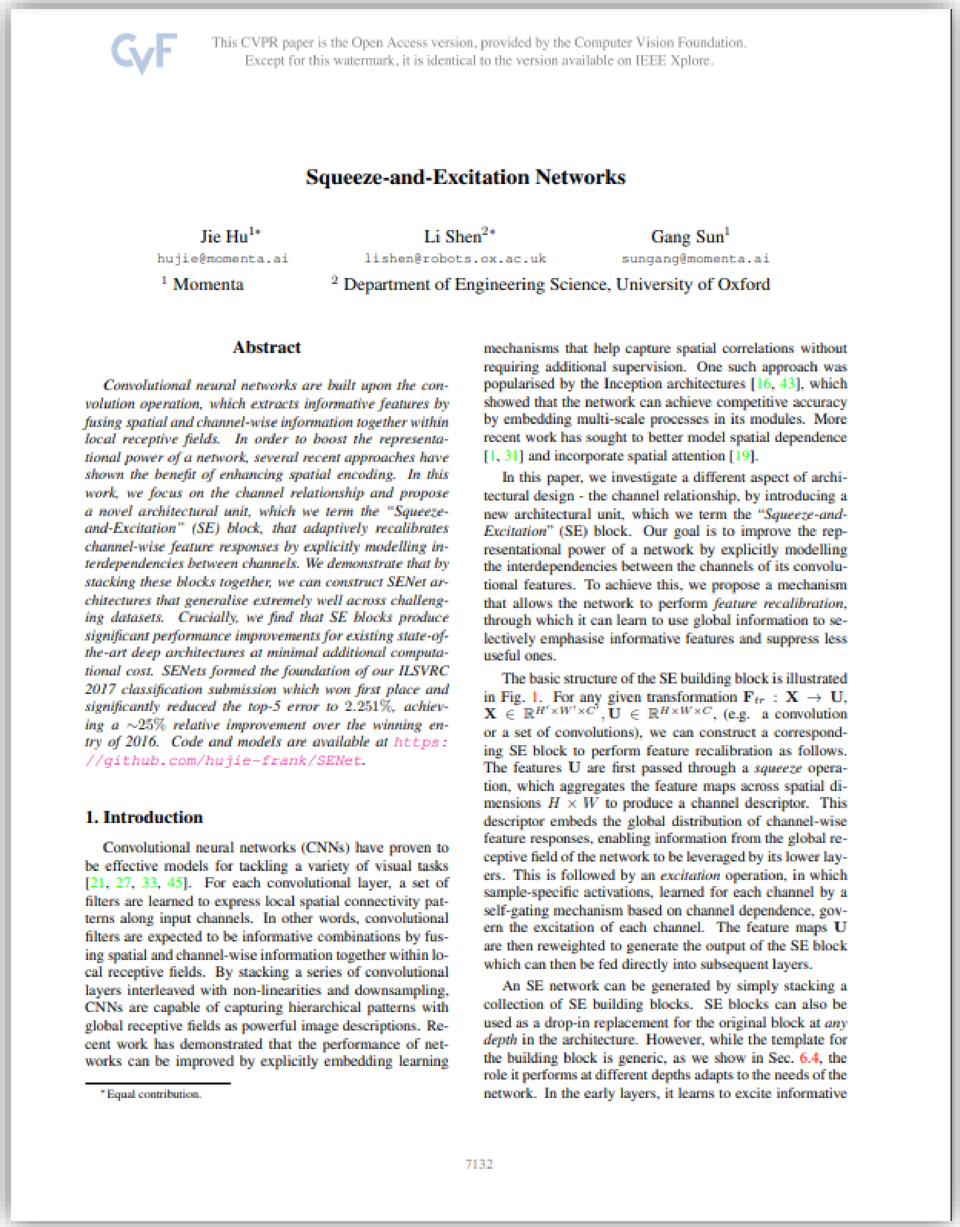
SQUEEZE-AND-EXCITATION NETWORKS REVIEW

2023.07.?? 김채원



Squeeze-and-Excitation Networks

Jie Hu[1*] hujie@momenta.ai
Li Shen[2*] lishen@robots.ox.ac.uk
Gang Sun[1] sungang@momenta.ai

[1] Momenta
[2] Department of Engineering Science, University of Oxford

Abstract

Convolutional neural networks are built upon the convolution operation, which extracts informative features by fusing spatial and channel-wise information together within local receptive fields. In order to boost the representational power of a network, several recent approaches have shown the benefit of enhancing spatial encoding. In this work, we focus on the channel relationship and propose a novel architectural unit, which we term the "Squeeze-and-Excitation" (SE) block, that adaptively recalibrates channel-wise feature responses by explicitly modelling interdependencies between channels. We demonstrate that by stacking these blocks together, we can construct SENet architectures that generalise extremely well across challenging datasets. Crucially, we find that SE blocks produce significant performance improvements for existing state-of-the-art deep architectures at minimal additional computational cost. SENets formed the foundation of our ILSVRC 2017 classification submission which won first place and significantly reduced the top-5 error to 2.251%, achieving a ∼25% relative improvement over the winning entry of 2016. Code and models are available at `https://github.com/hujie-frank/SENet`.

1. Introduction

Convolutional neural networks (CNNs) have proven to be effective models for tackling a variety of visual tasks [21, 27, 33, 45]. For each convolutional layer, a set of filters are learned to express local spatial connectivity patterns along input channels. In other words, convolutional filters are expected to be informative combinations by fusing spatial and channel-wise information together within local receptive fields. By stacking a series of convolutional layers interleaved with non-linearities and downsampling, CNNs are capable of capturing hierarchical patterns with global receptive fields as powerful image descriptions. Recent work has demonstrated that the performance of networks can be improved by explicitly embedding learning mechanisms that help capture spatial correlations without requiring additional supervision. One such approach was popularised by the Inception architectures [16, 43], which showed that the network can achieve competitive accuracy by embedding multi-scale processes in its modules. More recent work has sought to better model spatial dependence [1, 31] and incorporate spatial attention [19].

In this paper, we investigate a different aspect of architectural design - the channel relationship, by introducing a new architectural unit, which we term the "*Squeeze-and-Excitation*" (SE) block. Our goal is to improve the representational power of a network by explicitly modelling the interdependencies between the channels of its convolutional features. To achieve this, we propose a mechanism that allows the network to perform *feature recalibration*, through which it can learn to use global information to selectively emphasise informative features and suppress less useful ones.

