

# Learning Deep Features for Discriminative Localization

**Junhong Kim**

**Korea University**

**{Junhongkim@korea.ac.kr}**

**DS·BA**

Data science & Business analytics

# Convolution neural network applications

- 단순 classification도 할 수 있지만 convolution neural network 를 통하여 실제 이미지에서 많은 방향으로 접근 해 볼 수 있음

**Semantic Segmentation**



GRASS, CAT,  
TREE, SKY

No objects, just pixels

**Classification + Localization**



CAT

Single Object

**Object Detection**



DOG, DOG, CAT

Multiple Object

**Instance Segmentation**



DOG, DOG, CAT

This image is CC0 public domain

Fei-Fei Li & Justin Johnson & Serena Yeung

Lecture 11 - 17 May 10, 2017

# Paper highlights

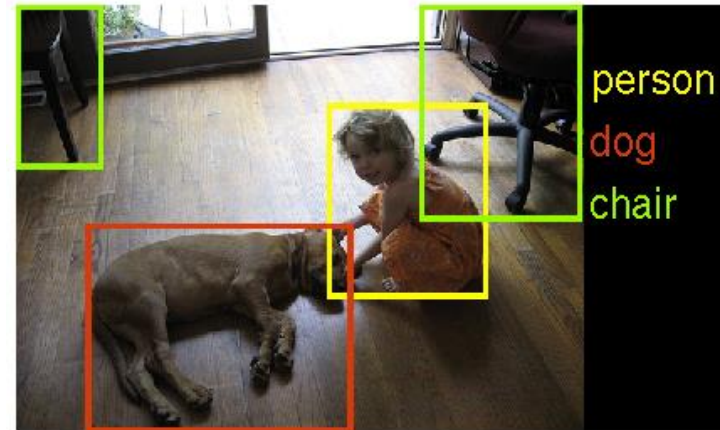
- Label만 있는 상황에서 classification을 하였을 때 Object detection처럼 Semantic segmentation처럼 각 class에 대하여 spatial정보를 추출 할 수 있을까?



(a) Image

(b) G.T.

Example of semantic segmentation dataset



Example of semantic object detection dataset



# Paper highlights

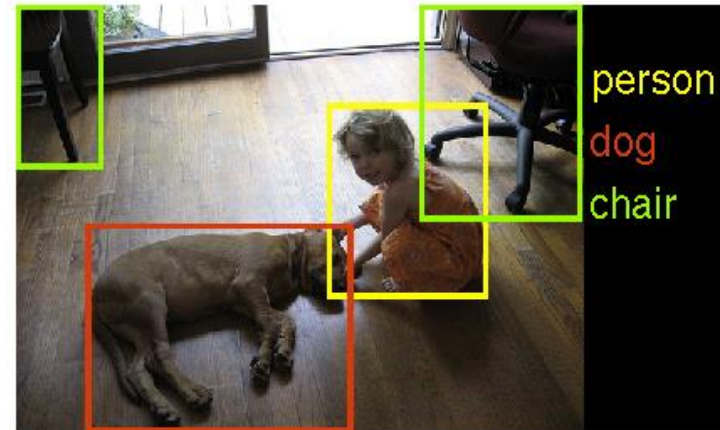
- Label만 있는 상황에서 classification을 하였을 때 Object detection처럼 Semantic segmentation처럼 각 class에 대하여 spatial정보를 추출 할 수 없을까? → CNN의 특성을 이용해서 할 수 있음!



(a) Image

(b) G.T.

