

$$y_k = f(z_k)$$

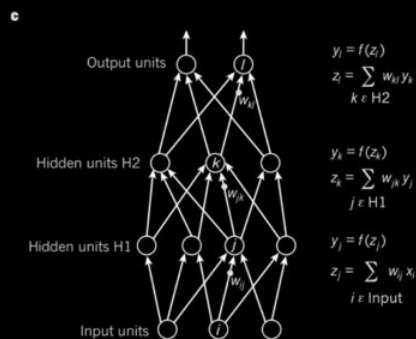
$$z_k = \sum_{l \in H2} w_{kl} y_l$$

$$y_k = f(z_k)$$

$$z_k = \sum_{j \in H1} w_{jk} y_j$$

$$y_j = f(z_j)$$

$$z_j = \sum_{i \in \text{Input}} w_{ij} x_i$$



$$y_k = f(z_k)$$

$$z_k = \sum_{l \in H2} w_{kl} y_l$$

$$y_k = f(z_k)$$

$$z_k = \sum_{j \in H1} w_{jk} y_j$$

$$y_j = f(z_j)$$

$$z_j = \sum_{i \in \text{Input}} w_{ij} x_i$$

d

Compare outputs with correct answer to get error derivatives

MONTHLY PAPER READING SESSIONS

$$\frac{\partial E}{\partial y_k} = \sum_{l \in \text{out}} w_{kl} \frac{\partial E}{\partial z_l}$$

$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_k}$$

MLT__init__

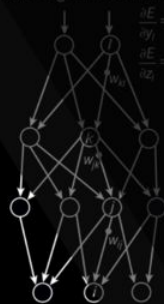
PAPER READING & DISCUSSION

d

Compare outputs with correct answer to get error derivatives

$$\frac{\partial E}{\partial y_k} = \sum_{l \in \text{out}} w_{kl} \frac{\partial E}{\partial z_l}$$

$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_k}$$



$$\frac{\partial E}{\partial y_l} = \sum_{k \in H2} w_{kl} \frac{\partial E}{\partial z_k}$$

$$\frac{\partial E}{\partial z_l} = \frac{\partial E}{\partial y_l} \frac{\partial y_l}{\partial z_l}$$



MACHINE
LEARNING
TOKYO

Session # 3: Squeeze & Excitation Nets

Alisher Abdulkhaev | 2021.03.14



@alisher_ai

Self Introduction

- ❑ Name: **Alisher Abdulkhaev**
- ❑ Board Director @ **Machine Learning Tokyo (MLT)**
- ❑ Computer Vision Engineer @ **Browzzin, Inc**
(AI powered social fashion App)
- ❑ Ph.D. student @ **CVLAB**, University of Tsukuba (supervisor: **Prof. Fukui**)
- ❑ Teaching Fellow @ **Tokyo Data Science** (online data science school
executed by Michal Fabinger)
- ❑ Twitter: **@alisher_ai**



Outline

- ❑ Universal Approximation Theorem — Neural Networks
- ❑ Inductive Biases
- ❑ Squeeze-and-Excitation Networks (SE Nets)
 - ❑ Introduction
 - ❑ Squeeze-and-Excitation Block
 - ❑ Results
 - ❑ Conclusions
- ❑ Suggested links & PwA
- ❑ Free discussion + Q&A

Neural Networks* & Universal Approximation Theorem

- ❑ NN aim to find any mathematical function or (series of) transformation which map an input (x) to output (y);

$$y = f(x)$$

- ❑ The Universal Approximation Theorem tells us that Neural Networks (with sufficient number of nodes/neurons and layers) can approximate any function $f(x)$, i.e. NN has a universality

*** Multi Layer Perceptron | Fully Connected NN | Dense NN**

Inductive Bias

The final meaning of *bias* we will consider here is *inductive bias*; in this meaning, all machine learning relies on some sort of bias. This type of bias is frequently implicit in the technique, and is also referred to as *model bias* or *representation bias*. Rather than a source of statistical error, this type of bias can be thought of as a set of assumptions being made (usually *a priori*) about which possible solutions are worth considering, which should be preferred, and which can be ignored. Without a bias of this type, machine learning not would be possible.