The basic structure of the SE building block is illustrated in Fig. 1. For any given transformation $\mathbf{F}_{tr} : \mathbf{X} \rightarrow \mathbf{U}$, $\mathbf{X} \in \mathbb{R}^{H' \times W' \times C'}, \mathbf{U} \in \mathbb{R}^{H \times W \times C}$, (e.g. a convolution or a set of convolutions), we can construct a corresponding SE block to perform feature recalibration as follows. The features $\mathbf{U}$ are first passed through a *squeeze* operation, which aggregates the feature maps across spatial dimensions $H \times W$ to produce a channel descriptor. This descriptor embeds the global distribution of channel-wise feature responses, enabling information from the global receptive field of the network to be leveraged by its lower layers. This is followed by an *excitation* operation, in which sample-specific activations, learned for each channel by a self-gating mechanism based on channel dependence, govern the excitation of each channel. The feature maps $\mathbf{U}$ are then reweighted to generate the output of the SE block which can then be fed directly into subsequent layers.

An SE network can be generated by simply stacking a collection of SE building blocks. SE blocks can also be used as a drop-in replacement for the original block at *any depth* in the architecture. However, while the template for the building block is generic, as we show in Sec. 6.4, the role it performs at different depths adapts to the needs of the network. In the early layers, it learns to excite informative

[*] Equal contribution.

Conference : CVPR (2018)

Published date: 2017.09.05

Author : Jie Hu , Li shen (Momenta)
Gang Sun (Oxford univ.)

1.Abstract

- Existing studies have focused on **spatial encoding** to improve the performance of CNN

- In this paper, they focused on **channel relationship** instead of spatial encoding

- new architecture -> **SENet**

1.Abstract

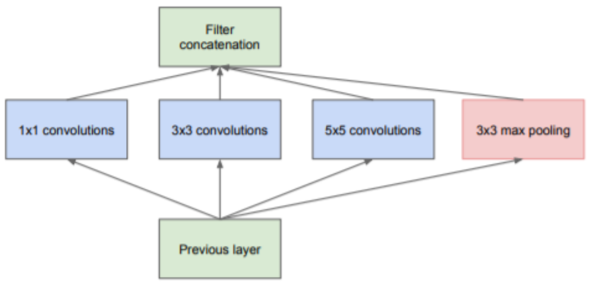
- SENet is a network consisting of **squeeze-and-excitation (SE) blocks**

- It recalculates the channel-wise feature according to its importance by reflecting the interdependency between channels

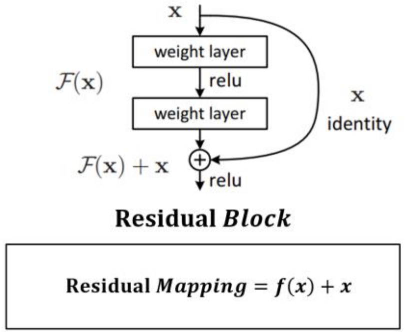
- It's easy to apply to existing architectures, and good performance improvements over complexity growth high

1.Introduction

- Existing studies have focused on spatial encoding to improve the performance of CNN



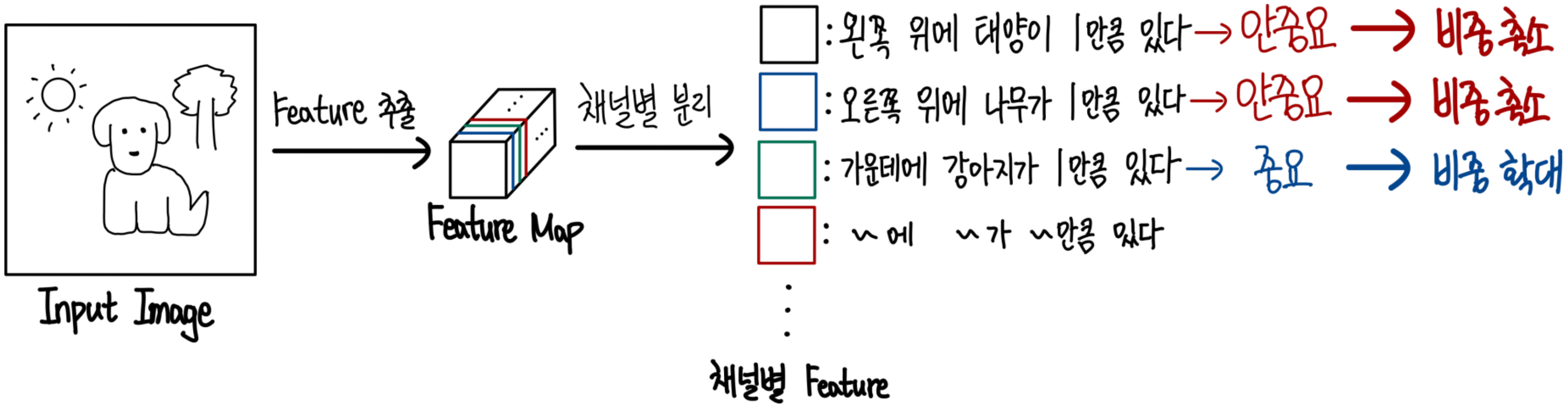
Inception block


x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu
Residual Block
Residual Mapping = $f(x) + x$

Residual block

1.Introduction

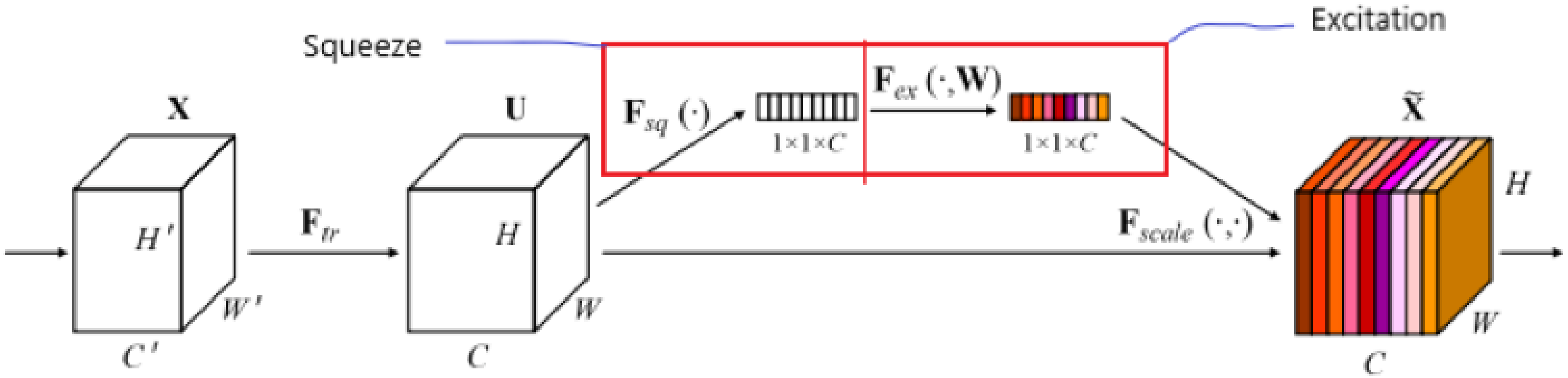
- However, CNN has no feature-specific attention and ignore function