Example of semantic segmentation dataset



Example of semantic object detection dataset

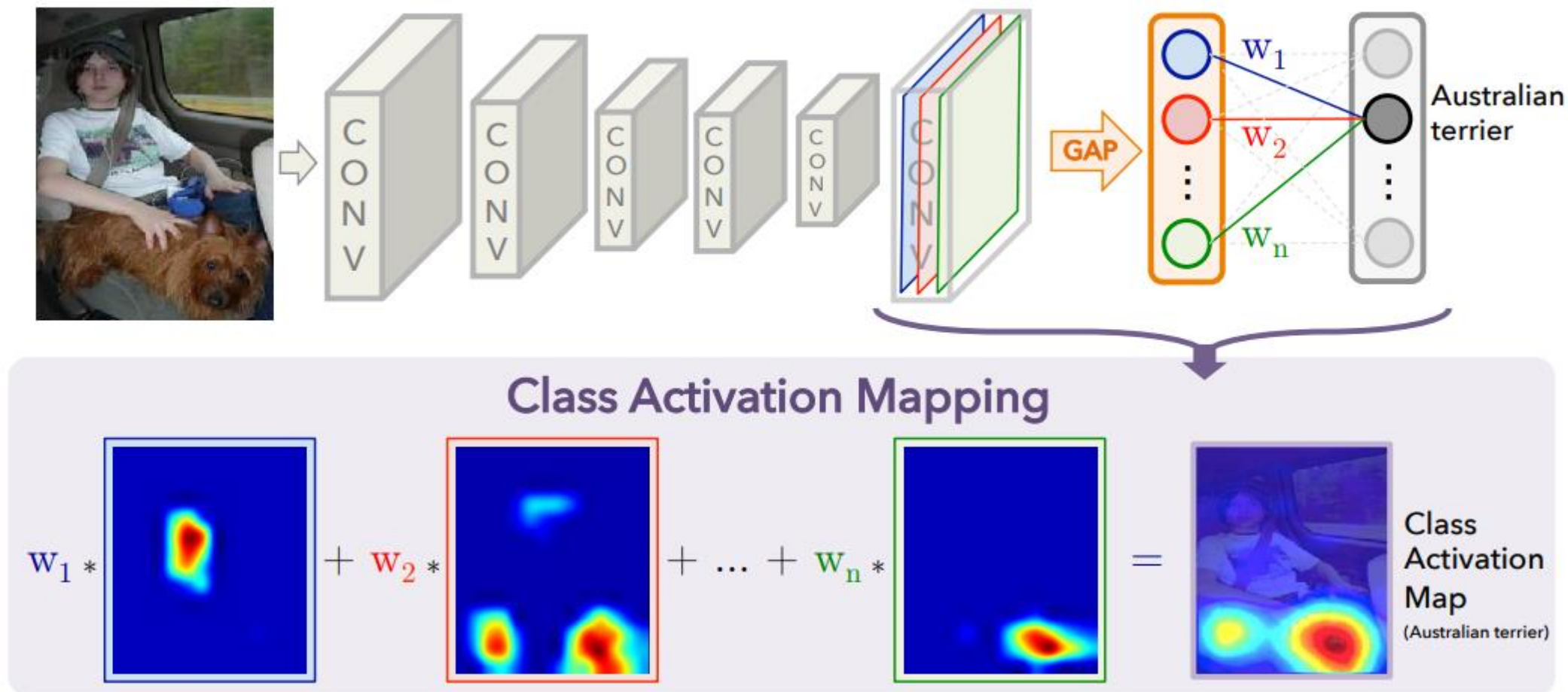
## Class activation map

- What the CNN is looking and how it shifts the attention in the video

<https://youtu.be/fZvOy0VXWAI>

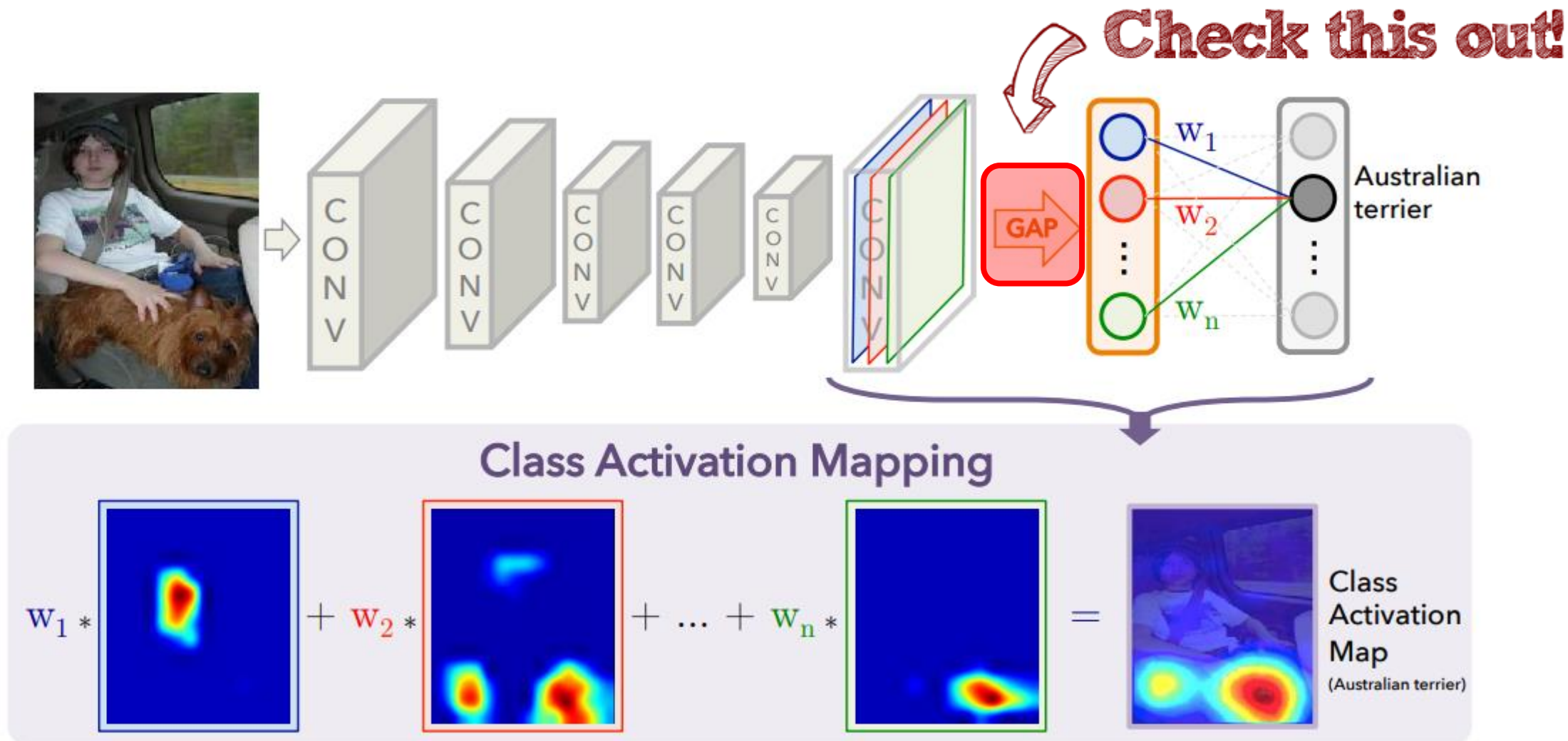
# Class activation map

- Definition of Class activation map



# Class activation map

- Definition of Class activation map

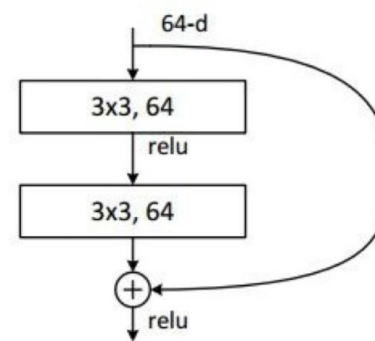




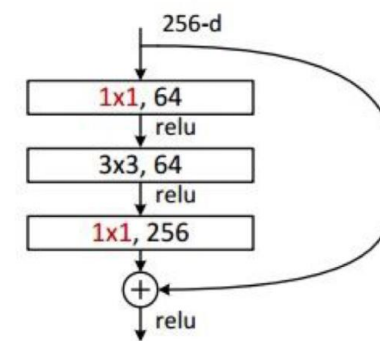
# Class activation map

## ● Global Average Pooling in Resnet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$



all-3x3



**bottleneck**  
(for ResNet-50/101/152)

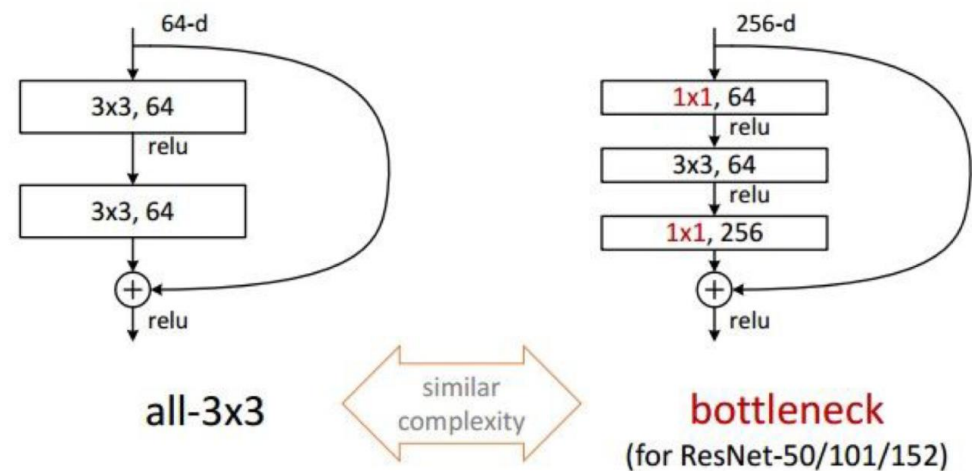


# Class activation map

- Global Average Pooling in Resnet

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

**GAP**



# Class activation map

## ● Global Average Pooling in DenseNet

Layers	Output Size	DenseNet-121( $k = 32$ )	DenseNet-169( $k = 32$ )	DenseNet-201( $k = 32$ )	DenseNet-161( $k = 48$ )
Convolution	$112 \times 112$	$7 \times 7$ conv, stride 2			
Pooling	$56 \times 56$	$3 \times 3$ max pool, stride 2			
Dense Block (1)	$56 \times 56$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	$56 \times 56$	$1 \times 1$ conv			
	$28 \times 28$	$2 \times 2$ average pool, stride 2			
Dense Block (2)	$28 \times 28$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	$28 \times 28$	$1 \times 1$ conv			
	$14 \times 14$	$2 \times 2$ average pool, stride 2			
Dense Block (3)	$14 \times 14$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	$14 \times 14$	$1 \times 1$ conv			
	$7 \times 7$	$2 \times 2$ average pool, stride 2			
Dense Block (4)	$7 \times 7$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	$1 \times 1$	$7 \times 7$ global average pool			
		1000D fully-connected, softmax			

**GAP**

**Table 1:** DenseNet architectures for ImageNet. The growth rate for the first 3 networks is  $k = 32$ , and  $k = 48$  for DenseNet-161. Note that each “conv” layer shown in the table corresponds the sequence BN-ReLU-Conv.

# Class activation map

- CAM (Global Maximum Pooling == > Global Average Pooling)

Oquab, Maxime, et al. "Is object localization for free?-weakly-supervised learning with convolutional neural networks."  
*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

## Is object localization for free? – Weakly-supervised learning with convolutional neural networks

Maxime Oquab\*  
INRIA Paris, France

Léon Bottou†  
MSR, New York, USA

Ivan Laptev\*  
INRIA, Paris, France

Josef Sivic\*  
INRIA, Paris, France

CVPR  
2015

GMP

Zhou, Bolei, et al. "Learning deep features for discriminative localization."  
*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

## Learning Deep Features for Discriminative Localization

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba  
Computer Science and Artificial Intelligence Laboratory, MIT  
{bzhou, khosla, agata, oliva, torralba}@csail.mit.edu

CVPR  
2016

GAP

# Class activation map

- CAM (Global Maximum Pooling == > Global Average Pooling)

Oquab, Maxime, et al. "Is object localization for free?-weakly-supervised learning with convolutional neural networks."  
*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

## Is object localization for free? – Weakly-supervised learning with convolutional neural networks

Maxime Oquab\*  
INRIA Paris, France

Léon Bottou†  
MSR, New York, USA

Ivan Laptev\*  
INRIA, Paris, France

Josef Sivic\*  
INRIA, Paris, France

Check this out

CVPR  
2015

Check this out

GMP

Zhou, Bolei, et al. "Learning deep features for discriminative localization."  
*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

## Learning Deep Features for Discriminative Localization

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba  
Computer Science and Artificial Intelligence Laboratory, MIT  
{bzhou, khosla, agata, oliva, torralba}@csail.mit.edu

Check this out

CVPR  
2016

Check this out

GAP



# Class activation map

- CAM (Global Maximum Pooling == > Global Average Pooling)



# Class activation map

- CAM (Global Maximum Pooling == > Global Average Pooling)



Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

**Global average pooling (GAP) vs global max pooling (GMP):** Given the prior work [16] on using GMP for weakly supervised object localization, we believe it is important to highlight the intuitive difference between GAP and GMP. We believe that GAP loss encourages the net-

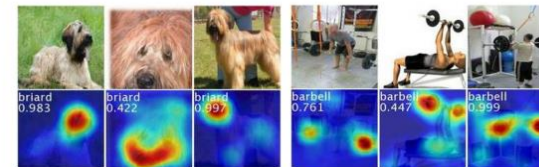


Figure 3. The CAMs of two classes from ILSVRC [21]. The maps highlight the discriminative image regions used for image classification, the head of the animal for *briard* and the plates in *barbell*.

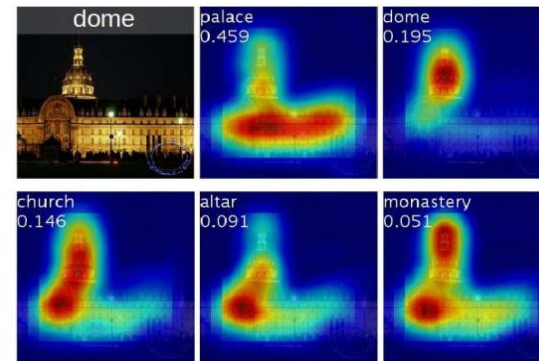


Figure 4. Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome. The predicted class and its score are shown above each class activation map. We observe that the highlighted regions vary across predicted classes e.g., *dome* activates the upper round part while *palace* activates the lower flat part of the compound.