The spatial inductive bias of deep learning, Benjamin R. Mitchell

- ❑ Instead of considering huge neural networks for all kind of data and tasks, we would prefer to make some assumptions on the nature of the data.

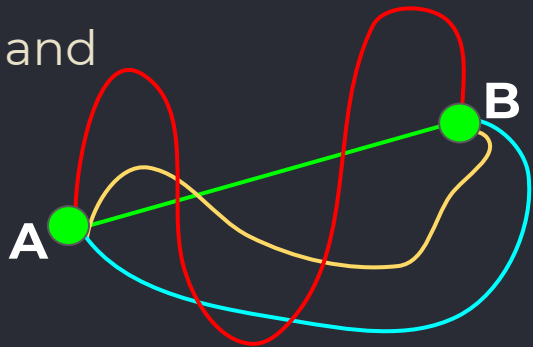
Without an Inductive Bias

- ❑ Without some form of inductive bias to restrict the hypothesis space;
 - ❑ the basic tasks of machine learning would become impossible: generalization and evaluation,
 - ❑ need a dataset representing the true distribution of the data.

The Need for Biases in Learning Generalizations, Tom Mitchell

With Inductive Bias

- ❑ Restricts the hypothesis space, i.e. ease the neural network design depending on the data being worked on.
- ❑ A number of possible models crossing points A and B is infinitesimal. Inductive bias would help to limit the choices (e.g., **linear model**)
- ❑ Less data and less computation



Examples for Inductive Biases

- ❑ Occam's Razor: a bias towards the simplicity.
- ❑ Graph NN — relational inductive bias
- ❑ Recurrent NN — recurrent inductive bias (sequential input)
- ❑ Transformers process the input in parallel, not sequentially...

Samira Abnar's Blog Post: <https://samiraabnar.github.io/articles/2020-05/recurrence>

- ❑ Convolutional Neural Networks ...


CNN as an Inductive Bias

- ❑ When we work on data with spatial dimensions, we can leverage the prior belief/knowledge of visual data.
- ❑ CNNs are special type of NN which have a very powerful inductive bias — spatial structure such as local connectivity.
- ❑ I think, all CNN architecture designs try to enhance the inductive bias:
 - ❑ Residual connections
 - ❑ Attention

Squeeze & Excitation Network



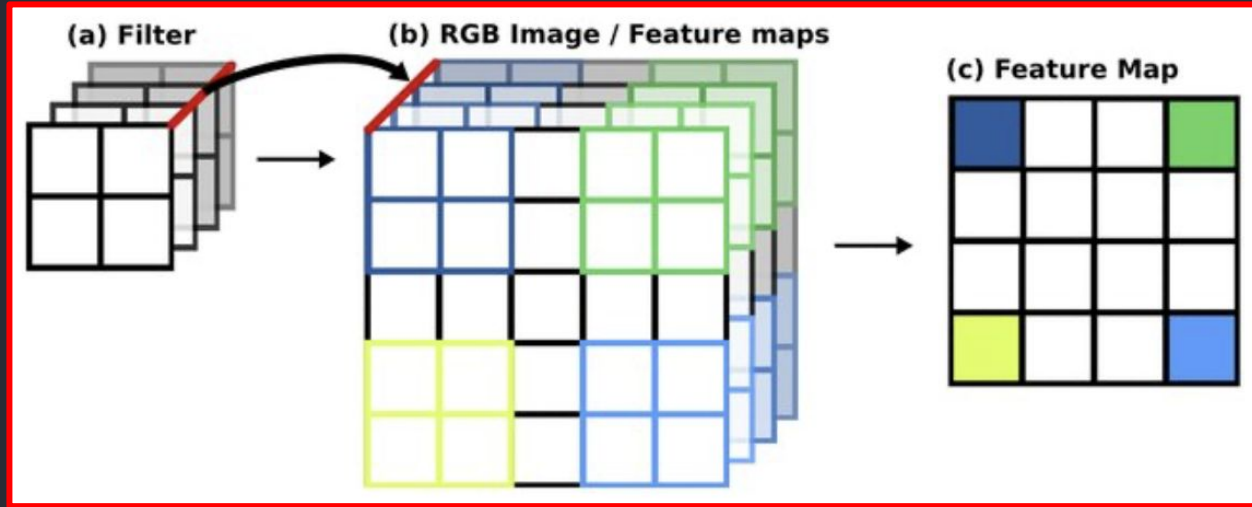
SENet proposes some additional “inductive bias”
— the relationship between feature map
channels and feature recalibration.



