

Squeeze-and-Excitation Networks

Jie Hu, Li Shen, Gang Sun

CVPR 2018

조건희

Index

- Introduction
- Squeeze-and-Excitation Blocks
- Experiments
- Conclusion

Introduction

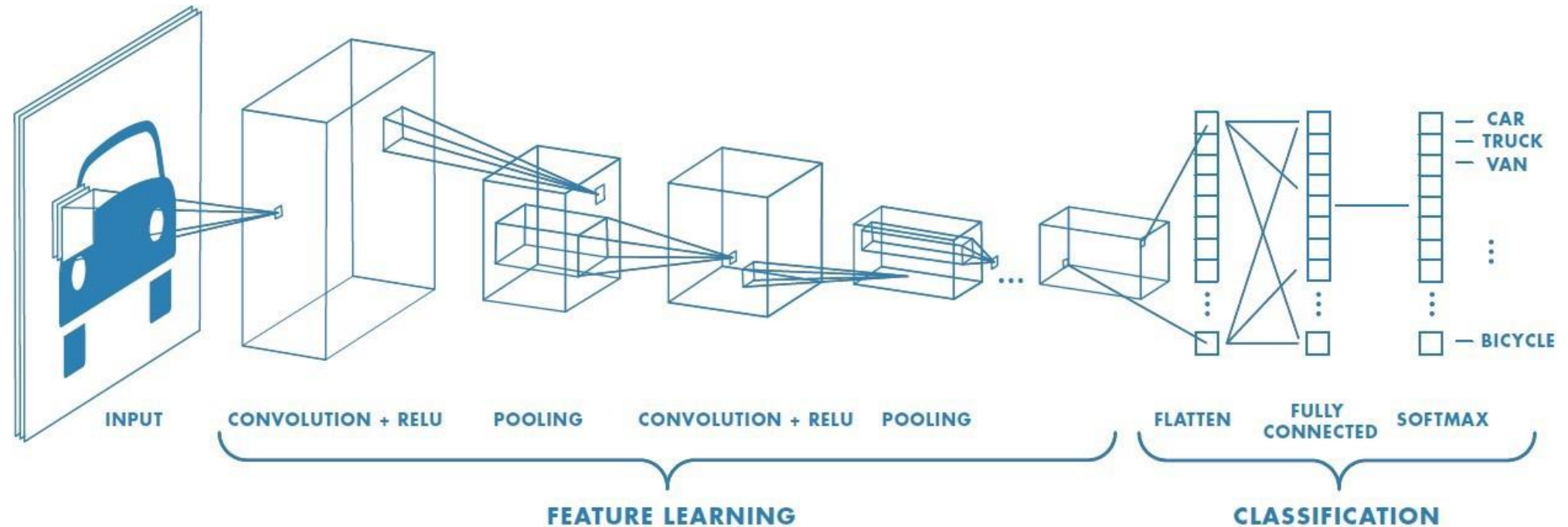
Introduction

CNN 모델의 이미지 feature 추출 성능 개선하여 2017년 ILSVRC 우승
2016년 우승한 모델의 비해 약 25% 정도 개선

Introduction

CNN 모델

각 컨볼루션 레이어에서 filter 가 local spatial connectivity 를 학습.

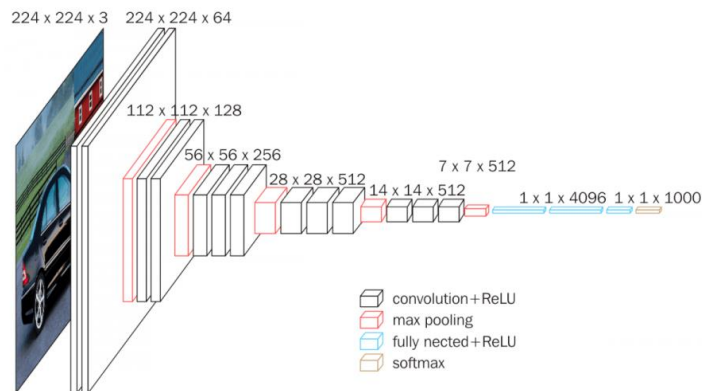


Introduction

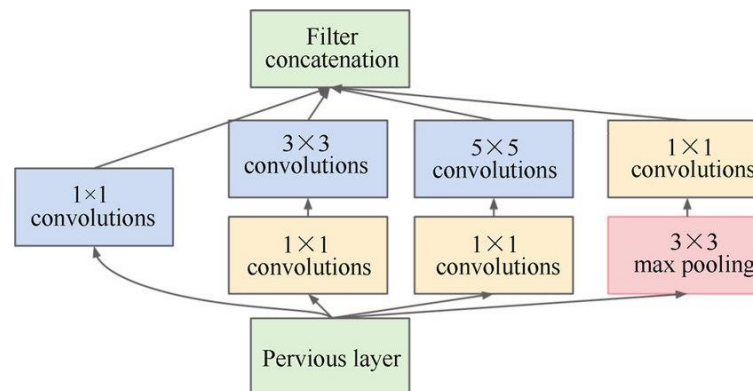
기존 연구

Spatial information 으로부터 중요한 feature를 추출하는 데에 중점을 둔 연구가 많았음.