Input Image
Feature 추출
Feature Map
채널별 분리
:왼쪽 위에 대양이 1만큼 있다 → 안중요 → 비중 축소
:오른쪽 위에 나무가 1만큼 있다 → 안중요 → 비중 축소
:가운데에 강아지가 1만큼 있다 → 중요 → 비중 확대
: ~에 ~가 ~만큼 있다
채널별 Feature

- Add attention function to improve performance of CNN

1.Introduction

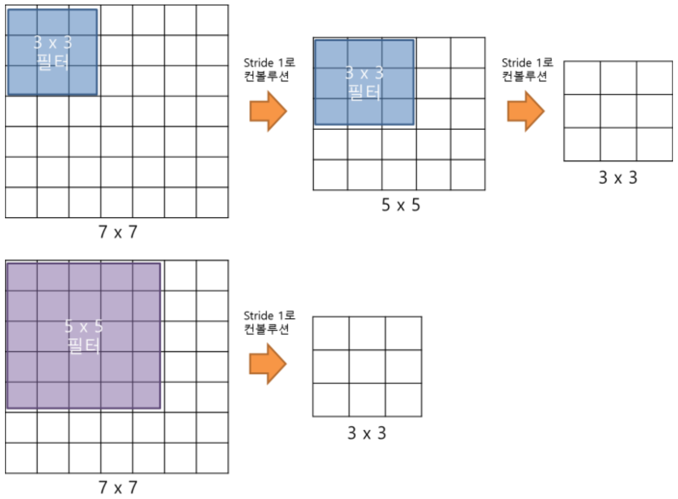
▶ Squeeze-and-Excitation (SE) block



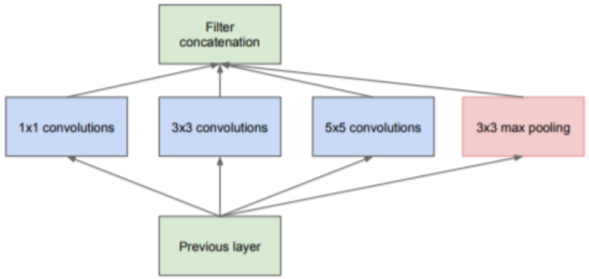
- Focus on channel relationship
- A heavily weighted channel contains important features
- calculate weight by channel

2.Related work

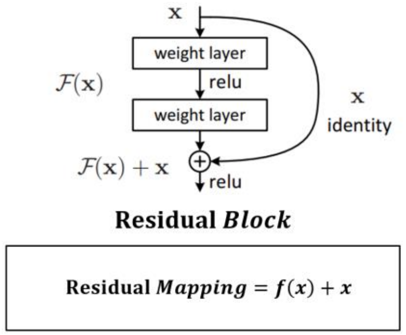
- Deep architectures



VGG net’s 3x3 filter



Inception block



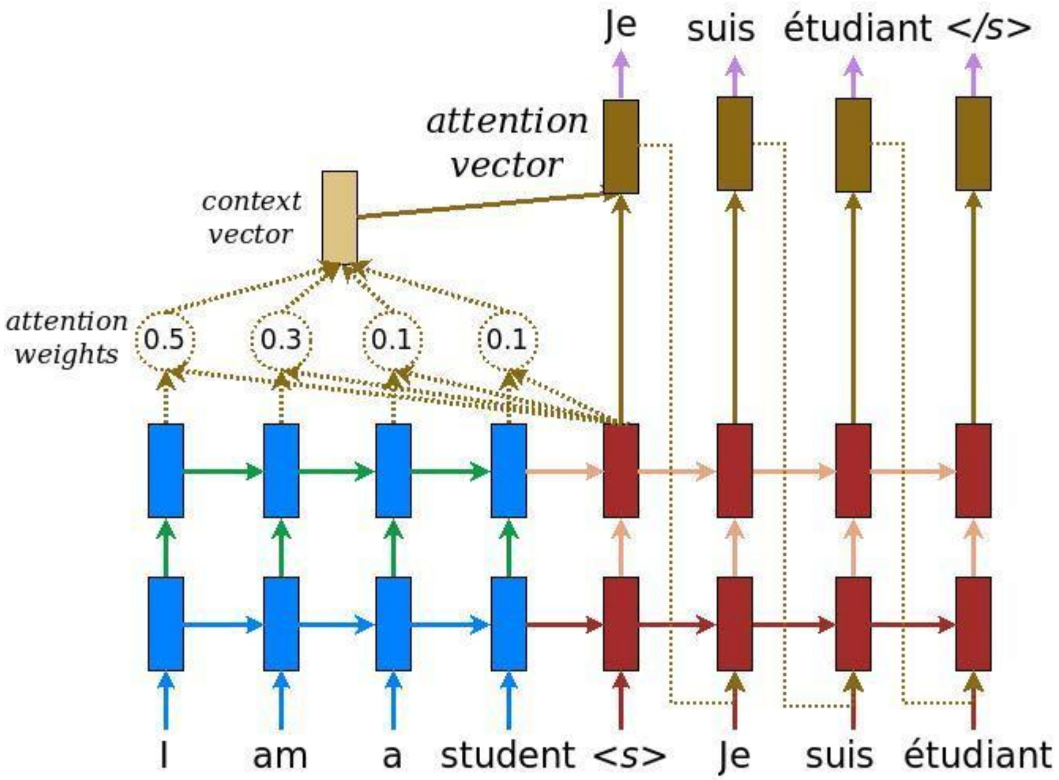
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu
Residual *Block*
Residual *Mapping* $= f(x) + x$