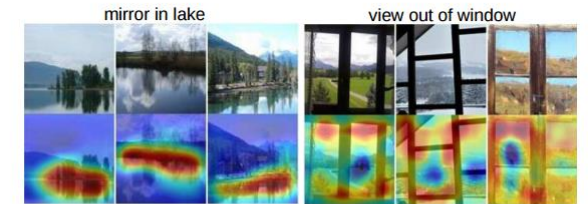


Figure 10. Informative regions for the concept learned from weakly labeled images. Despite being fairly abstract, the concepts are adequately localized by our GoogLeNet-GAP network.



Figure 11. Learning a weakly supervised text detector. The text is accurately detected on the image even though our network is not trained with text or any bounding box annotations.



Figure 12. Examples of highlighted image regions for the predicted answer class in the visual question answering.



# Class activation map

- CAM (Global Maximum Pooling == > Global Average Pooling)

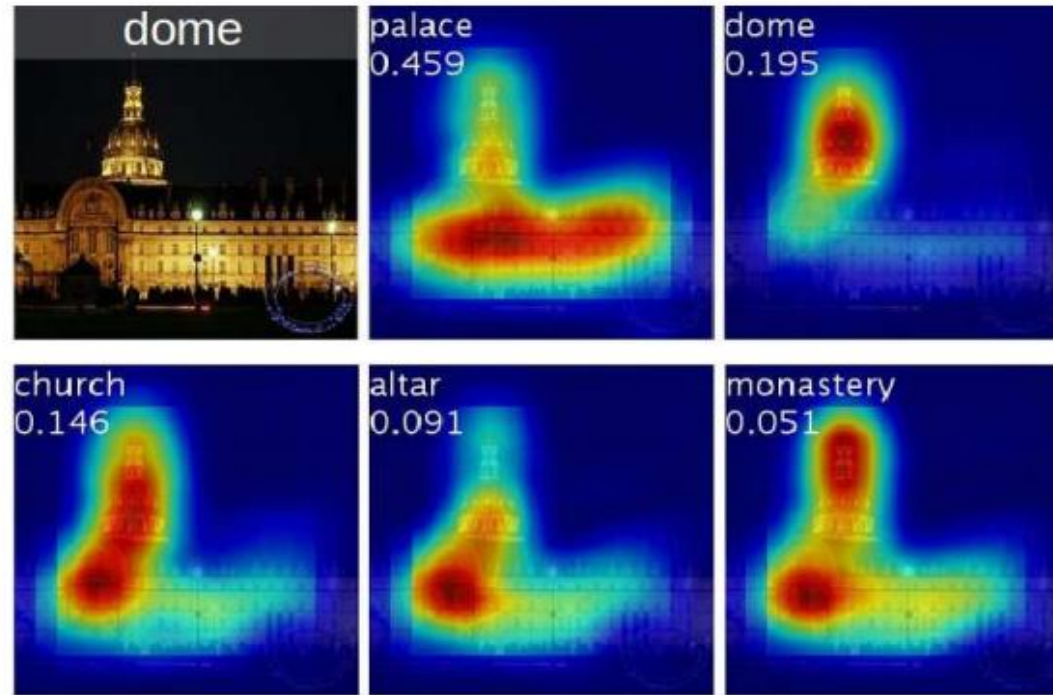


Figure 4. Examples of the CAMs generated from the top 5 predicted categories for the given image with ground-truth as dome. The predicted class and its score are shown above each class activation map. We observe that the highlighted regions vary across predicted classes e.g., dome activates the upper round part while palace activates the lower flat part of the compound.



# Class activation map

- CAM (Global Maximum Pooling == > Global Average Pooling)



Figure 11. Learning a weakly supervised text detector. The text is accurately detected on the image even though our network is not trained with text or any bounding box annotations.



# Class activation map

## Example! (Sine vs Noise Sine)

**Junhong Kim**

**Korea University**

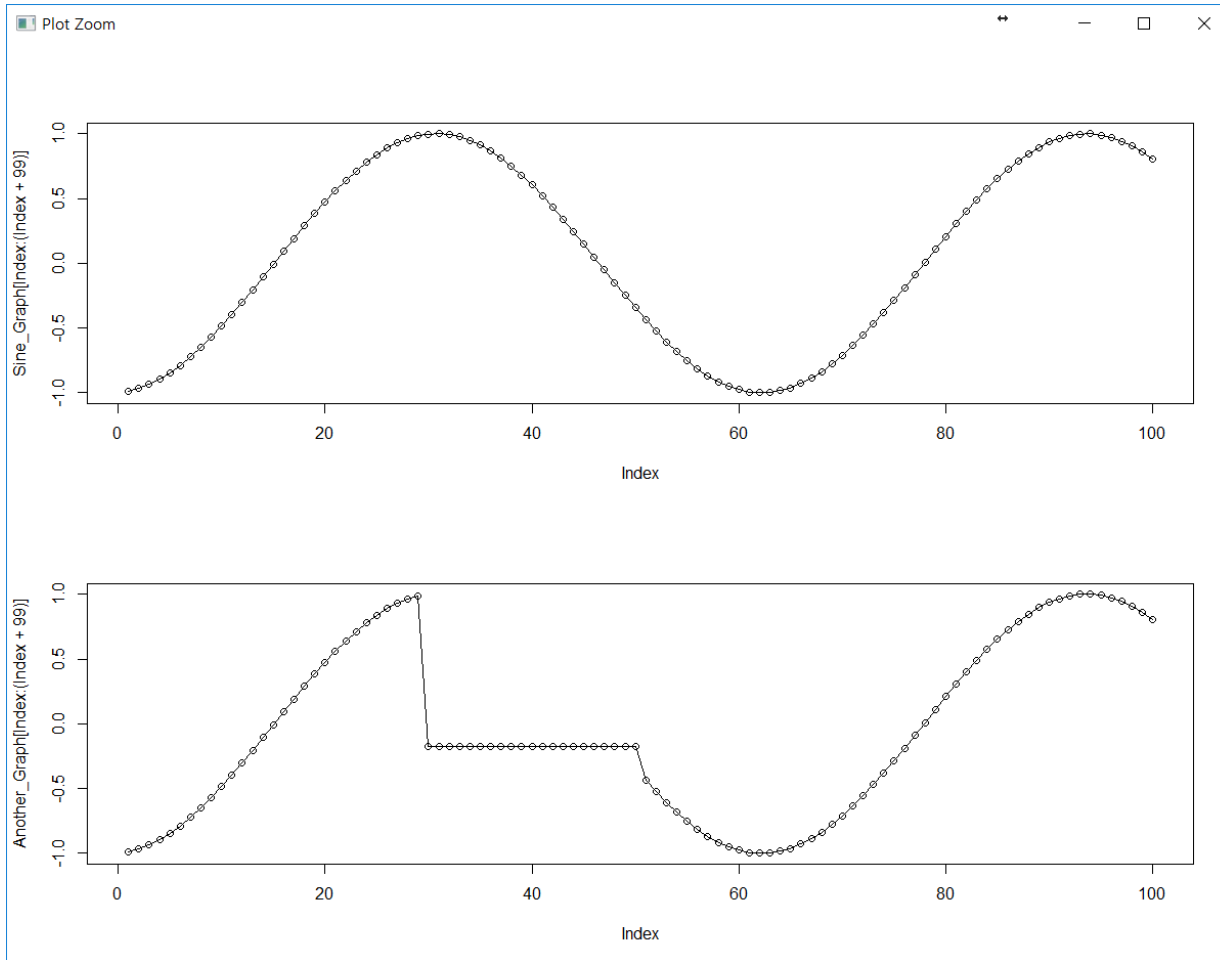
**{Junhongkim@korea.ac.kr}**

**DS·BA**

Data science & Business analytics

# Practice!

- Class activation map for detect noise period! (Sine vs Noise Sine)
- Raw\_Data\_Generate.R → Raw data generate code