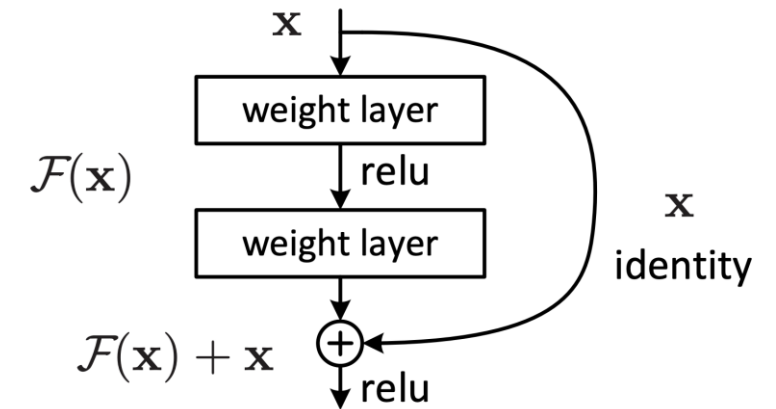
Deep architecture



VGGNet
2014



Inception
2014



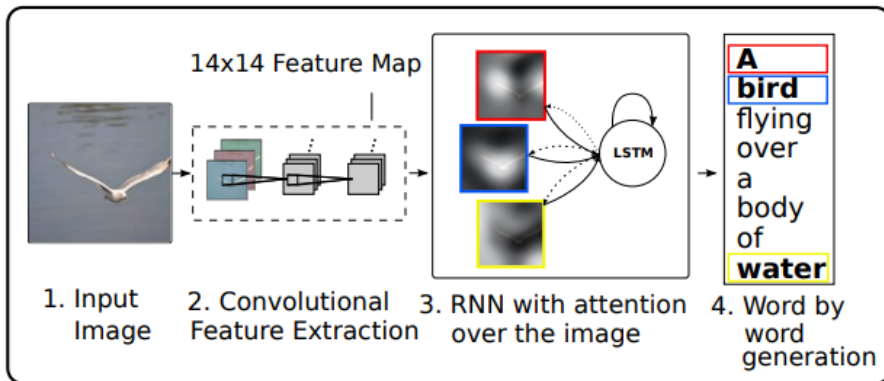
ResNet
2015

Introduction

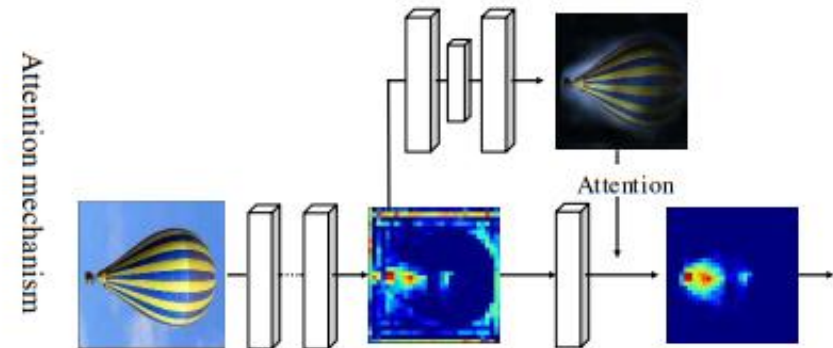
기존 연구

Spatial information 으로부터 중요한 feature를 추출하는 데에 중점을 둔 연구가 많았음.

Attention and gating mechanism



Show, Attend and Tell
2015



Residual Attention Network
2017

Introduction

이 논문에서는 다른 관점에서 접근하여 feature 추출 성능을 개선
→ Channel relationship

Introduction

“Squeeze-and-Excitation” (SE) block

“컨볼루션 필터의 채널 간 상호의존성을 explicitly 모델링함으로써, 네트워크의 표현력을 개선시키겠다.”

이를 위해서 *feature recalibration* 이라는 메커니즘을 제안함.

중요한 feature를 선택적으로 강조하고
상대적으로 덜 유용한 feature는 억제하기 위해서
Global information 을 사용하는 메커니즘

Squeeze-and-Excitation Blocks

Squeeze-and-Excitation Blocks

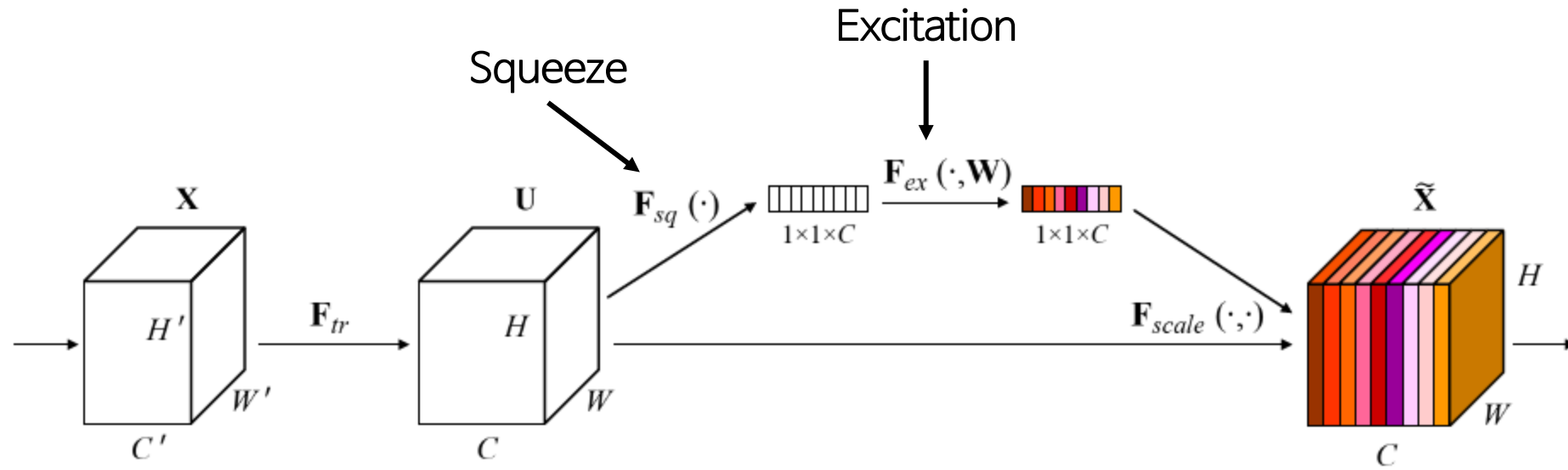


Figure 1: A Squeeze-and-Excitation block.

