at the transitioning block, use average pooling layer.

ResNeSt: Network Tweaks

- Tweaks from ResNet-D

I. The first 7x7 convolutional layer is replaced with three consecutive 3x3 layers, which have the same receptive field size with a lesser computational cost.

parameters: $3 \times 3^2C^2$ (vs) 7^2C^2

II. 2x2 average pooling layer is added to the shortcut connection prior to the 1x1 convolutional layer for the transitioning blocks.

ResNeSt: Training Strategy

- Large Mini-batch Distributed Training
- Label Smoothing
- Auto Augmentation
- Mix-up Training
- Large Crop Size
- Regularization

ResNeSt: Large Mini-batch Distributed Training

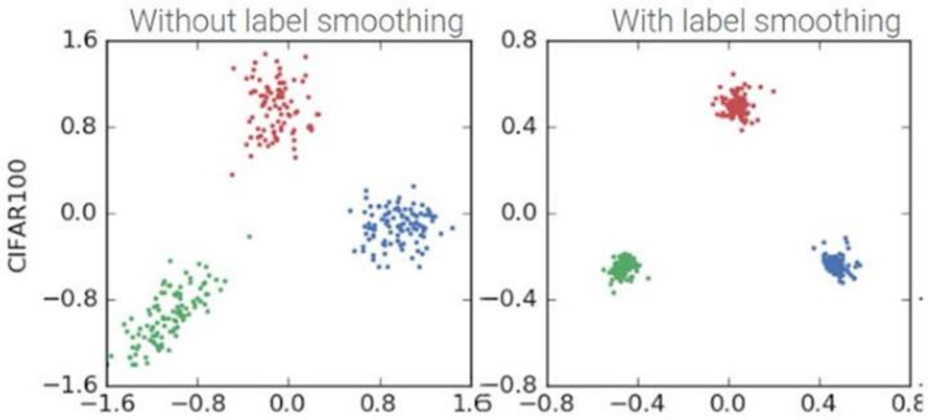
- Distributed training on 8 servers(64 GPU in total).
- Used cosine scheduling, and linearly scaled-up the initial learning rate based on the mini batch size ($\eta = B/256 * \eta_{base}$)

where, B is the mini-batch size, and we use $\eta_{base} = 0.1$ as the base learning rate

ResNeSt: Label Smoothing

Label Smoothing is a regularization technique that introduces noise for the labels.

This accounts for the fact that datasets may have mistakes in them, so maximizing the likelihood of $\log p(Y/X)$ directly can be harmful.



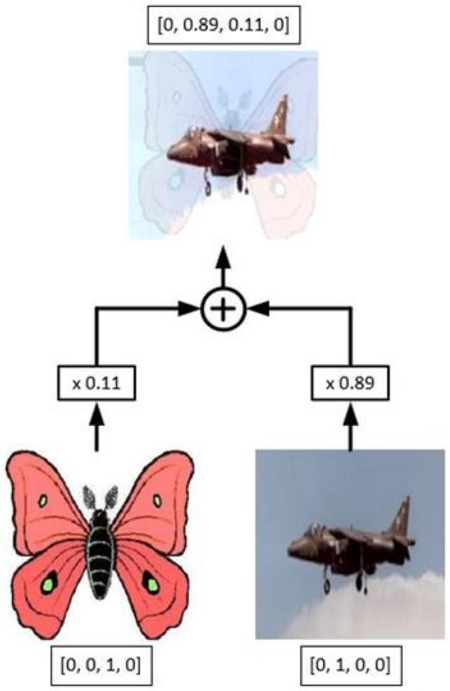
ResNeSt: Auto-Augment

The idea of Auto-Augment is to learn the best augmentation policies for a given dataset. 16 different types of image jittering transformations are introduced, and from these, one augments the data based on 24 different combinations of two consecutive transformations such as shift, rotation, and color jittering.

ResNeSt: Mixup

It is another data augmentation strategy that generates a weighted combinations of random image pairs from the training data. Given two images and their ground truth labels: (x^i, y^i), (x^j, y^j) a synthetic training example $(\hat{x}^i, \hat{y}^i)$ is generated as:

$$
\begin{aligned}
\hat{x} &= \lambda x^i + (1-\lambda) x^j, \\
\hat{y} &= \lambda y^i + (1-\lambda) y^j,
\end{aligned}
$$