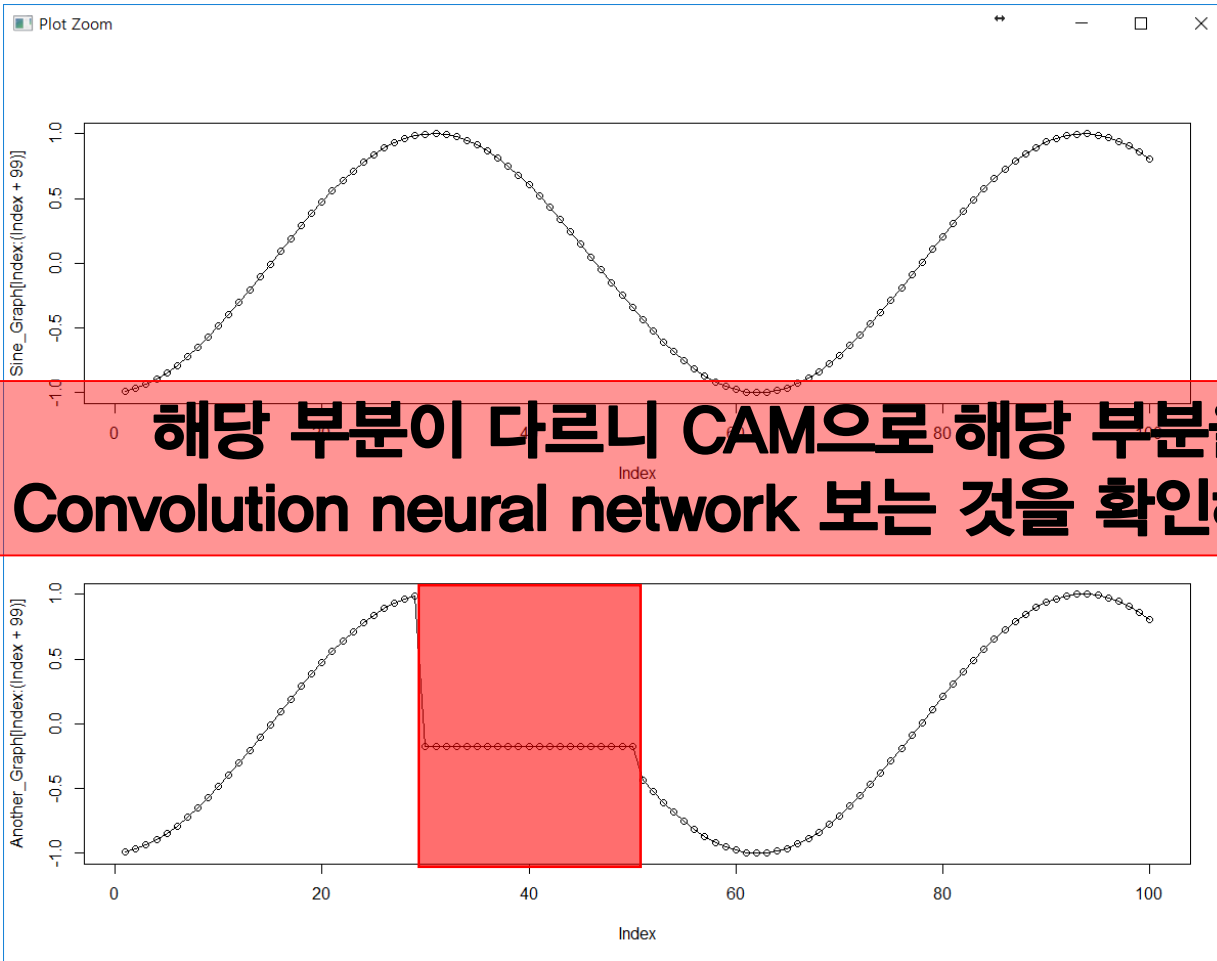


〈 Class 1 〉  
100 Point의 일반적인 Sine graph

〈 Class 2 〉  
100 Point의 일반적인 Sine graph  
+20 Point에 noise 추가함

# Practice!

- Class activation map for detect noise period! (Sine vs Noise Sine)
- Raw\_Data\_Generate.R → Raw data generate code



해당 부분이 다르니 CAM으로 해당 부분을  
Convolution neural network 보는 것을 확인해 보자!

〈 Class 1 〉  
100 Point의 일반적인 Sine graph

〈 Class 2 〉  
100 Point의 일반적인 Sine graph  
+20 Point에 noise 추가함

# Practice!

- Class activation map for detect noise period! (Sine vs Noise Sine)
- Binary classification 이기 때문에 One-hot encoding시 2차원의 vector로 표현

```
-----  
Check of Dataset Size!  
-----  
Train_Input_Size : (22400, 100)  
Test_Input_Size : (9600, 100)  
Train_Target_Size : (22400, 2)  
Train_Target_Size : (9600, 2)
```



# Practice!

- 생각해 보자. Input Dimension은 1\*100\*1임 따라서 우리는 1-D convolution을 하는 것임
- 1-D convolution은 signal dataset혹은 time series dataset에서 많이 사용함
- 저번시간에 배운 image 2D convolution을 생각해보면서 비교해 봅시다 (생각해봅시다!)

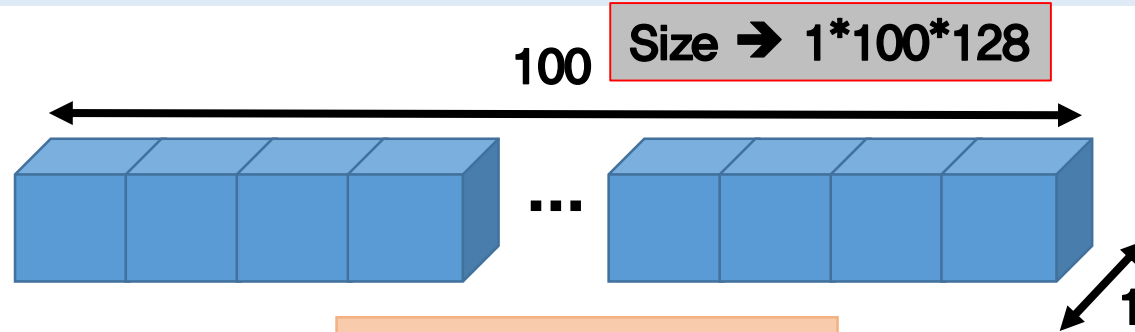
```
In[23]: np.shape(Train_Input)
Out[23]:
(22400, 1, 100, 1)

In[24]: np.shape(Test_Input)
Out[24]:
(9600, 1, 100, 1)
```

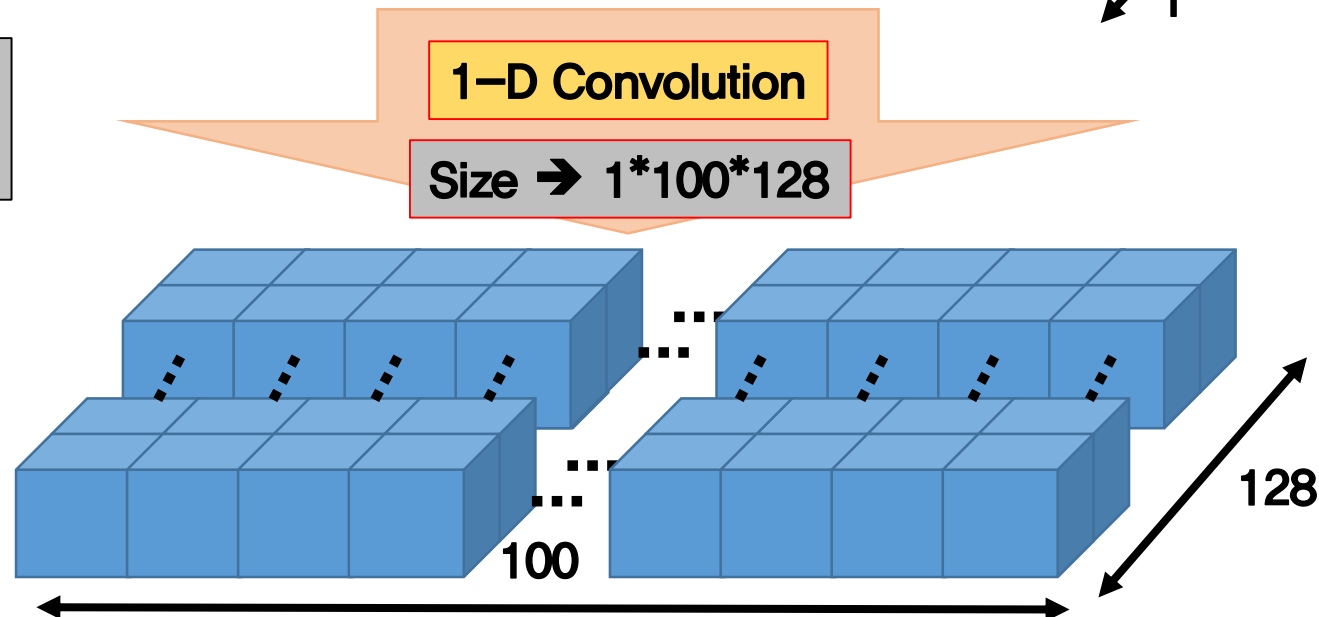
# Practice!

- 생각해 보자. Input Dimension은  $1 \times 100 \times 1$ 임 따라서 우리는 1-D convolution을 하는 것임
- 1-D convolution은 signal dataset혹은 time series dataset에서 많이 사용함
- 저번시간에 배운 image 2D convolution을 생각해보면서 비교해 봅시다 (생각해봅시다!)