Squeeze-and-Excitation Blocks

각 채널은 local receptive field(=filter) 로 학습 되므로
U 의 각 채널은 필터가 지나가는 지역 이외의 contextual 정보는 이용할 수 없다.

→ Global Information embedding

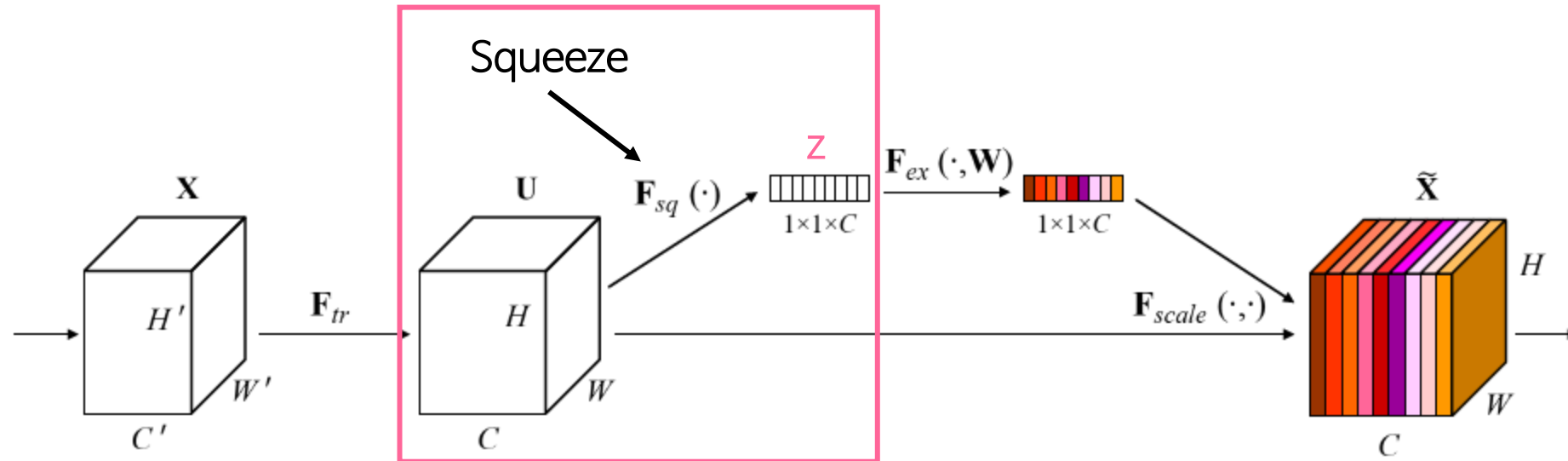


Figure 1: A Squeeze-and-Excitation block.

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

Global average pooling

Squeeze-and-Excitation Blocks

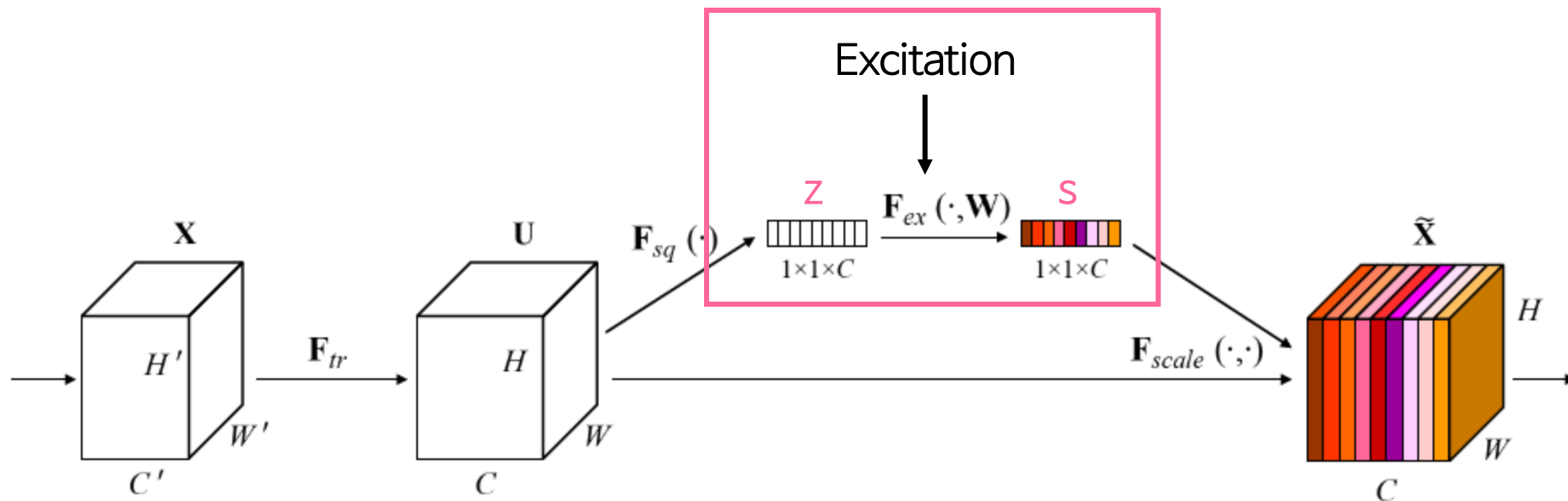


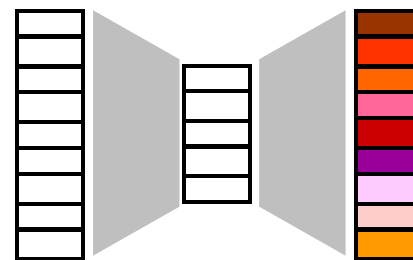
Figure 1: A Squeeze-and-Excitation block.