Residual block

2.Related work

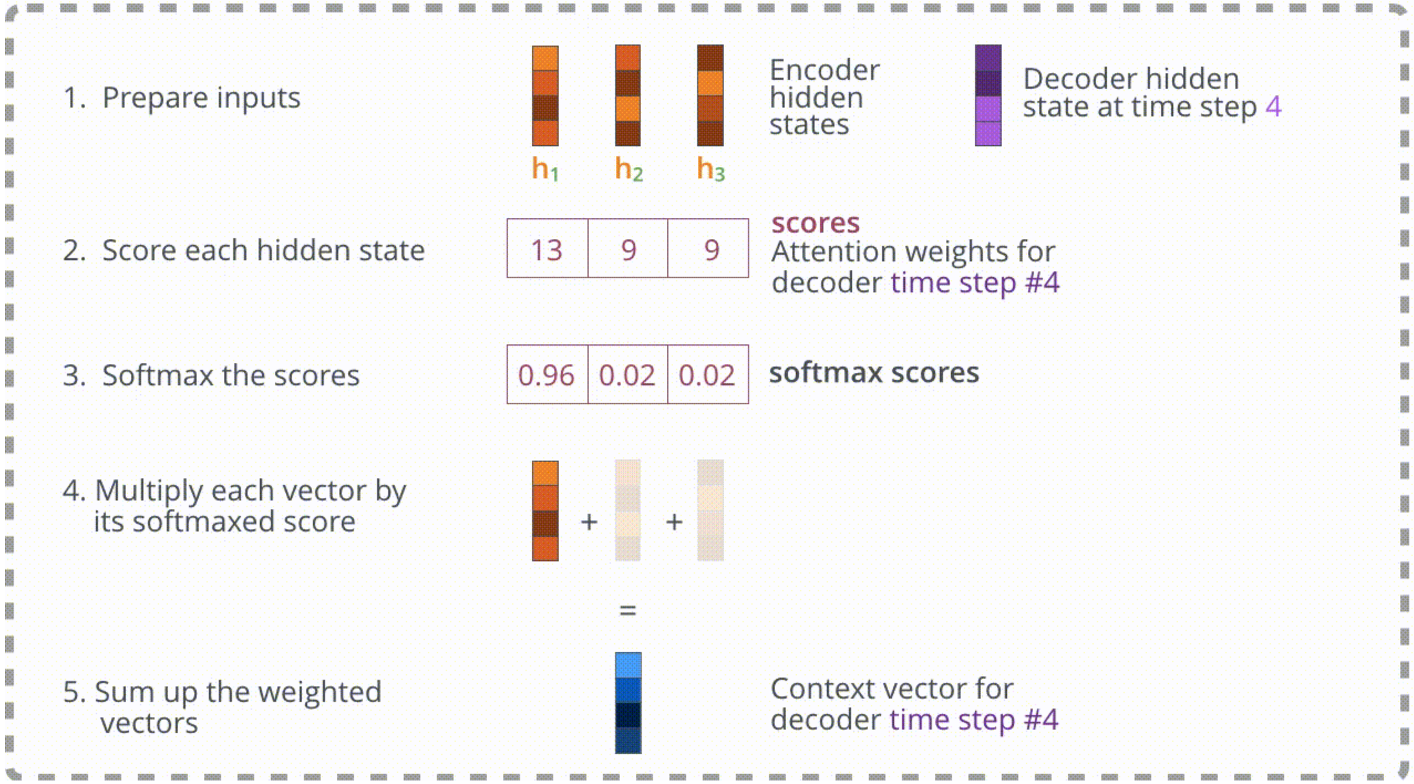
- Attention

- calculate attention weight for each feature



2.Related work

- Attention



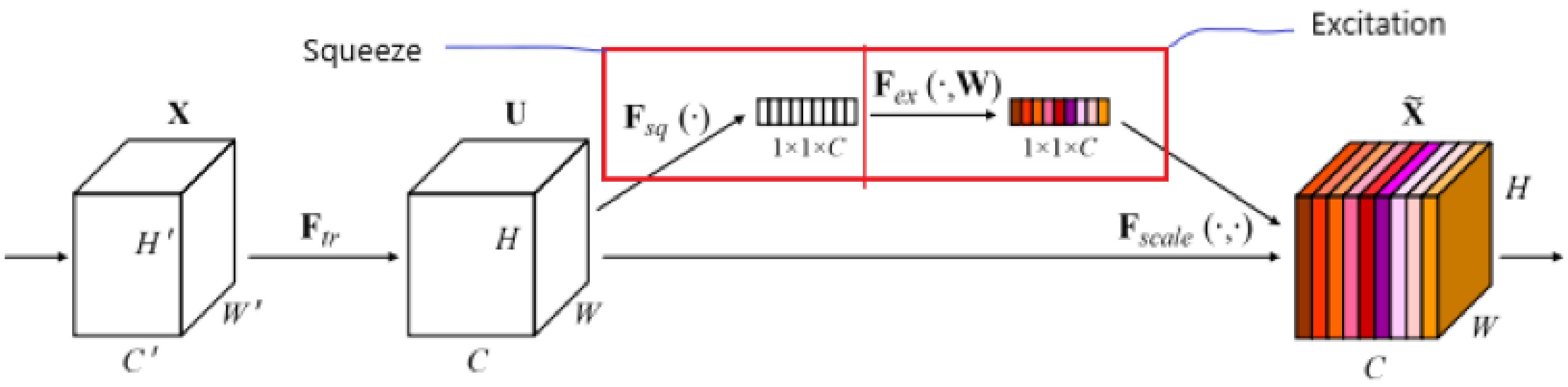
1. Prepare inputs
2. Score each hidden state
3. Softmax the scores
4. Multiply each vector by its softmaxed score
5. Sum up the weighted vectors

h_1
h_2
h_3
Encoder hidden states
Decoder hidden state at time step 4
13
9
9
scores
Attention weights for decoder time step #4
0.96
0.02
0.02
softmax scores
+
+
=
Context vector for decoder time step #4

2.Related work

▶결론

- Focus on channel relationship instead of spatial encoding for better performance on CNN
- To this end, a method called squeeze and extraction (SE) block based on attention mechanism was introduced



3.Method

▶ Other Network

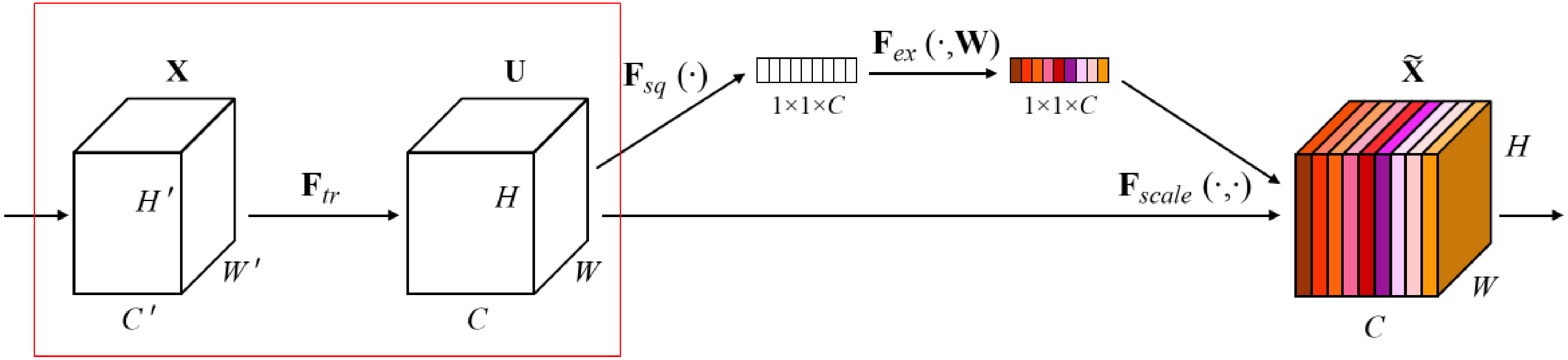


Figure 1: A Squeeze-and-Excitation block.