Input

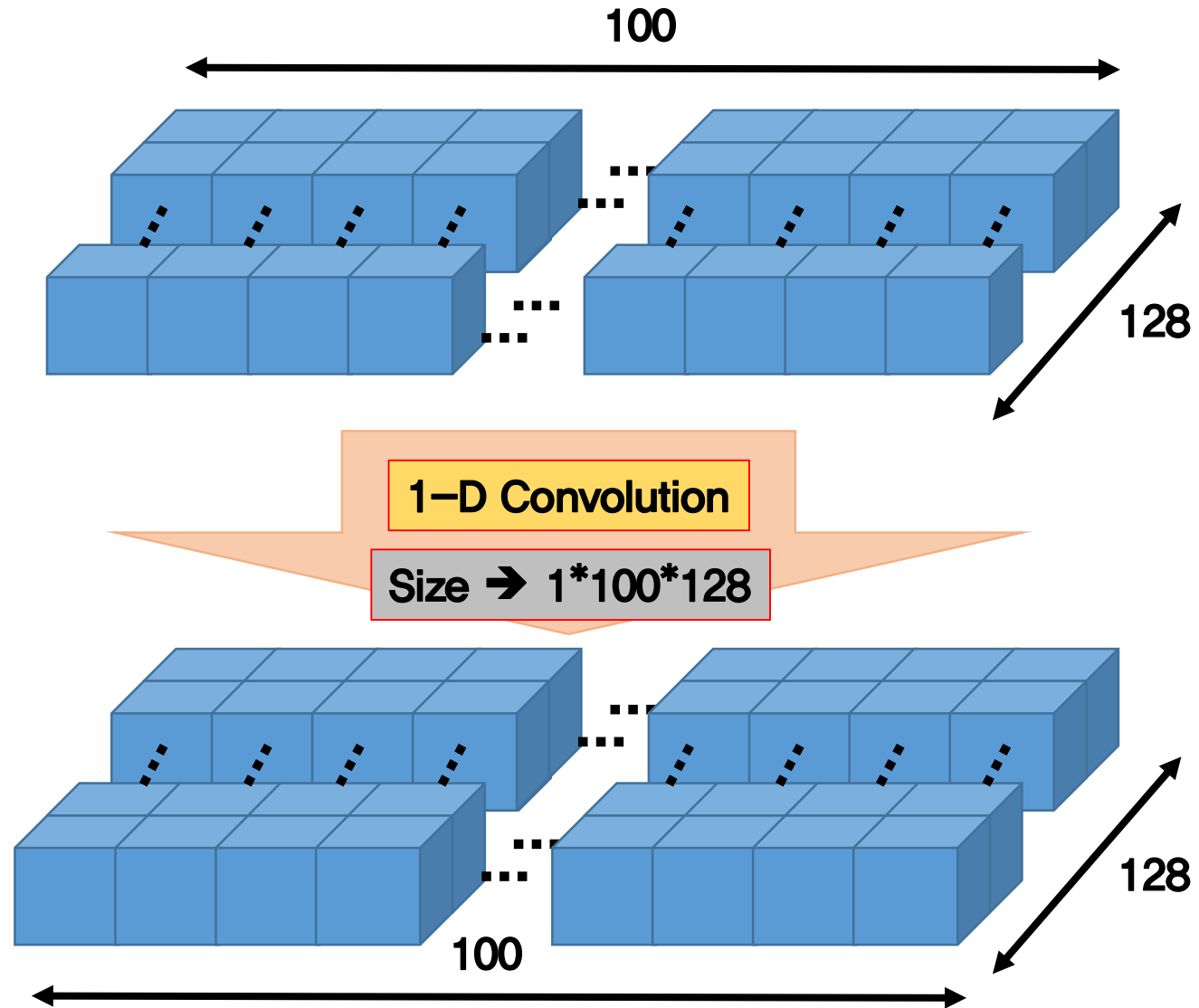


First conv



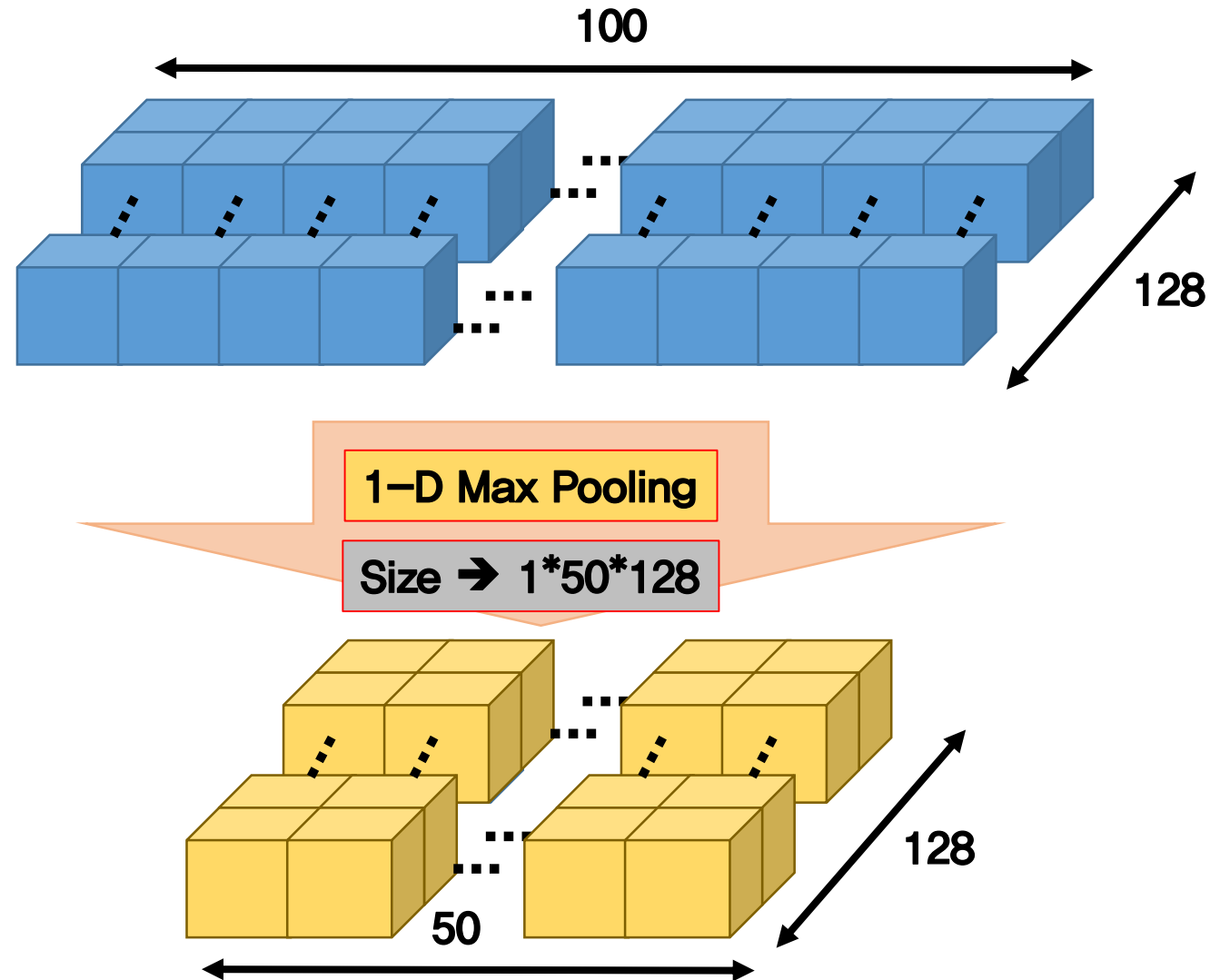
# Practice!

## Second conv



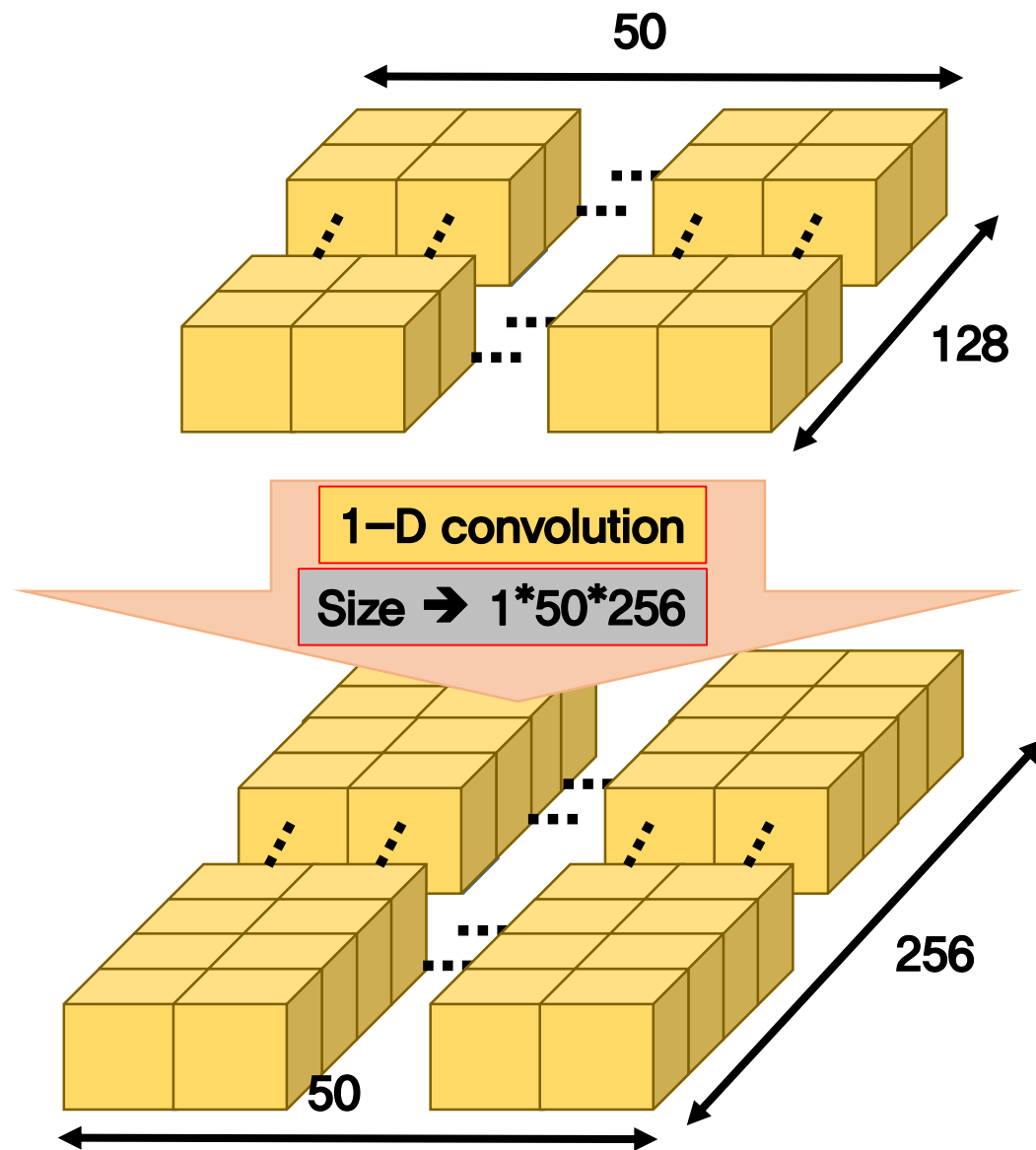
# Practice!