ResNeSt: Large Crop Size

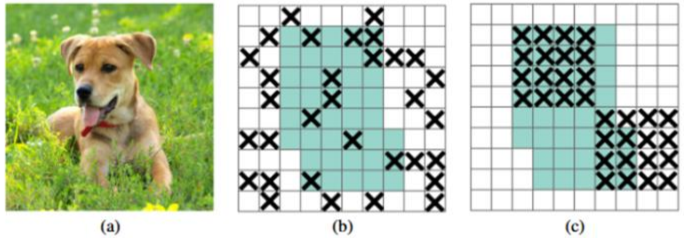
- Large Crop Size: -
  - For fair comparison, we use a crop size of 224 when comparing our ResNeSt with ResNet variants, and a crop size of 256 when comparing with other approaches.
  - EfficientNet method has demonstrated that increasing the input image size for a deeper and wider network may better trade off accuracy vs. FLOPS.

ResNeSt: Regularization

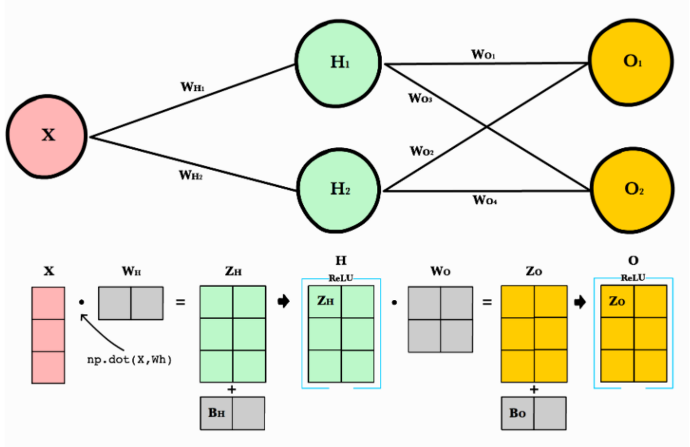
- Regularization: -

I. A dropout layer with the dropout probability of 0.2 is applied before the final fully-connected layer to the networks with more than 200 layers.

II. Also Applied DropBlock layers to the convolutional layers at the last two stages of the network, which is more effective than dropout for specifically regularizing layers.



(a) Is an input image, the green regions in (b) and (c) include the activation units which contain semantic information in the input image.



ResNeSt: - Summary

Explored a simple architectural modification of the ResNet called ResNeSt

- Requires no additional computation and is easy to be adopted as a backbone for other vision tasks.
- Set large scale benchmarks on image classifications and transfer learning applications. Tested on image classification, object detection, instance segmentation, and semantic segmentation.

ResNeSt : Ablation study observations

	#P	GFLOPs	acc(%)
ResNetD-50 [26]	25.6M	4.34	78.31
+ mixup	25.6M	4.34	79.15
+ autoaug	25.6M	4.34	79.41
ResNeSt-50-fast	27.5M	4.34	80.64
ResNeSt-50	27.5M	5.39	81.13

Variant	#P	GFLOPs	img/sec	acc(%)
0s1x64d	25.6M	4.34	688.2	79.41
1s1x64d	26.3M	4.34	617.6	80.35
2s1x64d	27.5M	4.34	533.0	80.64
4s1x64d	31.9M	4.35	458.3	80.90
2s2x40d	26.9M	4.38	481.8	81.00

Ablation study for ImageNet image classification. (Left) breakdown of improvements. (Right) radix vs. cardinality under ResNeSt-fast setting. 2s2x40d denotes radix=2, cardinality=2 and width=40.

ResNeSt : Results (Image classification)

	#P	crop	img/sec	acc(%)
ResNeSt-101(ours)	48M	256	**291.3**	**83.0**
EfficientNet-B4 [56]	19M	380	149.3	83.0
SENet-154 [27]	146M	320	133.8	82.7
NASNet-A [78]	89M	331	103.3	82.7
AmoebaNet-A [46]	87M	299	-	82.8
ResNeSt-200 (ours)	70M	320	**105.3**	**83.9**
EfficientNet-B5 [56]	30M	456	84.3	83.7
AmoebaNet-C [46]	155M	299	-	83.5
ResNeSt-269 (ours)	111M	416	**51.2**	**84.5**
GPipe	557M	-	-	84.3
EfficientNet-B7 [56]	66M	600	34.9	84.4