SENet: Introduction

- ❑ Central theme in Computer Vision is to search for task and data dependent powerful representation for visual data.
- ❑ Deep Learning is about representation learning...
- ❑ CNN is a DNN consists of convolutional, non-linear activation and pooling and fully connected layers in principle.

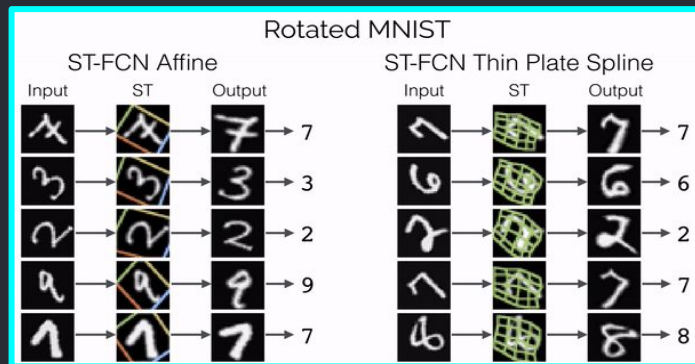
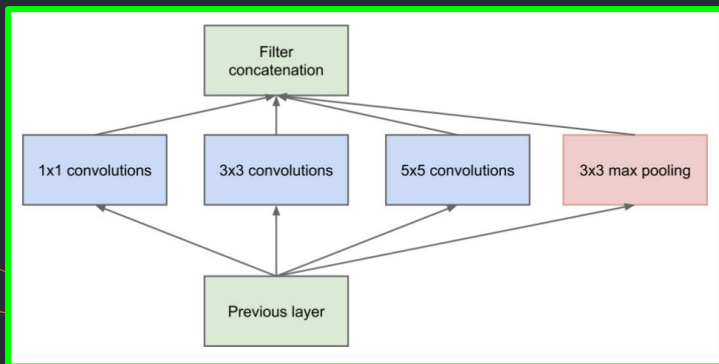
SENet: Introduction



- ❑ Convolutional layers fuse spatial and channel-wise information together.

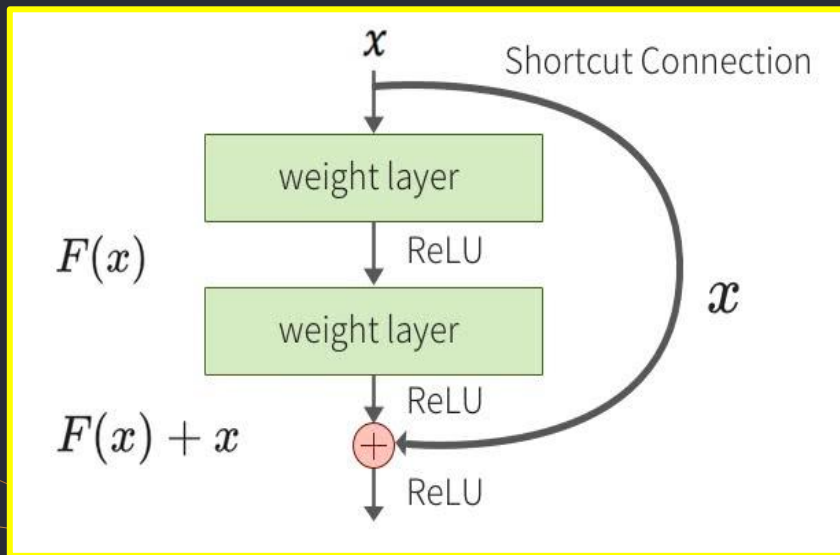
SENet: Introduction

- Representations produced by CNN can be strengthened by integrating learning mechanisms to capture spatial correlations between features.
- Inception family:
 - multi-scale processing
- Spatial Transformer Nets:
 - spatially transform feature maps