## First Pooling



# Practice!

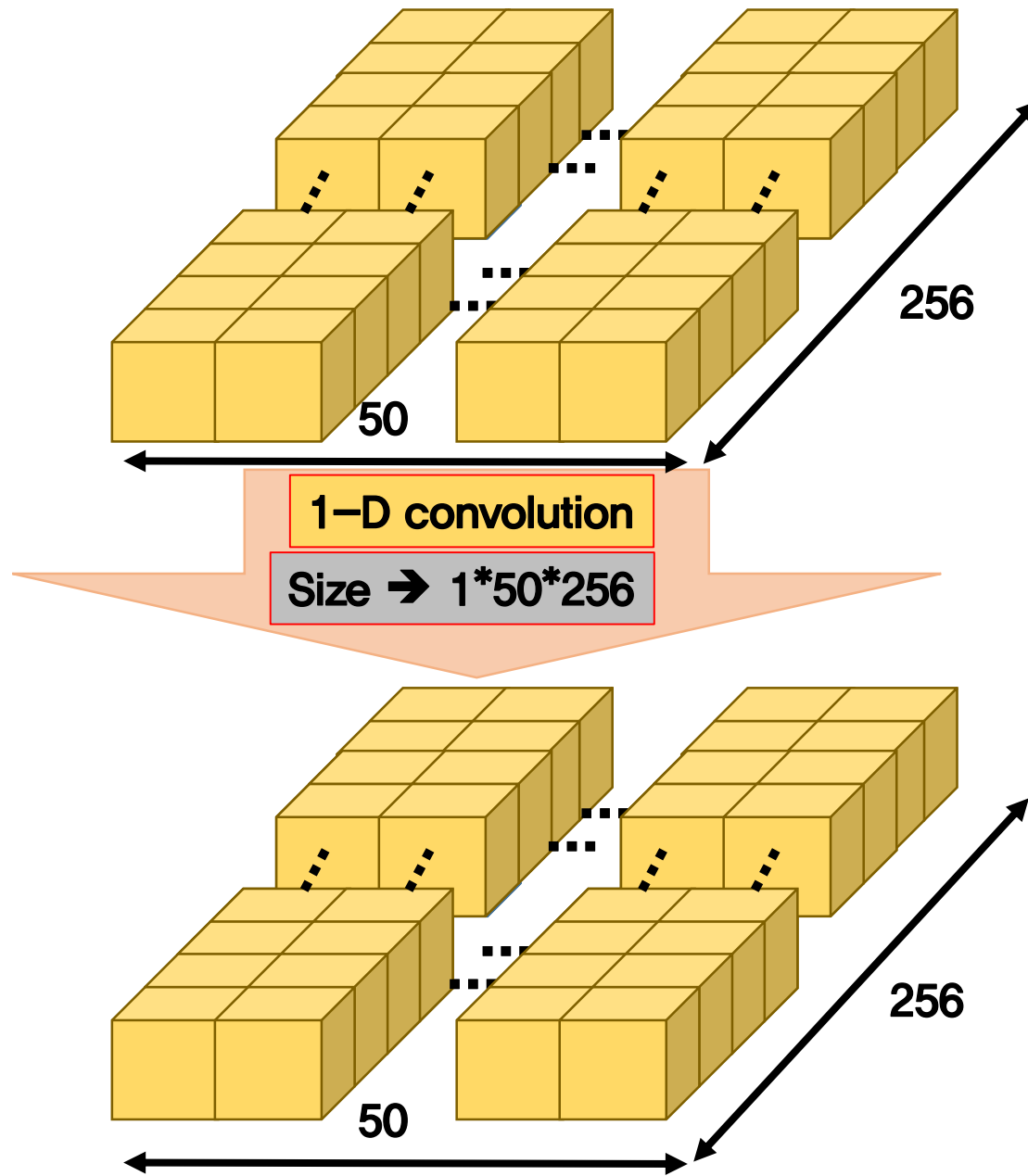
**Third  
conv**





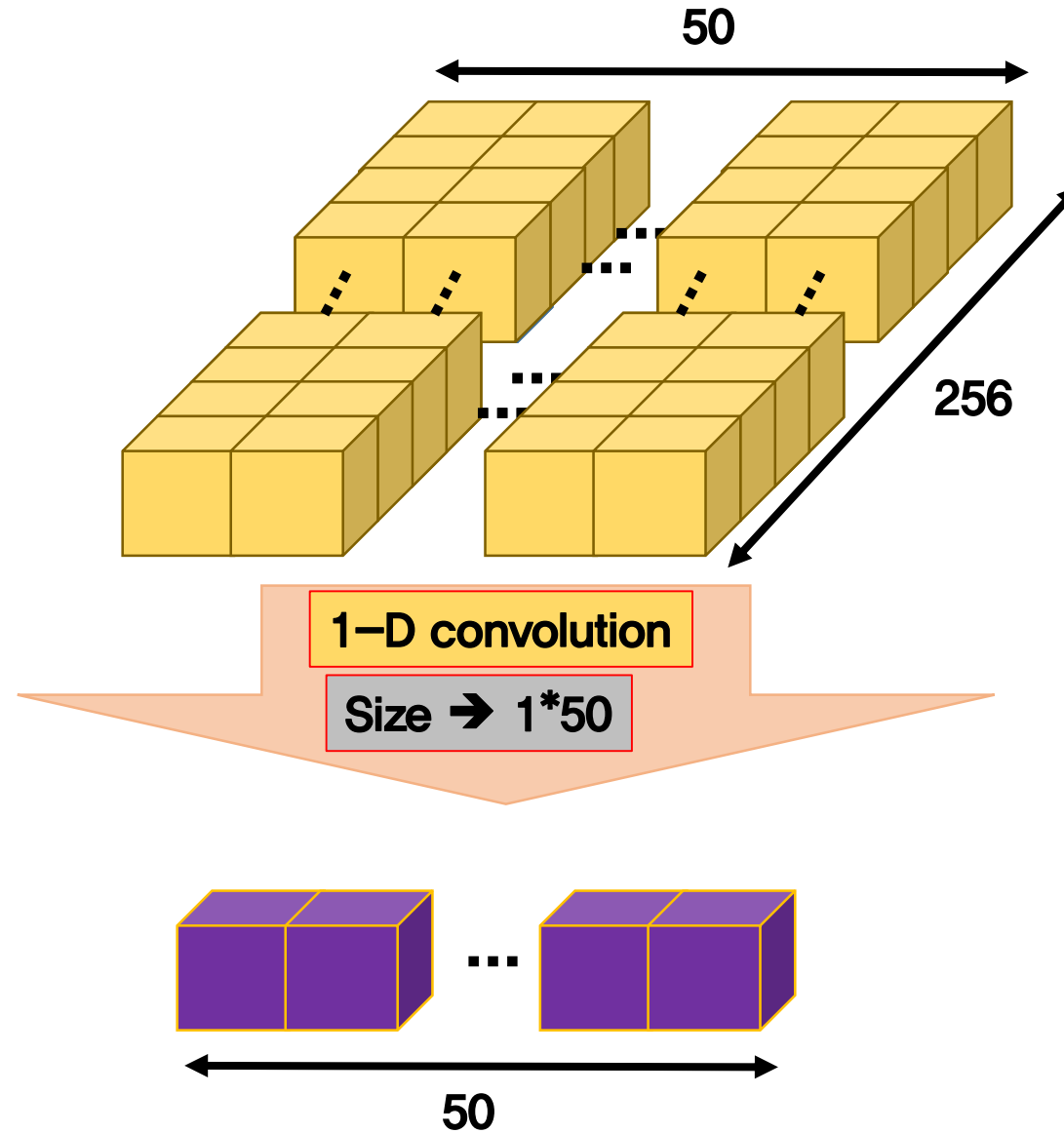
# Practice!