Channel-wise dependency 를 학습하는 함수가 필요

(아래 2가지 기준을 충족하는 함수)

- 1) 채널간 비선형적 관계를 학습해야 함
- 2) 채널 간 상호 배타적이지 않은 관계를 학습해야 함.

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)),$$



Squeeze-and-Excitation Blocks

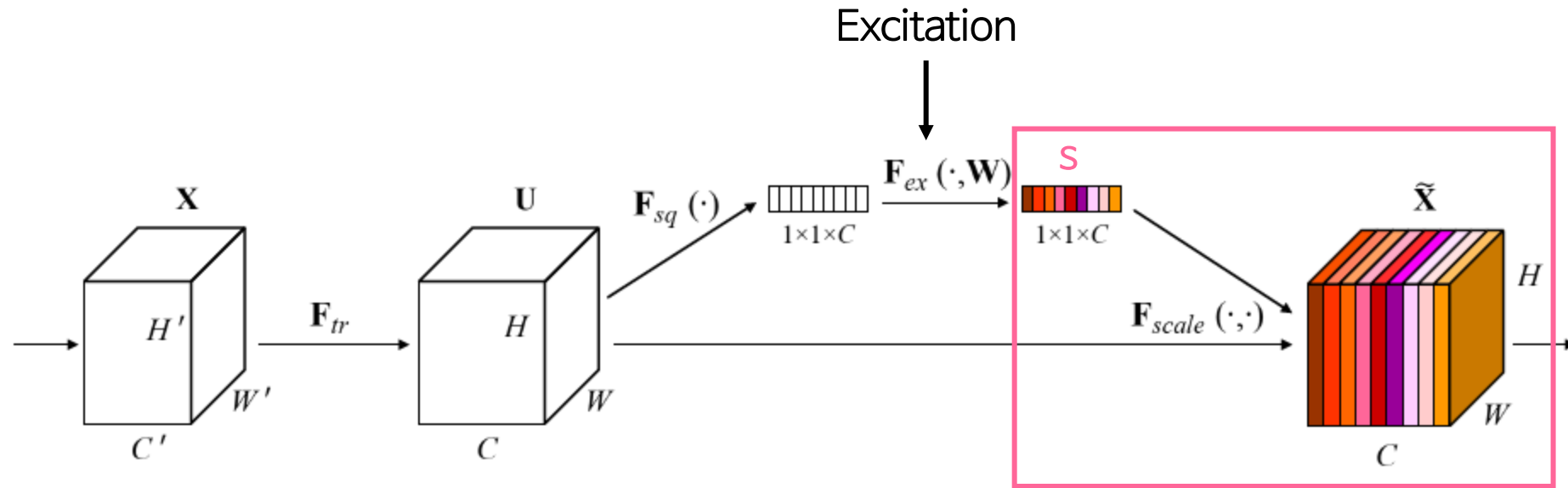
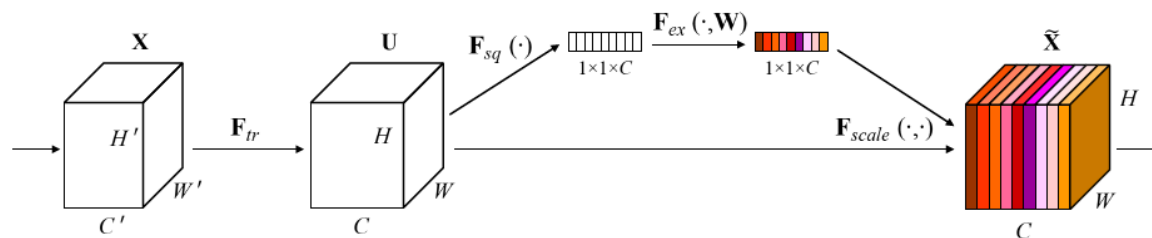


Figure 1: A Squeeze-and-Excitation block.

$$\tilde{\mathbf{x}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \cdot \mathbf{u}_c,$$