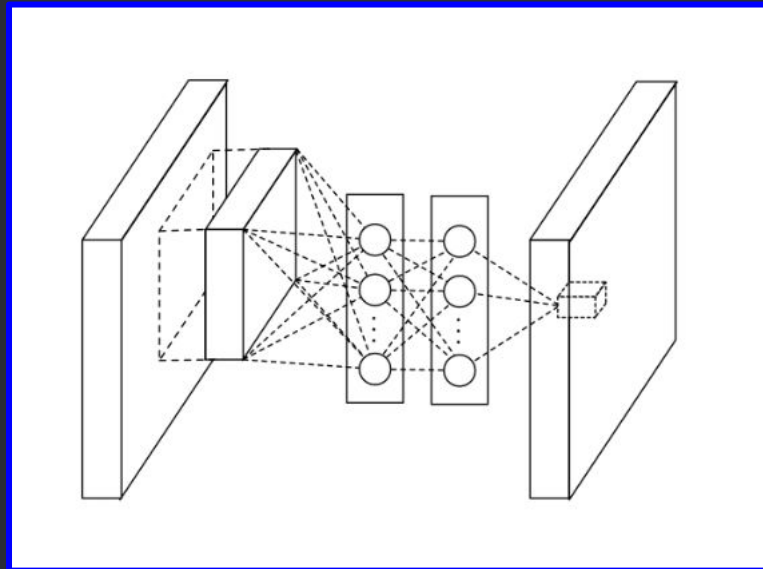


SENet: Introduction

- ResNet: deep and identity based skip connections



- Network in Network: cross-channel correlations

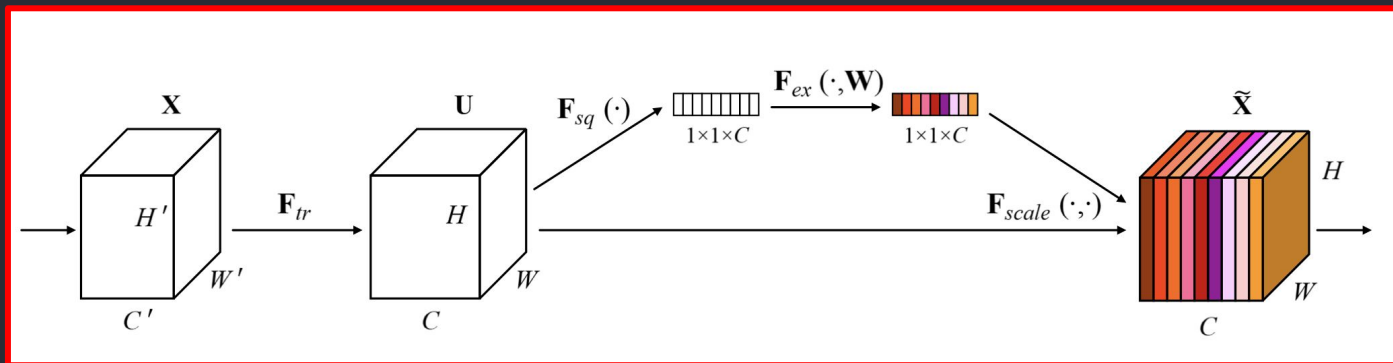


SENet: Introduction

- ❑ Each learned filters operates with a local receptive field — unable to exploit contextual information outside if the region.
- ❑ SENet investigates the relationship between feature map channels.
- ❑ Explicitly models the interdependencies between the channels.
- ❑ Feature recalibration: helps the network to learn global information to selectively **emphasise informative features** while **suppressing the less useful** ones which ease the learning process.