- other network or convolution operation before SE Block starts (ex. GoogLeNet,VGGNet,ResNet etc...)

3.Method

▶ Squeeze operation

10	1	1
1	1	1
1	1	1

Global Average Pooling → 2

합: 18

평균: 18/9=2

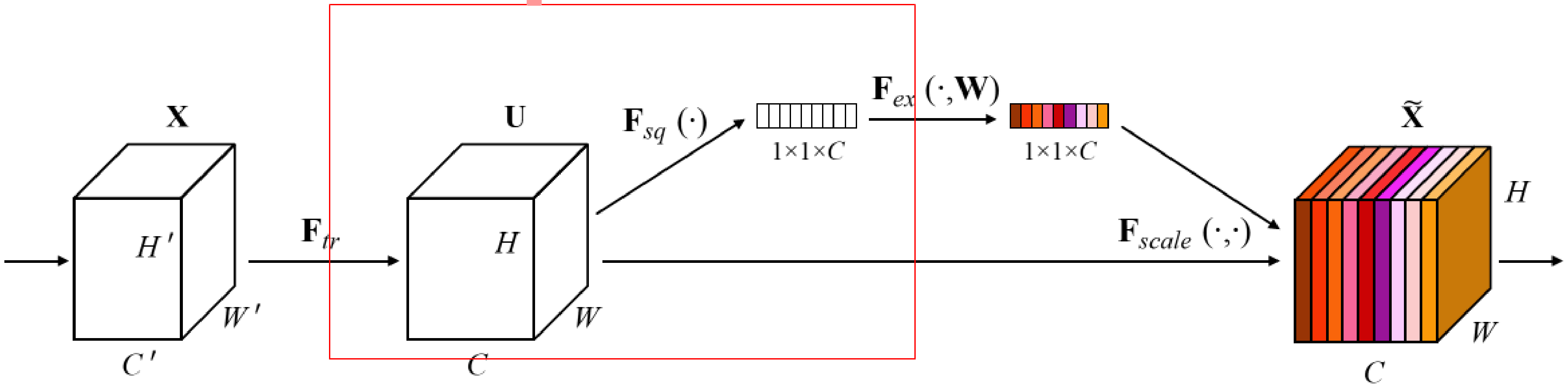
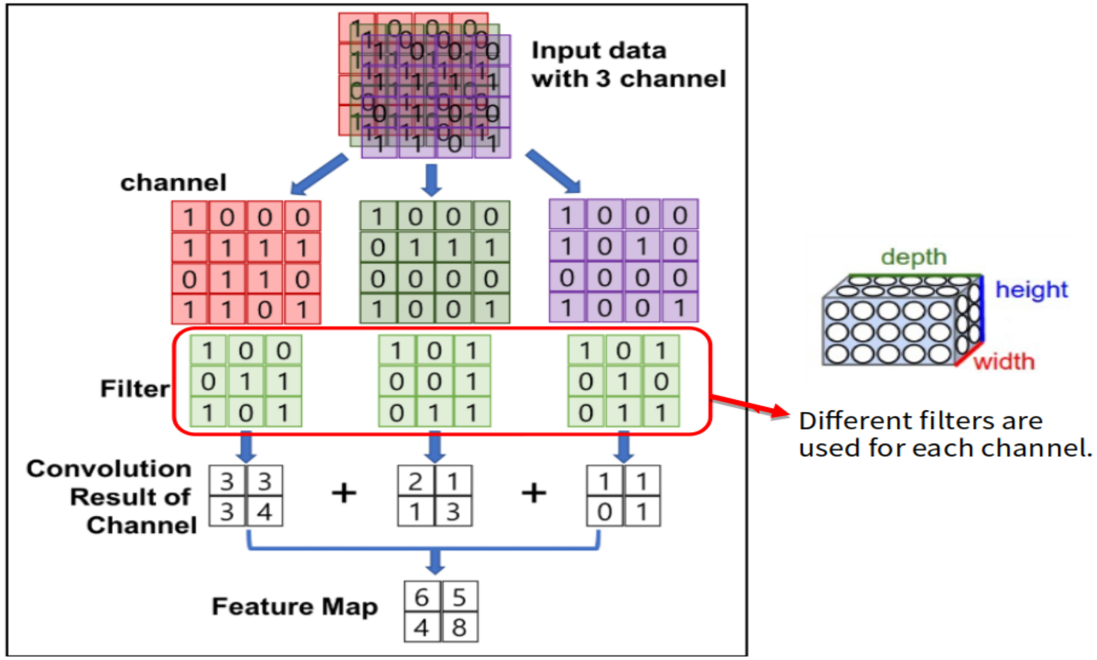


Figure 1: A Squeeze-and-Excitation block.

- Use Global Average Pooling to squeeze
- Why use global average pooling?

3.Method

- Squeeze operation



Different filters are used for each channel.

- Feature Map (=filter) cannot obtain global information because Receive Field is Local

- But we want to use global information not local information

3.Method

- Squeeze operation

- Create an H*W 's image as a Channel Descriptor by squeezing each channel

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j).$$

- Compression of Global Spatial Information of H*W size for each channel to 1*1

3.Method

▶ Excitation operation

- Obtaining the dependence of channel-specific values obtained through Squeeze operation

- Calculation of importance

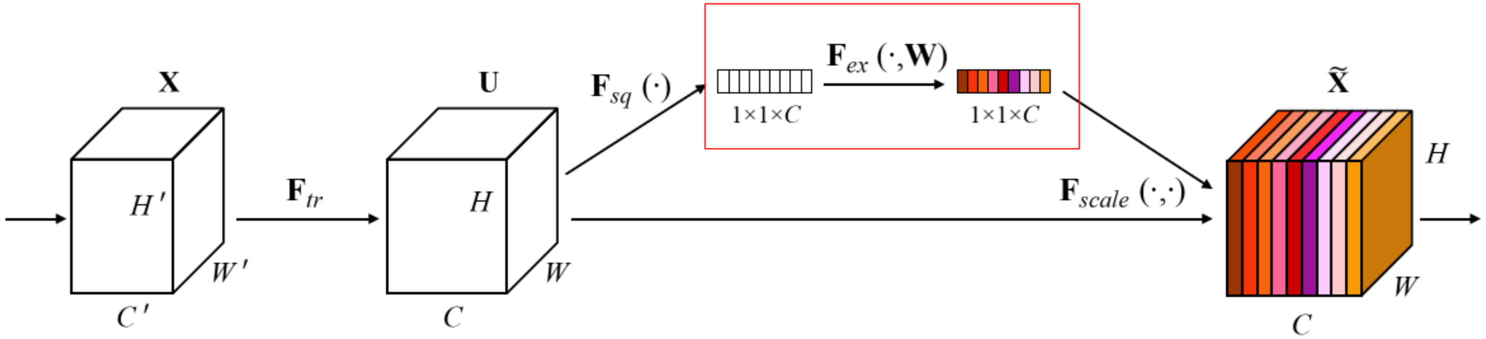


Figure 1: A Squeeze-and-Excitation block.