Squeeze-and-Excitation Blocks



```
def Squeeze_excitation_layer(self, input_x, out_dim, ratio, layer_name):
    with tf.name_scope(layer_name) :
        squeeze = Global_Average_Pooling(input_x)

        excitation = Fully_connected(squeeze, units=out_dim / ratio, layer_name=layer_name+'_fully_connected1')
        excitation = Relu(excitation)
        excitation = Fully_connected(excitation, units=out_dim, layer_name=layer_name+'_fully_connected2')
        excitation = Sigmoid(excitation)

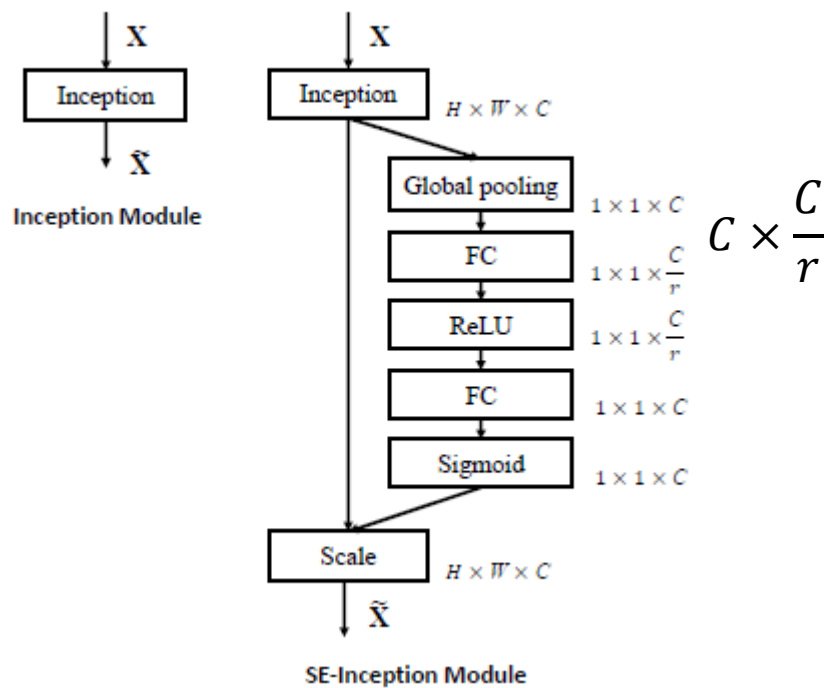
        excitation = tf.reshape(excitation, [-1,1,1,out_dim])

        scale = input_x * excitation

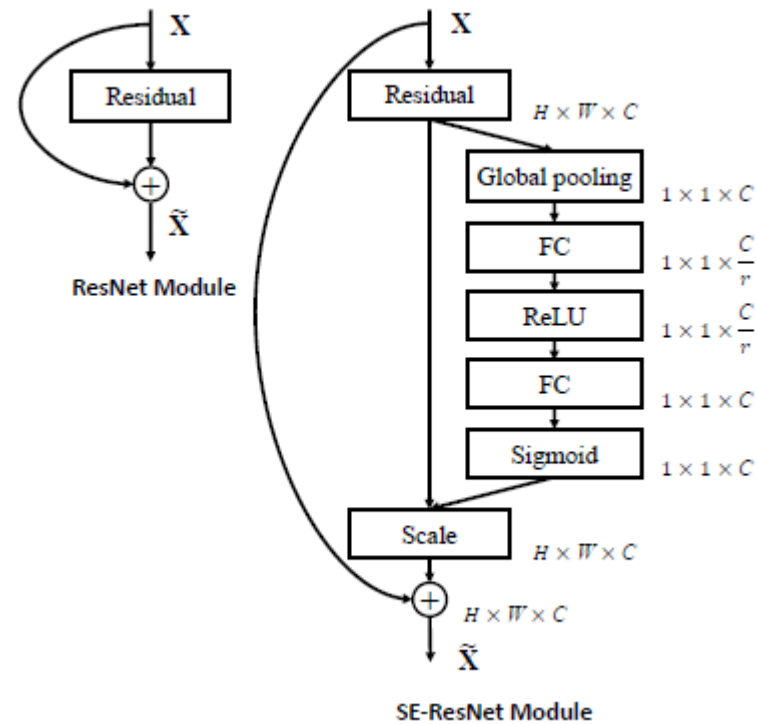
    return scale
```

Squeeze-and-Excitation Blocks

SE-Inception



SE-ResNet



Squeeze-and-Excitation Blocks

Computational Complexity

$$\frac{2}{r} \sum_{s=1}^S N_s \cdot C_s^2$$

ResNet-50 : 약 3.86 GFLOPs

SE-ResNet-50 : 약 3.87 GFLOPs

→ 계산량 증가 : 0.26%

Experiments

Experiments

ImageNet Classification

	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 [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 [†]	4.9 [†]	20.37	5.21	11.75	19.80 _(0.57)	4.79 _(0.42)	11.76

Table 2: Single-crop error rates (%) on the ImageNet validation set and complexity comparisons. The *original* column refers to the results reported in the original papers. To enable a fair comparison, we re-train the baseline models and report the scores in the *re-implementation* column. The *SENet* column refers to the corresponding architectures in which SE blocks have been added. The numbers in brackets denote the performance improvement over the re-implemented baselines. † indicates that the model has been evaluated on the non-blacklisted subset of the validation set (this is discussed in more detail in [42]), which may slightly improve results. VGG-16 and SE-VGG-16 are trained with batch normalization.

다양한 모델에 SE block 적용해봤을 때 모두 성능 증가

Experiments

ImageNet Classification

	original		re-implementation				SENet			
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	MFLOPs	Million Parameters	top-1 err.	top-5 err.	MFLOPs	Million Parameters
MobileNet [13]	29.4	-	29.1	10.1	569	4.2	25.3 _(3.8)	7.9 _(2.2)	572	4.7
ShuffleNet [52]	34.1	-	33.9	13.6	140	1.8	31.7 _(2.2)	11.7 _(1.9)	142	2.4

Table 3: Single-crop error rates (%) on the ImageNet validation set and complexity comparisons. Here, MobileNet refers to “1.0 MobileNet-224” in [13] and ShuffleNet refers to “ShuffleNet $1 \times (g = 3)$ ” in [52].