Squeeze-and-Excitation Block

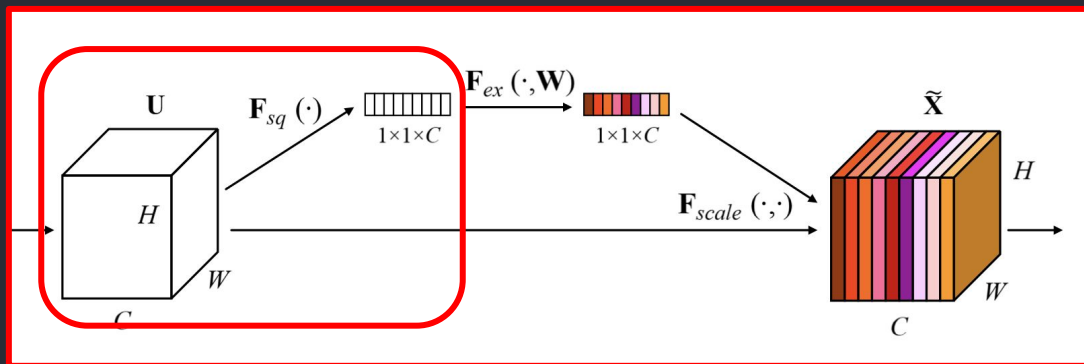
- SE Block intrinsically introduce dynamics conditioned on input — self-attention function.
- SE Block excites:
 - class-agnostic informative features at earlier layers,
 - class-specific informative features at later layers.



Squeeze-and-Excitation Block

❑ Squeeze: global information embedding

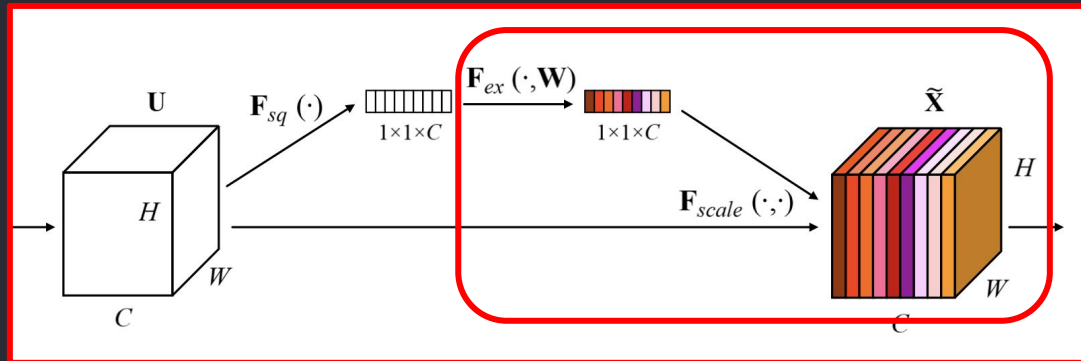
- ❑ Global average pooling: generates channel-wise statistics.
- ❑ Local descriptors with expressive statistics of the whole feature map.



Squeeze-and-Excitation Block

❑ Excitation: adaptive re-calibration

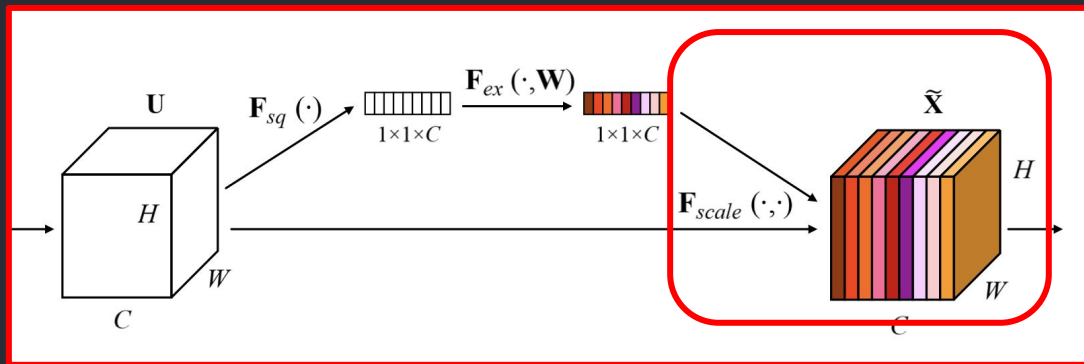
- ❑ Captures the channel-wise dependencies
- ❑ Maps input specific descriptor to a set of channel weights