- how to obtain the dependence of channel-specific?

3.Method

▶ Excitation operation condition

1.Flexible

Since the relationship between channels is not simple, it should be flexible enough to cover complex relationships.(Realize Non-linearity between channels)

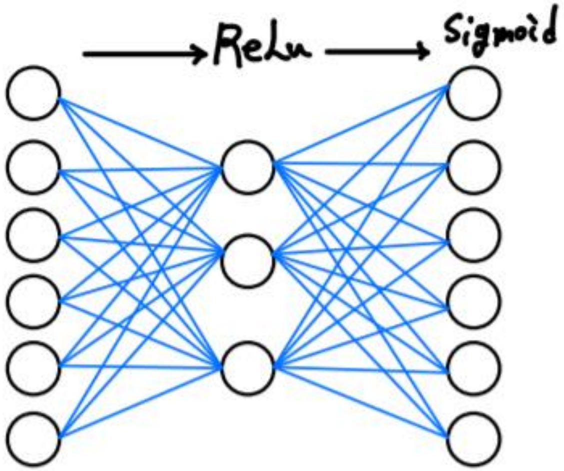
2. non-mutually-exclusive

Instead of emphasizing only one channel(one-hot activation), various channels should be emphasized

3.Method

- Excitation operation

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2\delta(\mathbf{W}_1\mathbf{z})),$$



- Fully Connected -> ReLU -> Fully Connected -> Sigmoid

3.Method

▶ Excitation operation

- In FC1, Channel compression

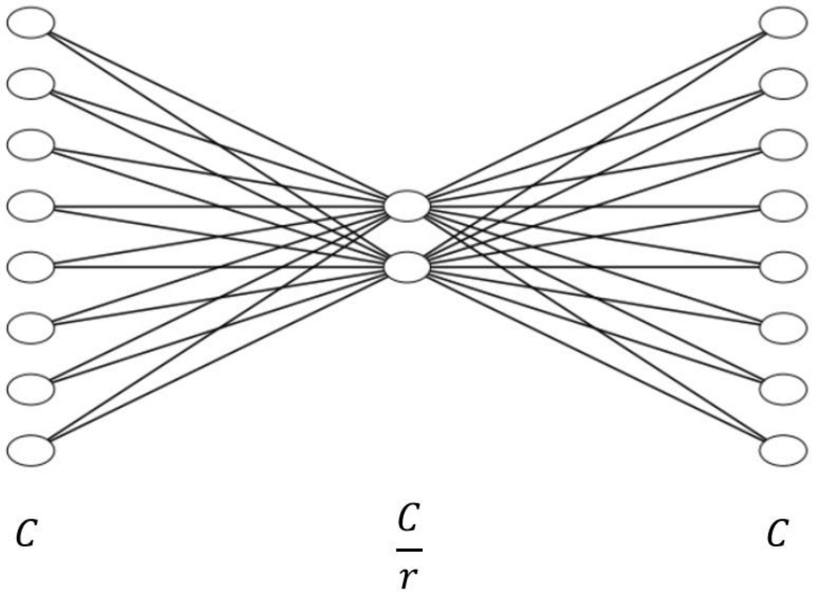
- In ReLU, implement nonlinearity

- In FC2, Channel expansion

bottleneck

- In sigmoid, Normalize to a value between 0 and 1 to express the importance of the channel

FC1 → ReLU → FC2 → Sigmoid



r : reduction ratio

3.Method

- Rescaling

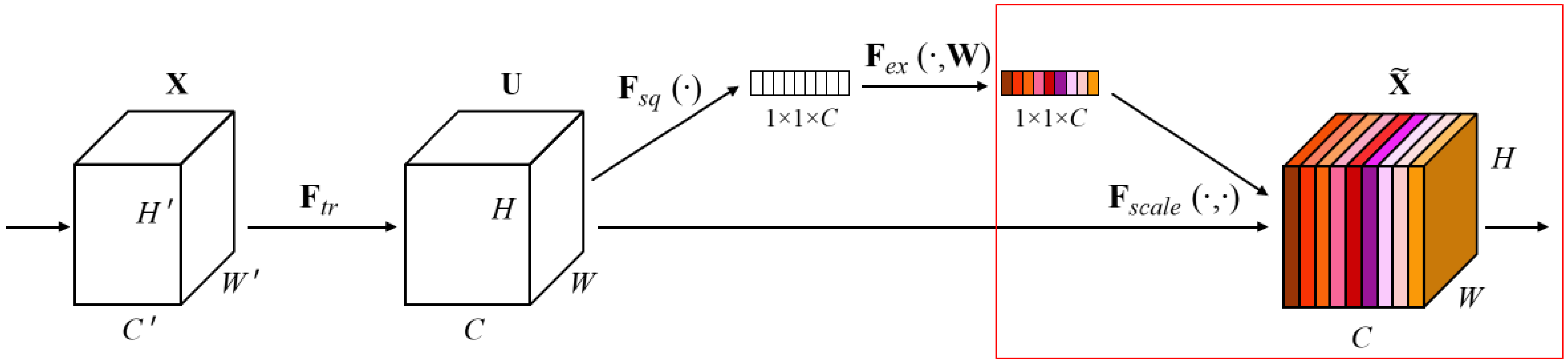
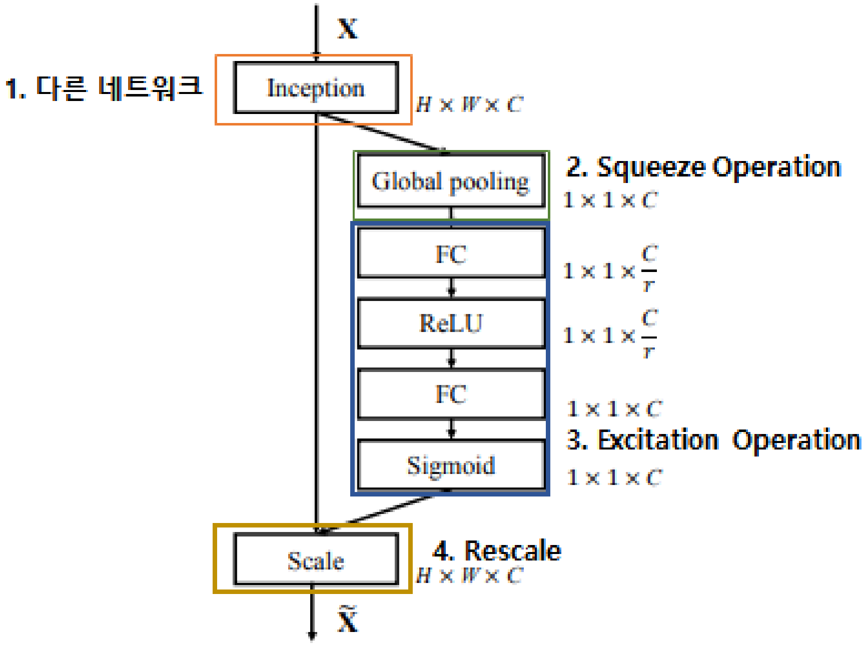