파라미터와 계산량이 적은 모델에 대해서도 실험 → 성능 증가

Experiments

ImageNet Classification

	224 × 224		320 × 320 / 299 × 299	
	top-1 err.	top-5 err.	top-1 err.	top-5 err.
ResNet-152 [10]	23.0	6.7	21.3	5.5
ResNet-200 [11]	21.7	5.8	20.1	4.8
Inception-v3 [44]	-	-	21.2	5.6
Inception-v4 [42]	-	-	20.0	5.0
Inception-ResNet-v2 [42]	-	-	19.9	4.9
ResNeXt-101 (64 × 4d) [47]	20.4	5.3	19.1	4.4
DenseNet-264 [14]	22.15	6.12	-	-
Attention-92 [46]	-	-	19.5	4.8
Very Deep PolyNet [51] †	-	-	18.71	4.25
PyramidNet-200 [8]	20.1	5.4	19.2	4.7
DPN-131 [5]	19.93	5.12	18.55	4.16
SENet-154	18.68	4.47	17.28	3.79
NASNet-A (6@4032) [55] †	-	-	17.3‡	3.8‡
SENet-154 (post-challenge)	-	-	16.88‡	3.58‡

Table 4: Single-crop error rates of state-of-the-art CNNs on ImageNet validation set. The size of test crop is 224×224 and $320 \times 320 / 299 \times 299$ as in [11]. † denotes the model with a larger crop 331×331 . ‡ denotes the post-challenge result. SENet-154 (post-challenge) is trained with a larger input size 320×320 compared to the original one with the input size 224×224 .

최신 모델에 대해서 실험 → 성능 증가

Experiments

Scene Classification

Places365-Challenge dataset

	top-1 err.	top-5 err.
Places-365-CNN [37]	41.07	11.48
ResNet-152 (ours)	41.15	11.61
SE-ResNet-152	40.37	11.01

Table 5: Single-crop error rates (%) on Places365 validation set.

이미지넷 말고 다른 데이터셋에 대하여 실험 → 성능 증가

Experiments

Object Detection

COCO dataset

	AP@IoU=0.5	AP
ResNet-50	45.2	25.1
SE-ResNet-50	46.8	26.4
ResNet-101	48.4	27.2
SE-ResNet-101	49.2	27.9

Table 6: Object detection results on the COCO 40k validation set by using the basic Faster R-CNN.

Classification 말고 Object detection에 적용 → 성능 증가

Experiments

파라미터 증가량에 따른 tradeoff

Ratio r	top-1 err.	top-5 err.	Million Parameters
4	23.21	6.63	35.7
8	23.19	6.64	30.7
16	23.29	6.62	28.1
32	23.40	6.77	26.9
original	24.80	7.48	25.6

Table 7: Single-crop error rates (%) on ImageNet validation set and parameter sizes for SE-ResNet-50 at different reduction ratios r . Here *original* refers to ResNet-50.

r 값이 크면 클수록 파라미터 수는 적어지지만 성능은 떨어짐

$r = 16$ 이 good tradeoff라서 모든 실험에 사용함

Experiments

Excitation 의 역할에 대한 분석

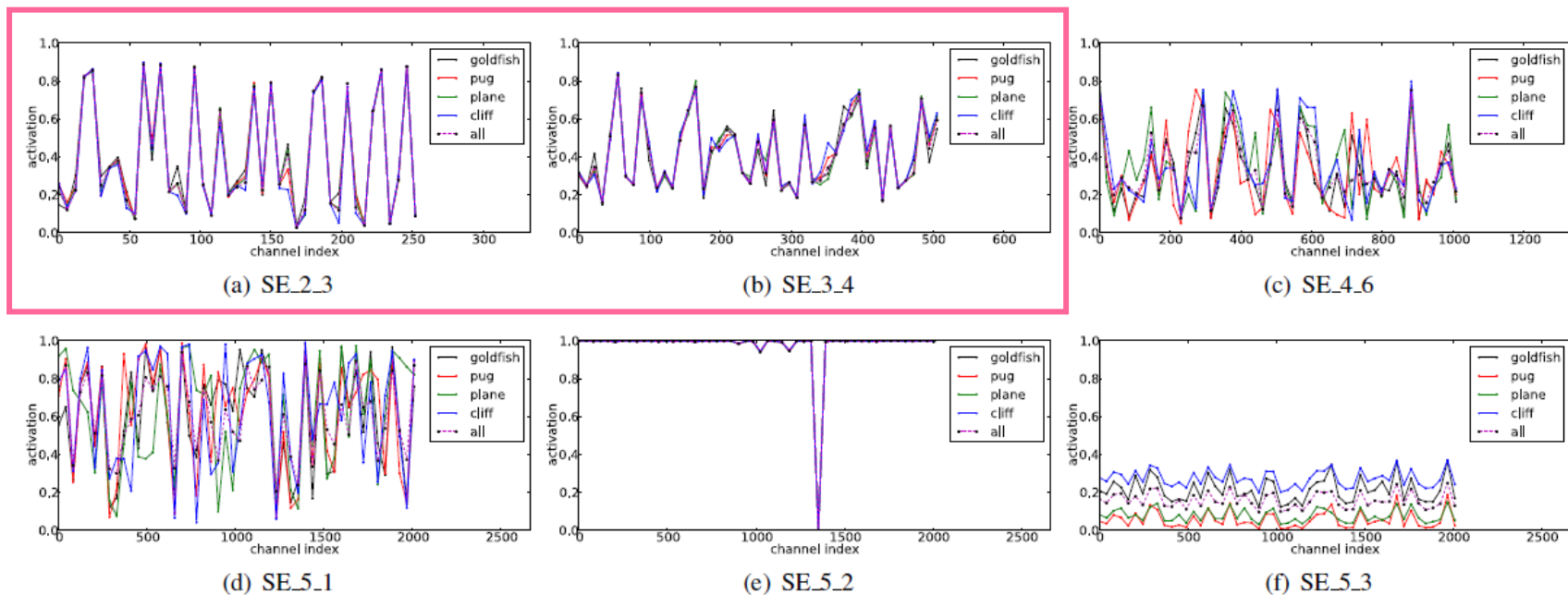


Figure 5: Activations induced by *Excitation* in the different modules of SE-ResNet-50 on ImageNet. The module is named as “SE_stageID_blockID”.