Squeeze-and-Excitation Block

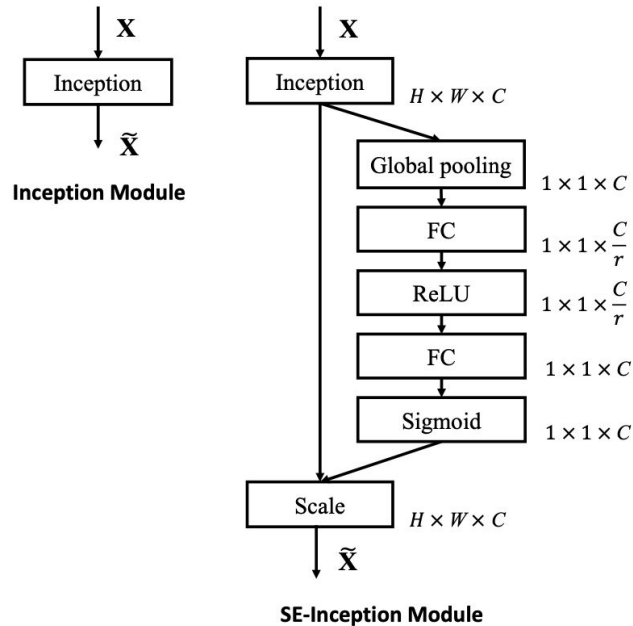
Re-scaling:

- Channel-wise multiplication between the scalar and the input feature map
- Constructs the new tensor with spatial dimension



Squeeze-and-Excitation Block

- SE Block can be used as add-on block for various CNN architectures.



- Input \rightarrow GAP \rightarrow FC \rightarrow ReLU \rightarrow FC \rightarrow sigmoid \rightarrow output

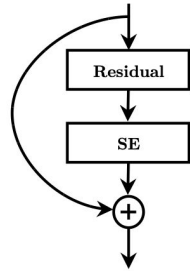
- r : reduction ratio

- Sigmoid: gives the (importance) weights for each channel

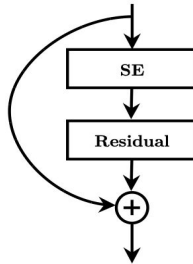
- Network-in-network design

Squeeze-and-Excitation Block

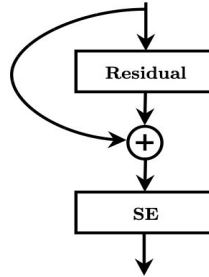
SE Block can be used as add-on block for various CNN architectures



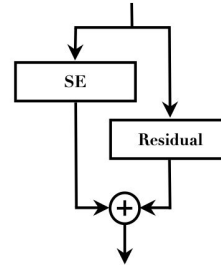
(b) Standard SE block



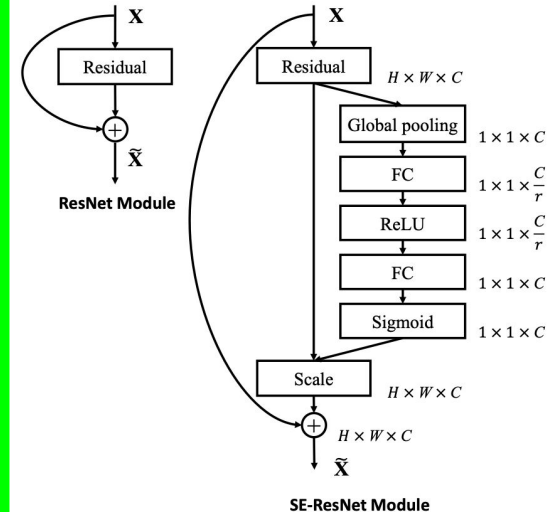
(c) SE-PRE block



(d) SE-POST block



(e) SE-Identity block



Results: ImageNet

- ❑ The validation set error-rates of CNN architectures and their SE counterparts:

	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [13]	24.7	7.8	24.80	7.48	3.86	23.29 _(1.51)	6.62 _(0.86)	3.87
ResNet-101 [13]	23.6	7.1	23.17	6.52	7.58	22.38 _(0.79)	6.07 _(0.45)	7.60
ResNet-152 [13]	23.0	6.7	22.42	6.34	11.30	21.57 _(0.85)	5.73 _(0.61)	11.32
ResNeXt-50 [19]	22.2	-	22.11	5.90	4.24	21.10 _(1.01)	5.49 _(0.41)	4.25
ResNeXt-101 [19]	21.2	5.6	21.18	5.57	7.99	20.70 _(0.48)	5.01 _(0.56)	8.00
VGG-16 [11]	-	-	27.02	8.81	15.47	25.22 _(1.80)	7.70 _(1.11)	15.48
BN-Inception [6]	25.2	7.82	25.38	7.89	2.03	24.23 _(1.15)	7.14 _(0.75)	2.04
Inception-ResNet-v2 [21]	19.9 [†]	4.9 [†]	20.37	5.21	11.75	19.80 _(0.57)	4.79 _(0.42)	11.76

Results: Places365

- ❑ Places 365 is a scene classification challenge/dataset.
- ❑ Scene understanding assesses the generalization and abstraction ability.