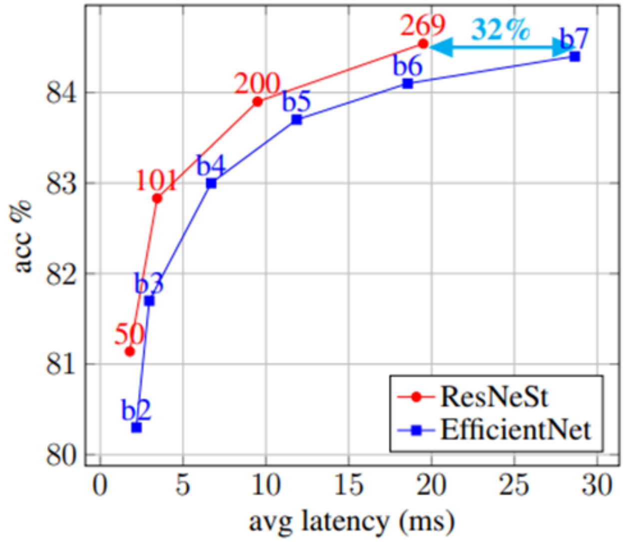
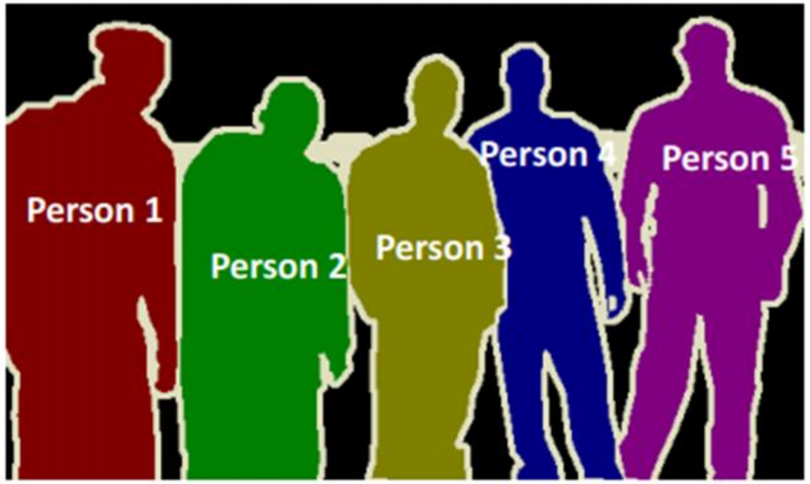


Figure : ResNeSt outperforms EfficientNet in accuracy-latency trade-offs on GPU.

ResNeSt: Results (Instance Segmentation)

	Method	Backbone	box mAP%	mask mAP%
Prior Work	DCV-V2 [76]	ResNet50	42.7	37.0
	HTC [6]	ResNet50	43.2	38.0
	Mask-RCNN [22]	ResNet101 [7]	39.9	36.1
	Cascade-RCNN [5]	ResNet101	44.8	38.0
Our Results	Mask-RCNN [22]	ResNet50 [60]	39.97	36.05
		ResNet101 [60]	41.78	37.51
		ResNeSt50 (ours)	42.81	38.14
		ResNeSt101 (ours)	**45.75**	**40.65**
	Cascade-RCNN [4]	ResNet50 [60]	43.06	37.19
		ResNet101 [60]	44.79	38.52
		ResNeSt50 (ours)	46.19	39.55
		ResNeSt101 (ours)	**48.30**	**41.56**

Instance Segmentation results on the MS-COCO validation set.

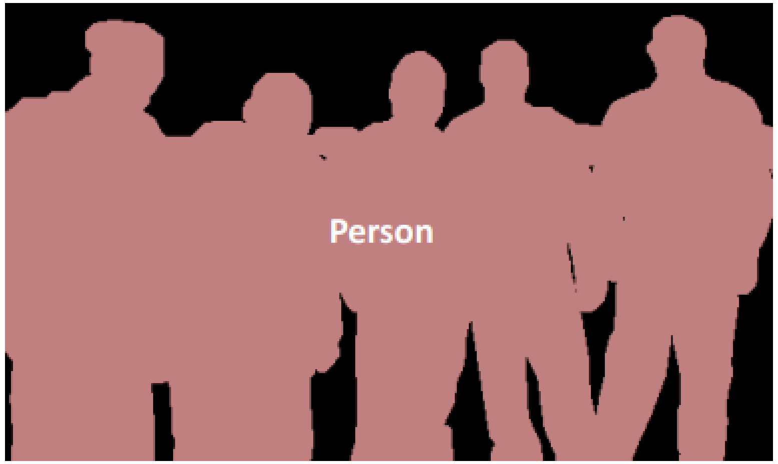


Instance Segmentation

ResNeSt: Results (Semantic Segmentation)