1) 초반 레이어에서는 클래스가 달라도 channel activation 이 거의 동일

Experiments

Excitation 의 역할에 대한 분석

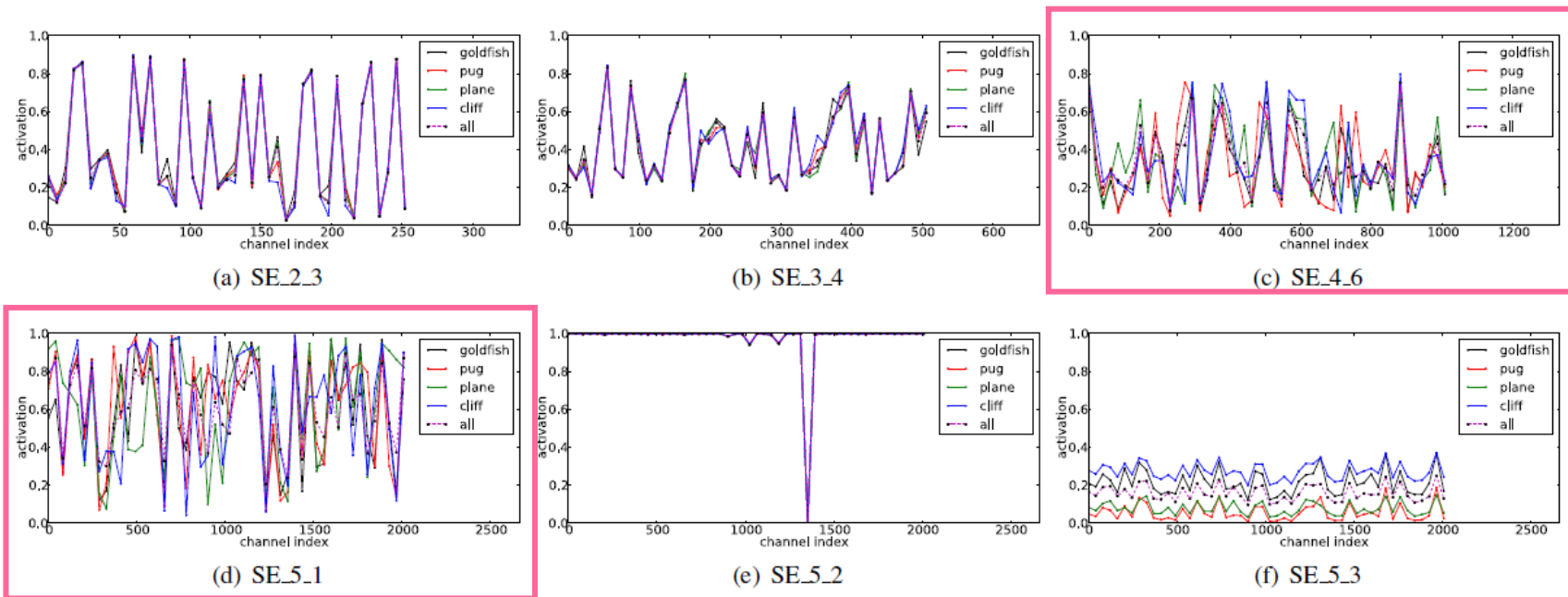


Figure 5: Activations induced by *Excitation* in the different modules of SE-ResNet-50 on ImageNet. The module is named as “SE_stageID_blockID”.