Figure 1: A Squeeze-and-Excitation block.

- self attention

3.Method

- SE block



SE-Inception Module

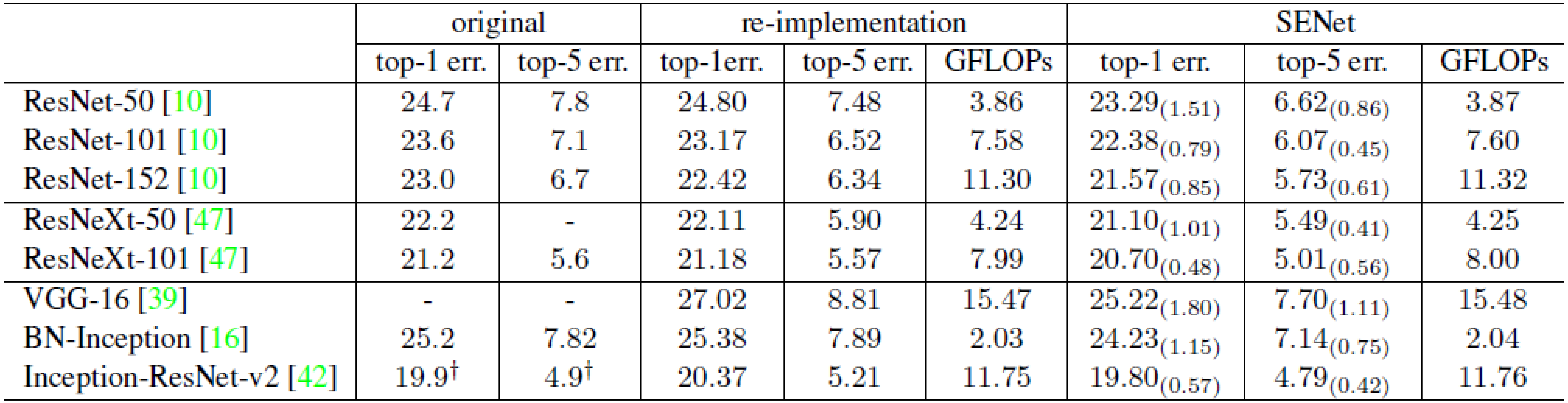
- Pros

1. It can be pasted onto an existing network

2. There are additional parameters, but the complexity does not increase that much

4.Results

- ImageNet Classification



	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [10]	24.7	7.8	24.80	7.48	3.86	$23.29_{(1.51)}$	$6.62_{(0.86)}$	3.87
ResNet-101 [10]	23.6	7.1	23.17	6.52	7.58	$22.38_{(0.79)}$	$6.07_{(0.45)}$	7.60
ResNet-152 [10]	23.0	6.7	22.42	6.34	11.30	$21.57_{(0.85)}$	$5.73_{(0.61)}$	11.32
ResNeXt-50 [47]	22.2	-	22.11	5.90	4.24	$21.10_{(1.01)}$	$5.49_{(0.41)}$	4.25
ResNeXt-101 [47]	21.2	5.6	21.18	5.57	7.99	$20.70_{(0.48)}$	$5.01_{(0.56)}$	8.00
VGG-16 [39]	-	-	27.02	8.81	15.47	$25.22_{(1.80)}$	$7.70_{(1.11)}$	15.48
BN-Inception [16]	25.2	7.82	25.38	7.89	2.03	$24.23_{(1.15)}$	$7.14_{(0.75)}$	2.04
Inception-ResNet-v2 [42]	$19.9^{\dagger}$	$4.9^{\dagger}$	20.37	5.21	11.75	$19.80_{(0.57)}$	$4.79_{(0.42)}$	11.76

GFLOPs = **G**PU **FL**oating point **O**perations **P**er **S**econd (calculation amount)

4.Results

- ▶ Scene Classification

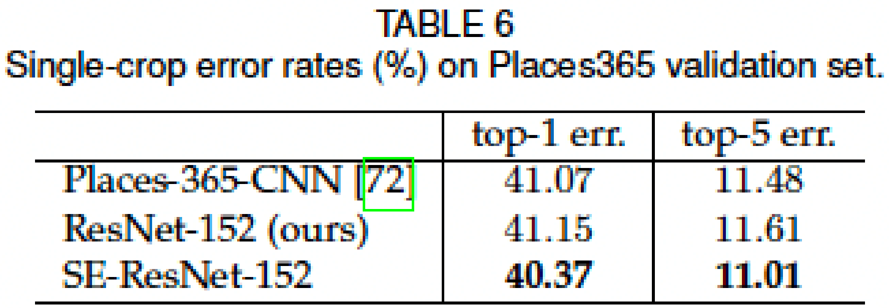


TABLE 6
Single-crop error rates (%) on Places365 validation set.