	Method	Backbone	pixAcc%	mIoU%
Prior Work	UperNet [62]	ResNet101	81.01	42.66
	PSPNet [71]	ResNet101	81.39	43.29
	EncNet [68]	ResNet101	81.69	44.65
	CFNet [69]	ResNet101	81.57	44.89
	OCNet [66]	ResNet101	-	45.45
	ACNet [16]	ResNet101	81.96	45.90
Ours	DeeplabV3 [9]	ResNet50 [19]	80.39	42.1
		ResNet101 [19]	81.11	44.14
		ResNeSt-50 (ours)	81.17	45.12
		ResNeSt-101 (ours)	**82.07**	**46.91**
		ResNeSt-200 (ours)	82.45	48.36

Semantic segmentation results on validation set of: ADE20K

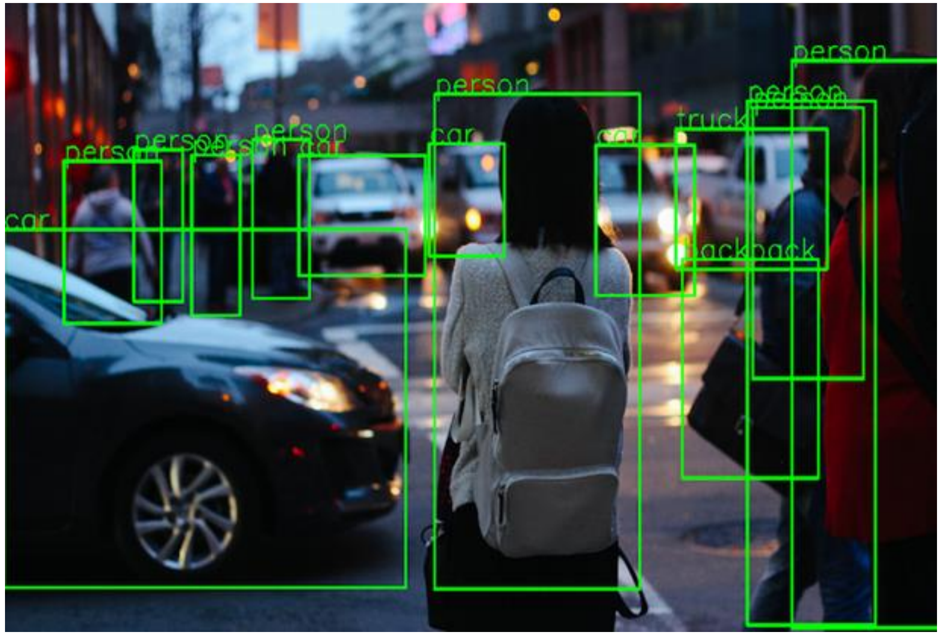


Semantic Segmentation

ResNeSt : Results (Object Detection)

	Method	Backbone	mAP%
Prior Work	Faster-RCNN [47]	ResNet101 [22]	37.3
		ResNeXt101 [7, 63]	40.1
		SE-ResNet101 [27]	41.9
	Faster-RCNN+DCN [13]	ResNet101 [7]	42.1
	Cascade-RCNN [4]	ResNet101	42.8
Our Results	Faster-RCNN [47]	ResNet50 [60]	39.25
		ResNet101 [60]	41.37
		ResNeSt50 (ours)	42.33
		ResNeSt101 (ours)	**44.72**
	Cascade-RCNN [4]	ResNet50 [60]	42.52
		ResNet101 [60]	44.03
		ResNeSt50 (ours)	45.41
		ResNeSt101 (ours)	**47.50**
	Cascade-RCNN [4]	ResNeSt200 (ours)	49.03

Object detection results on the MS-COCO validation set



Conclusion

- ResNeSt's Split-Attention block universally improved the learned feature representations to boost performance.
- In the downstream tasks, simply switching the backbone network to ResNeSt showed substantially better result.
- Depth-wise convolution is not optimal for training and inference efficiency on GPU.
- Increasing input image size can get better accuracy and FLOPS trade-off.

Future Work

- Implementation Of ResNeSt.
- Tweaking DropBlock Position.
- As ResNeSt is modification over ResNet, We can look go for similar modifications of other architectures as well.
- Will be looking for other recent developments that can be incorporated with ResNeSt.

Reference

https://towardsdatascience.com/squeeze-and-excitation-networks-9ef5e71eacd7

https://arxiv.org/abs/2004.08955

ResNeSt — A replacement for Resnet in Computer Vision | by Trung Thanh Tran (Mr. T) | Medium

An Overview of ResNet and its Variants | by Vincent Fung | Towards Data Science

Thank You