2) Depth 가 깊어질수록 class-specific 하게 변함

Experiments

Excitation 의 역할에 대한 분석

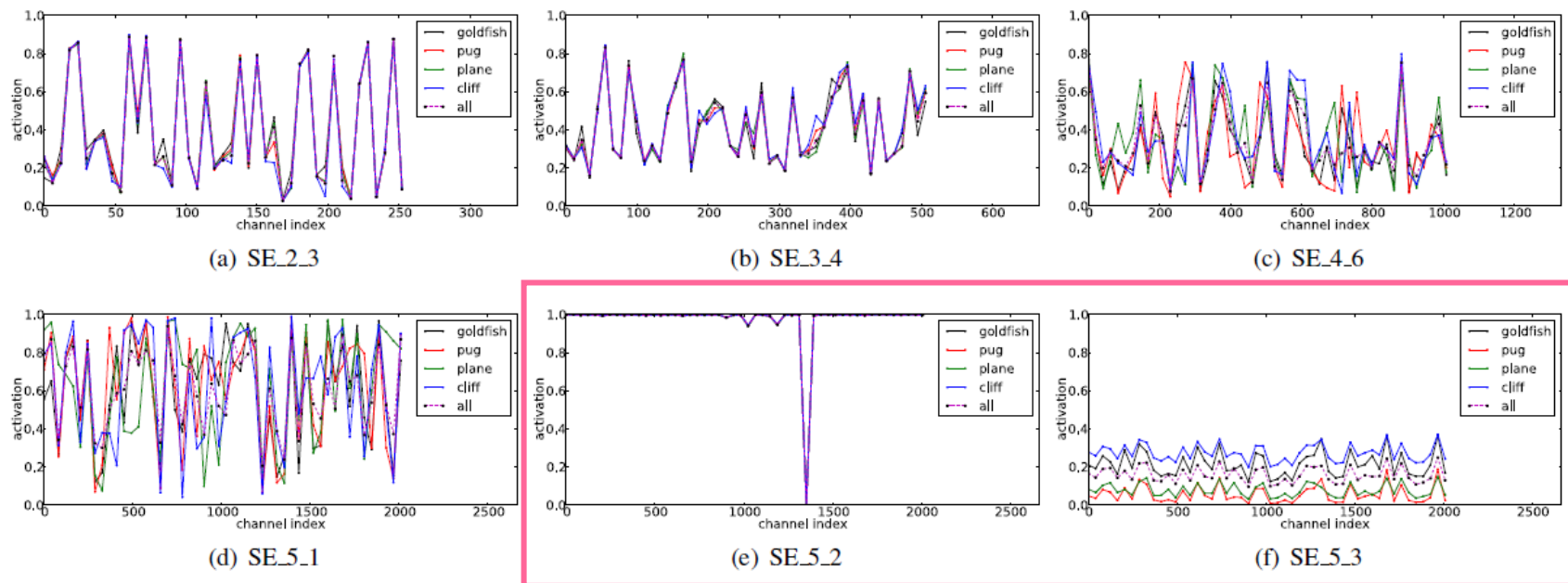


Figure 5: Activations induced by *Excitation* in the different modules of SE-ResNet-50 on ImageNet. The module is named as “SE_stageID_blockID”.

3) 마지막에서는 뭔가 다름

[1 또는 0] / [클래스 간 별 차이 없음] → less important than previous blocks

Experiments

Excitation 의 역할에 대한 분석

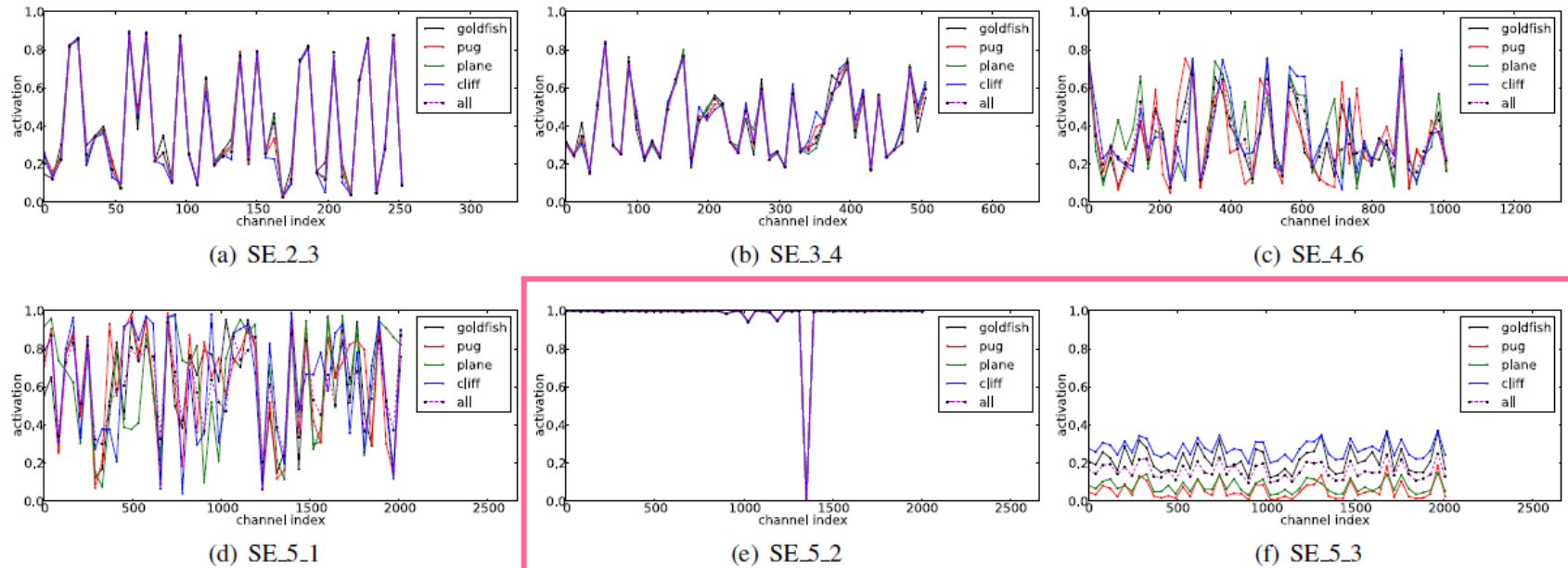


Figure 5: Activations induced by *Excitation* in the different modules of SE-ResNet-50 on ImageNet. The module is named as “SE_stageID_blockID”.

$$\frac{2}{r} \sum_{s=1}^S N_s \cdot C_s^2$$

Conclusion

Conclusion

- 1) 이 논문에서는 SE block 구조를 제안
(Channel-wise feature recalibration 을 수행하여 CNN 기반 모델의 성능 개선)
- 2) 실험을 통해 다양한 모델에 적용 가능하다는 것을 보여줌
- 3) 채널 별 feature 의존성을 모델링하여 기존 구조의 한계에 대한 통찰력을 제공함
- 4) SE 블록에 의해 feature 중요성을 파악할 수 있어서 네트워크 프루닝 분야에 도움이 될 수 있을 것임

Q & A

Thank you