TABLE 6
Single-crop error rates (%) on Places365 validation set.

	top-1 err.	top-5 err.
Places-365-CNN [72]	41.07	11.48
ResNet-152 (ours)	41.15	11.61
SE-ResNet-152	40.37	11.01

Results: COCO

- ❑ MS COCO is an object detection challenge/dataset.
- ❑ Object detection is another assessment to measure the model's localization ability.

TABLE 7
Faster R-CNN object detection results (%) on COCO *minival* set.

	AP@IoU=0.5	AP
ResNet-50	57.9	38.0
SE-ResNet-50	61.0	40.4
ResNet-101	60.1	39.9
SE-ResNet-101	62.7	41.9

Conclusion

- ❑ SE Block: an architectural unit which improves the representational power of a network.
- ❑ Enables to perform dynamic channel-wise feature calibration.
- ❑ SE Blocks shed light on the inability of previous CNN architectures to model channel-wise feature dependencies..
- ❑ SE Blocks may help to advance the network pruning by selecting the most informative filters and/or feature maps.
- ❑ Can SE Blocks be used effectively during the test time — excites the particular channels depending on the input image?

Suggested Links:

- ❑ Paper itself: [Squeeze-and-Excitation Networks, CVPR-2018](#)
- ❑ [Squeeze and Excitation Networks Explained with PyTorch Implementation](#)
- ❑ [Squeeze-and-Excitation Networks](#) by Paul-Louis Prove
- ❑ [Squeeze-and-Excitation Networks](#) by Rachel Draelos
- ❑ SENet Pytorch Implementation: [SENet-PyTorch](#)
- ❑ [The need for biases in learning generalizations](#)
- ❑ [The spatial inductive bias of deep learning](#)
- ❑ [Distilling inductive biases](#)

Last words

- ❑ Papers with Annotations (PwA)
- ❑ PwA version of the “Squeeze-and-Excitation Networks” paper is available at:
https://github.com/Machine-Learning-Tokyo/papers-with-annotations/blob/master/convolutional-neural-networks/Squeeze-and-Excitation_Networks.pdf

TL;DR:

- Squeeze-and-Excitation Block has been proposed as an add-on to various CNN archs with ease.
- SE Block investigates the relationship between feature map channels
- SE Block emphasise informative features while suppressing the less useful ones.

Squeeze-and-Excitation Networks

Jie Hu^{1*}

hujie@momenta.ai

¹ Momenta

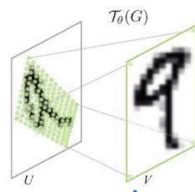
Li Shen^{2*}

lishen@robots.ox.ac.uk

² Department of Engineering Science, University of Oxford







Gang Sun¹

sungang@momenta.ai



THANK YOU FOR LISTENING!

Sessions

Date	Topic	Paper	Presenter	Video
10/Jan/2021	CV: Separable Convolutions	Xception	Jayson Cunanan	
14/Feb/2021	CV: Dilated Convolutions + ASPP	DeepLabv2	J. Miguel Valverde	
14/Mar/2021	CV: Attention in Images	Squeeze and Excitation	Alisher Abdulkhaev	
11/Apr/2021	CV: Attention in GANs	SAGAN	Mayank Bhaskar	
9/May/2021	NLP: Attention	RNN encoder-decoder for SMT	Ana Valeria	
13/Jun/2021	NLP: Attention	Sequence to Sequence Learning with Neural Networks	Charles Melby-Thompson	

Sessions will be held via Zoom starting at 5pm (JST) / 9am (CET). Check at what time is in your region [here](#).



Q & A