**Fouth  
conv**



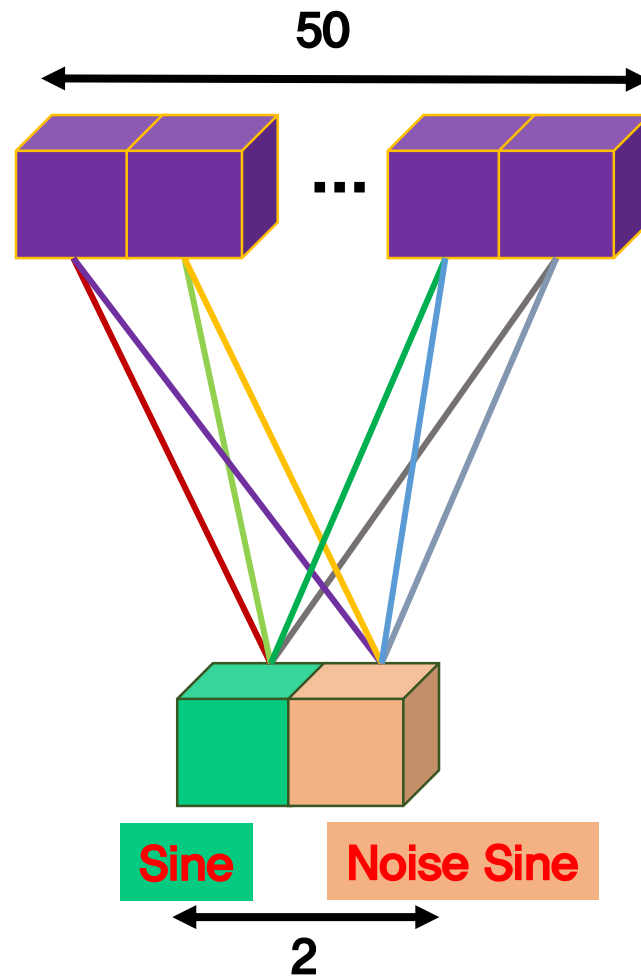
# Practice!

**Global  
Average  
Pooling**

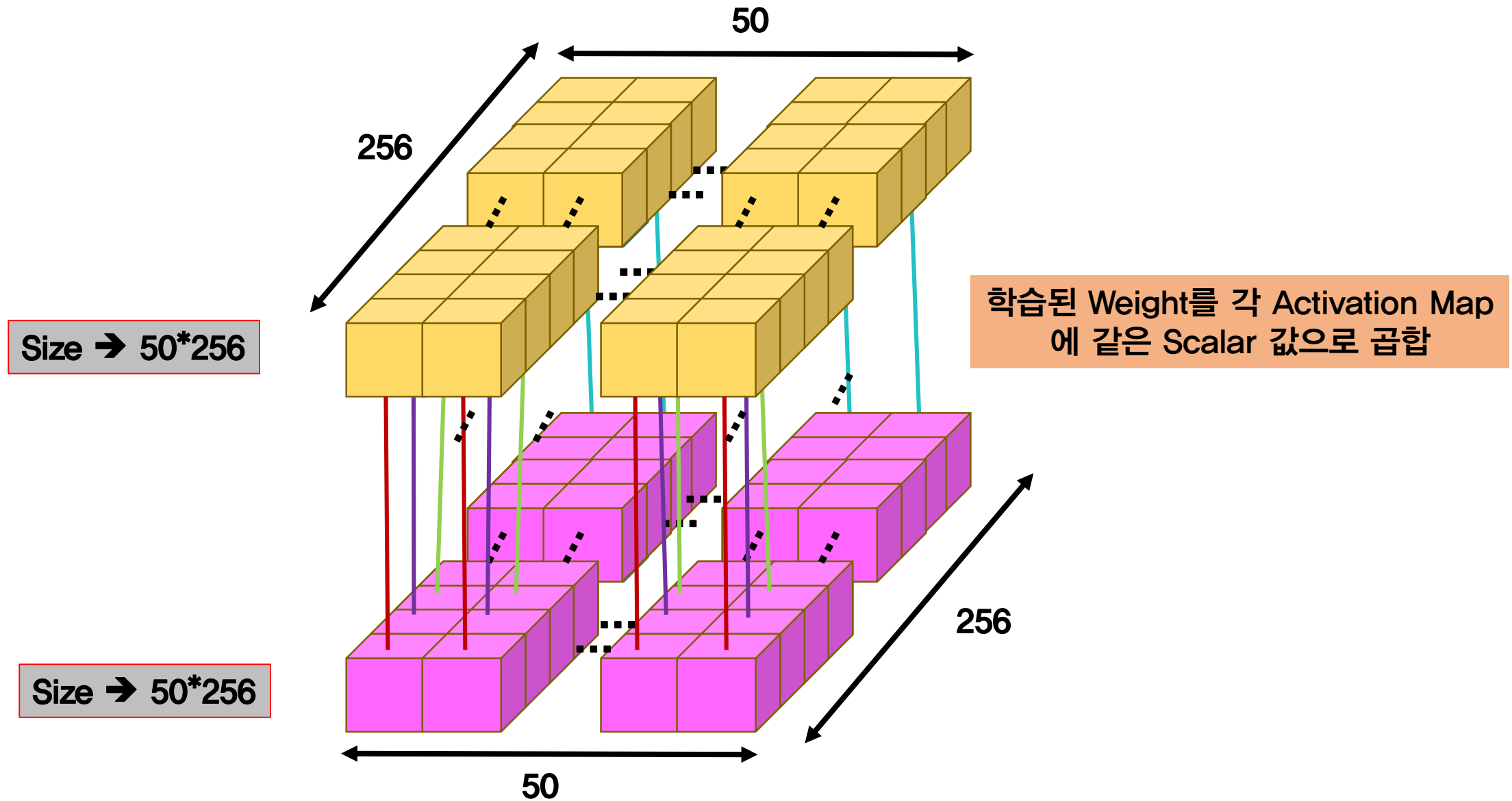


# Practice!

**Linear  
Combination**

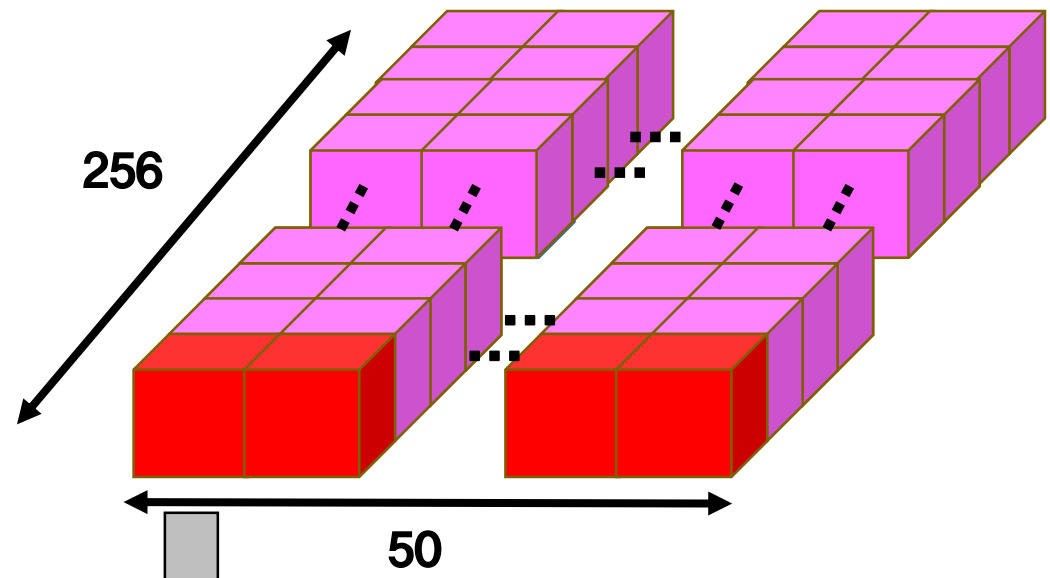


# Practice!



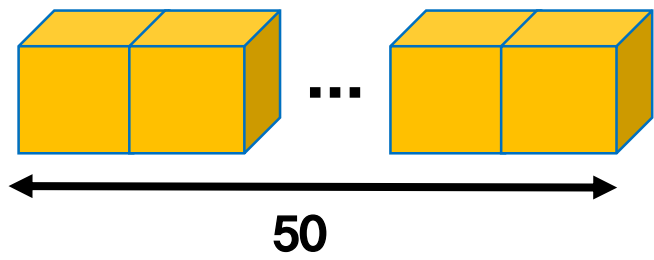
# Practice!