	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**

- ▶Object detection

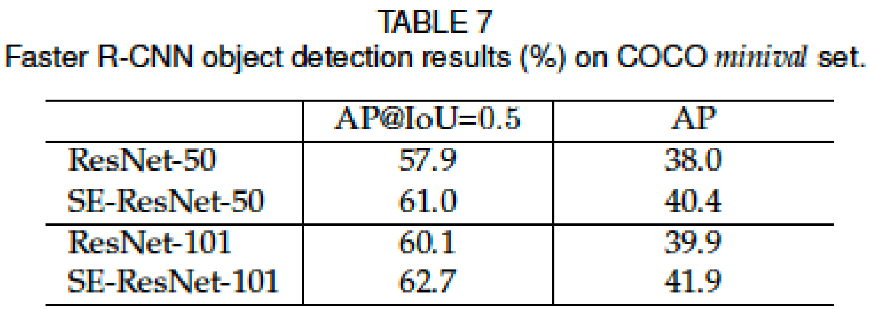
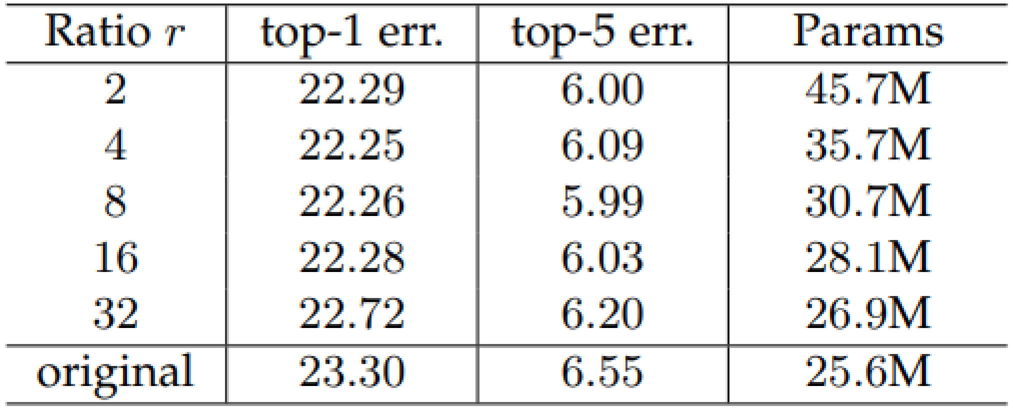


TABLE 7
Faster R-CNN object detection results (%) on COCO *minival* set.

	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

4.Results

- ▶ Reduction ratio



Ratio r	top-1 err.	top-5 err.	Params
2	22.29	6.00	45.7M
4	22.25	6.09	35.7M
8	22.26	5.99	30.7M
16	22.28	6.03	28.1M
32	22.72	6.20	26.9M
original	23.30	6.55	25.6M

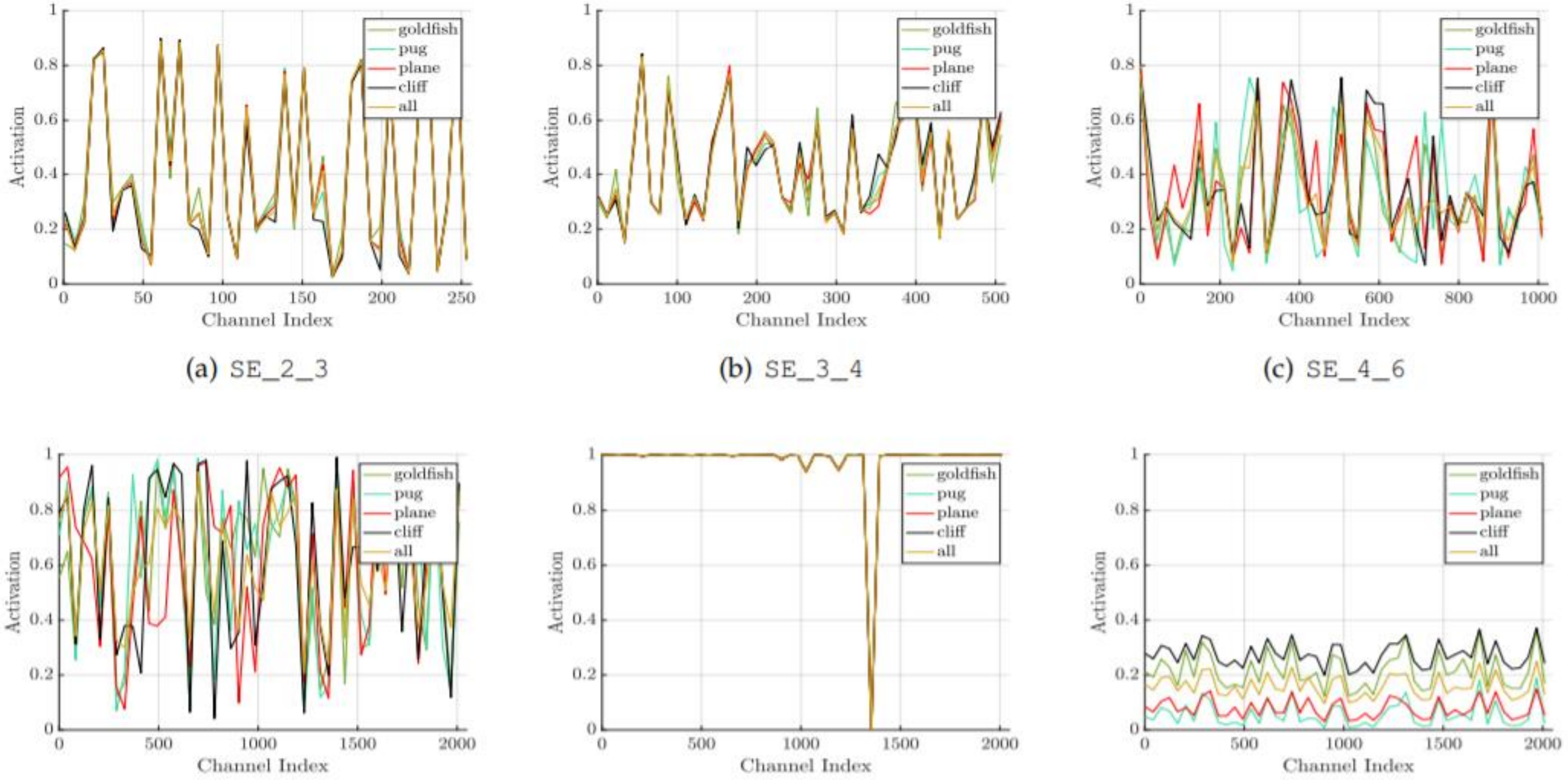
▲ When Ratio r is different

Even if (r = 16), it does not have a significant impact on the error and has a **38% parameter reduction** effect compared to (r = 2)

As a result, **r = 16** is best!!!

4.Results

- ▶ Role of Excitation



(a) SE_2_3

(b) SE_3_4

(c) SE_4_6

(d) SE_5_1

(e) SE_5_2

(f) SE_5_3

5.Conclusion

1. There was no attention function in the existing CNN, but SENet introduced the attention module for the first time

2. The SE module configured in this way has shown performance improvements in various dataset when inserted in the middle of the existing CNN model.