Size → 50\*256



Activation Map의 개수  
만큼을 전부 더함

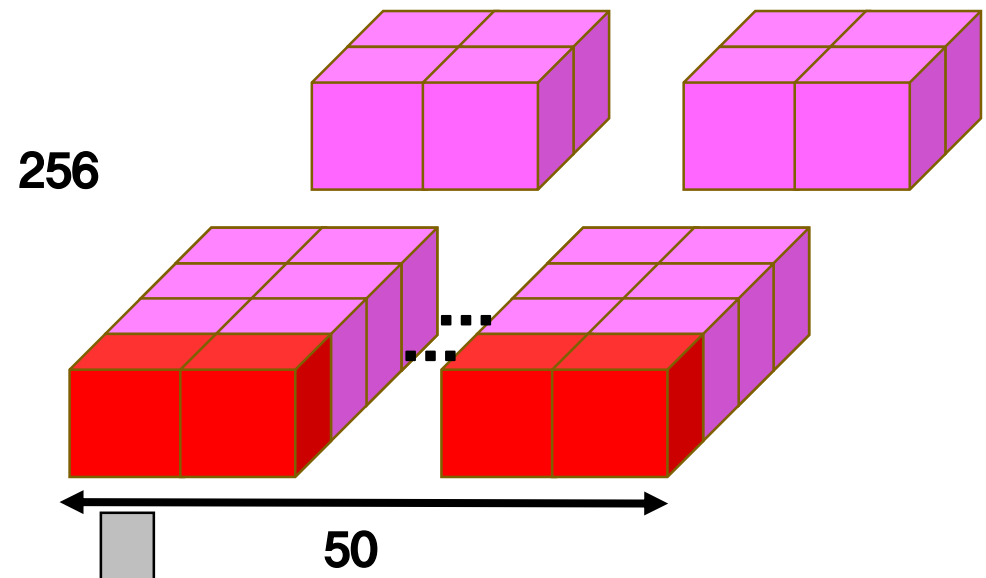
Size → 50





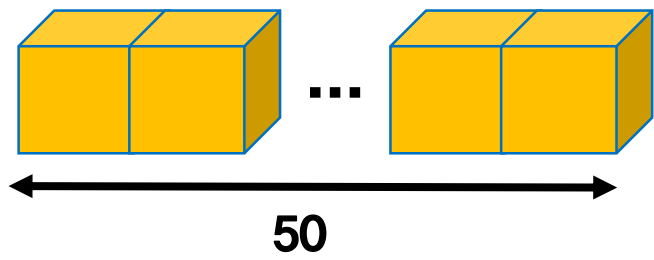
# Practice!

Size → 50\*256



Activation Map의 개수  
만큼을 전부 더함

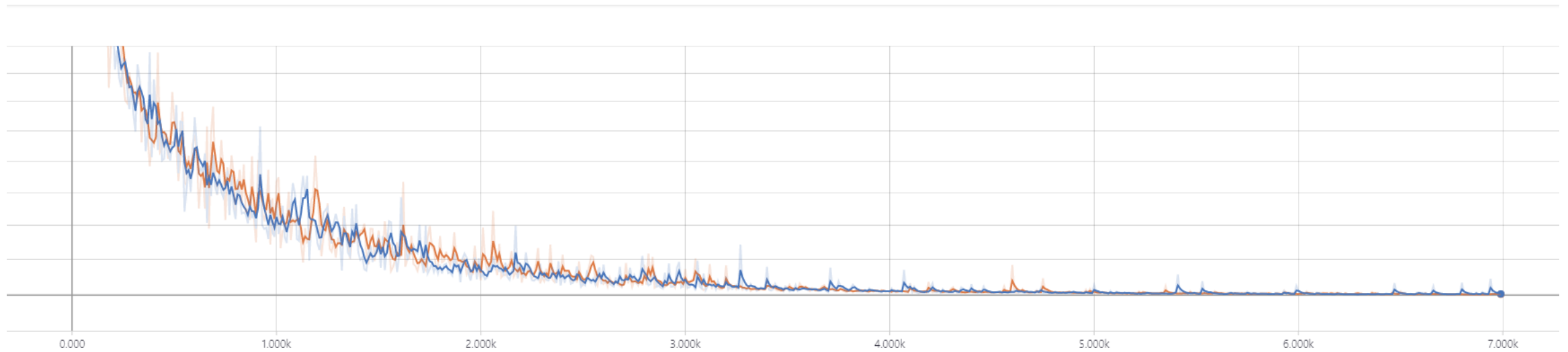
Size → 50  
256 차원은  
Element wise sum!



이것이 Class Activation Map score임

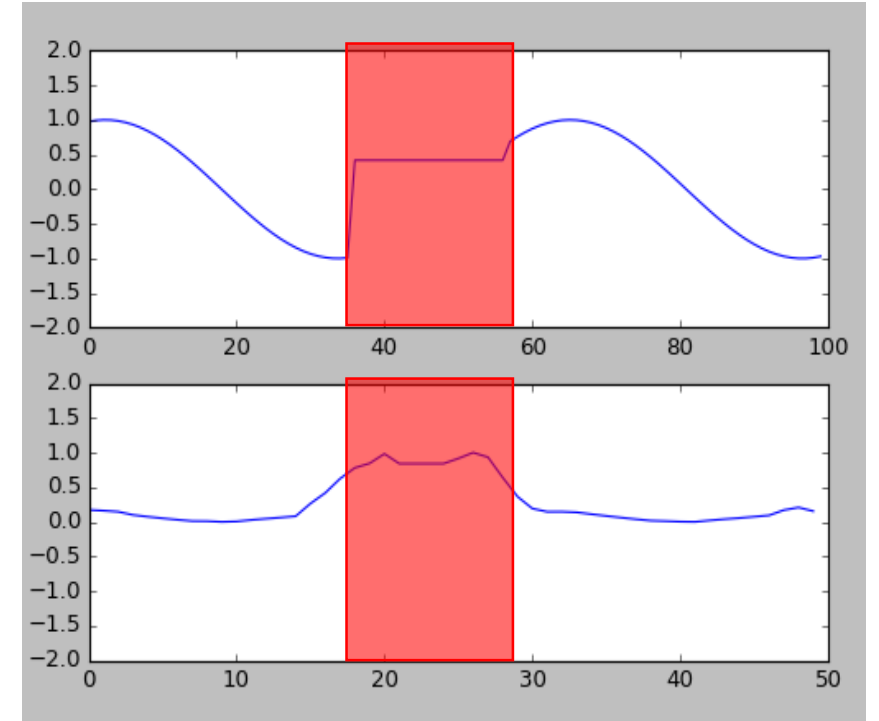
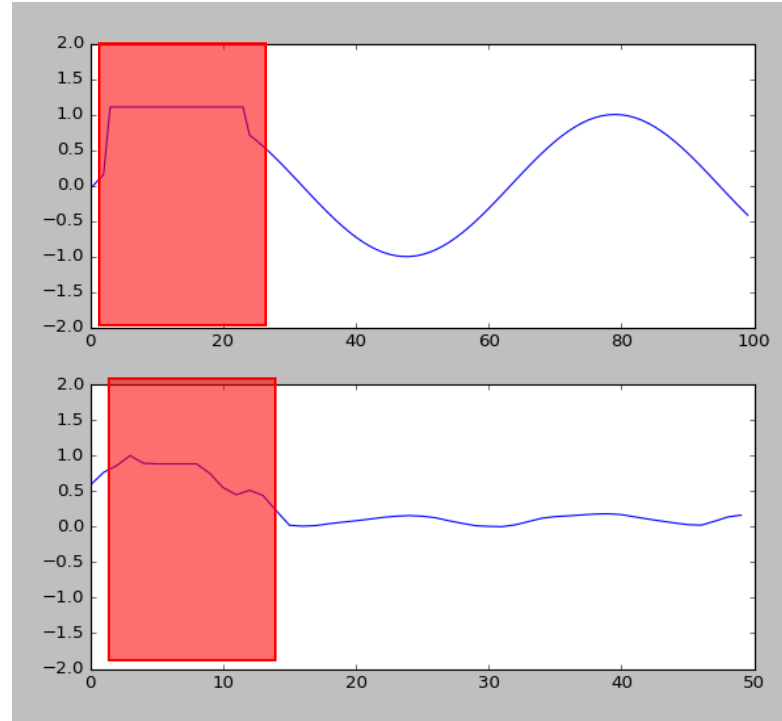
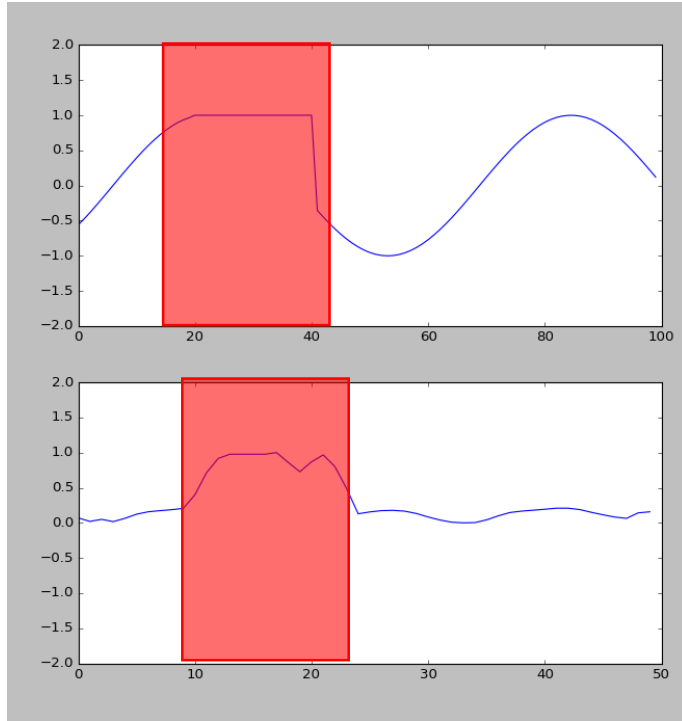
# Practice!

- Class activation map for detect noise period! (Sine vs Noise Sine)
- 10 Epoch!



# Practice!

- 원래 자료에 Mapping하려면 interpolation을 해야함
- 비율로써 시각화함 CAM을 통하여 Noise 부분을 잘 나타내는 것을 확인 할 수 있음



Thank you!