

# **CAM / Grad CAM**

김현우

# Learning Deep Feature For Discriminative Localization

n.org  
ht on

## Learning Deep Features for Discriminative Localization

3S  
a  
i  
ive ...

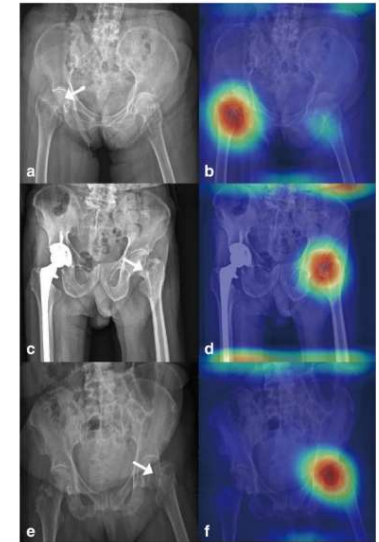
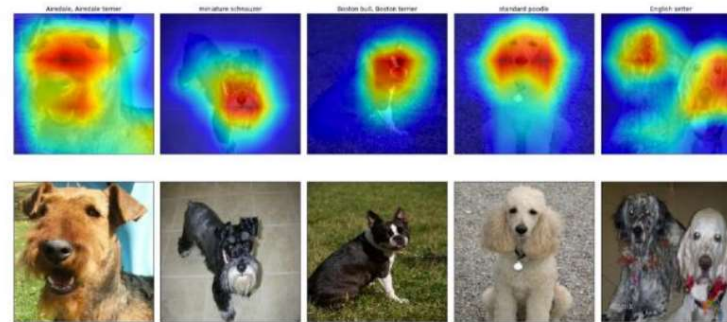
Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba  
Computer Science and Artificial Intelligence Laboratory, MIT  
{bzhou, khosla, agata, oliva, torralba}@csail.mit.edu

### Abstract

In this work, we revisit the global average pooling layer proposed in [13], and shed light on how it explicitly enables the convolutional neural network (CNN) to have remarkable localization ability despite being trained on image-level labels. While this technique was previously proposed as a means for regularizing training, we find that it actually builds a generic localizable deep representation that exposes the implicit attention of CNNs on an image. Despite the apparent simplicity of global average pooling, we are able to achieve 37.1% top-5 error for object localization on ILSVRC 2014 without training on any bounding box annotation. We demonstrate in a variety of experiments that our network is able to localize the discriminative image regions despite just being trained for solving classification task<sup>1</sup>.



Figure 1. A simple modification of the global average pooling layer combined with our class activation mapping (CAM) technique allows the classification-trained CNN to both classify the image and localize class-specific image regions in a single forward-pass e.g., the toothbrush for brushing teeth and the chainsaw for cutting trees.



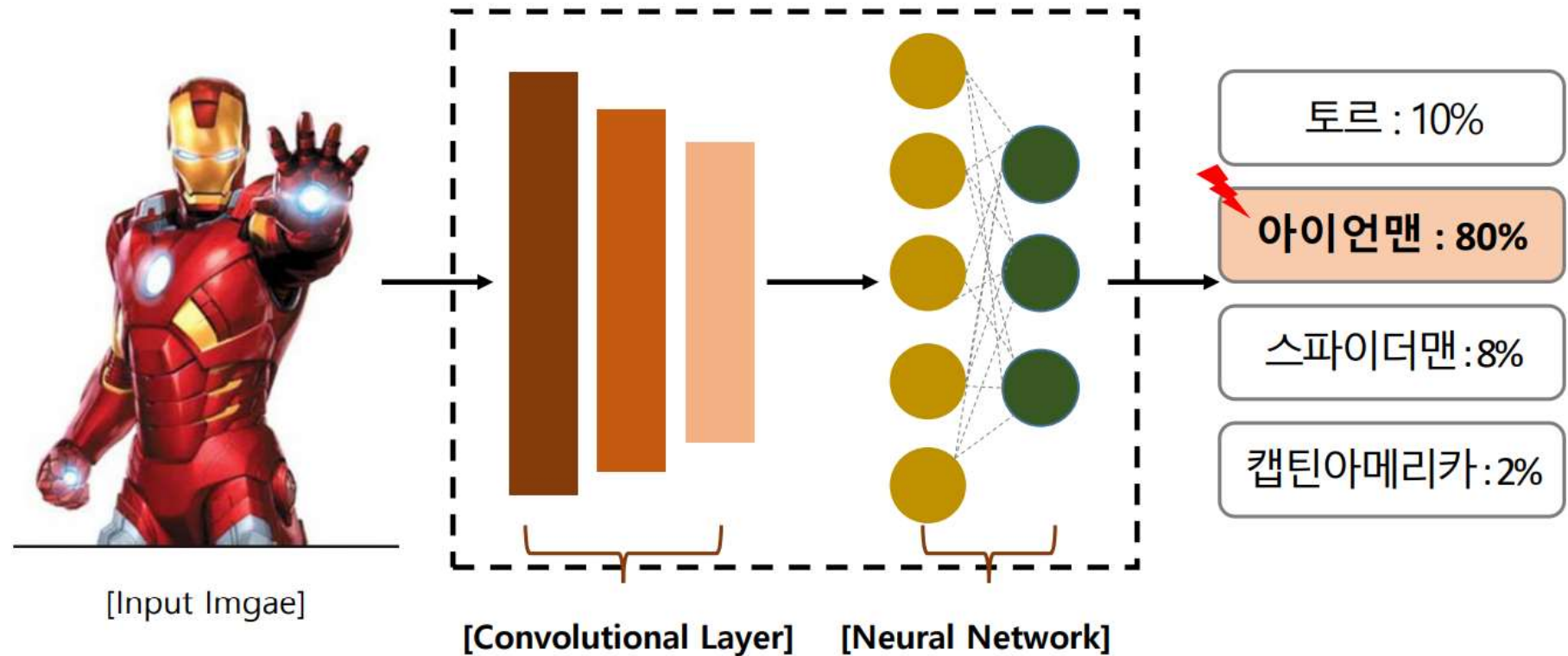
<분류 모델 원인 해석>  
→ 예측 모델 결과에 대한 신뢰성

<고관절 골절 탐지>  
→ 병 원인 진단

- 목적 : CNN을 해석 하고자 하는 방법 중 하나
- 아이디어 : Class를 분류할 때 이미지의 어떤 영역이 큰 영향을 미쳤는지 파악
- 방법 : Global Average Pooling

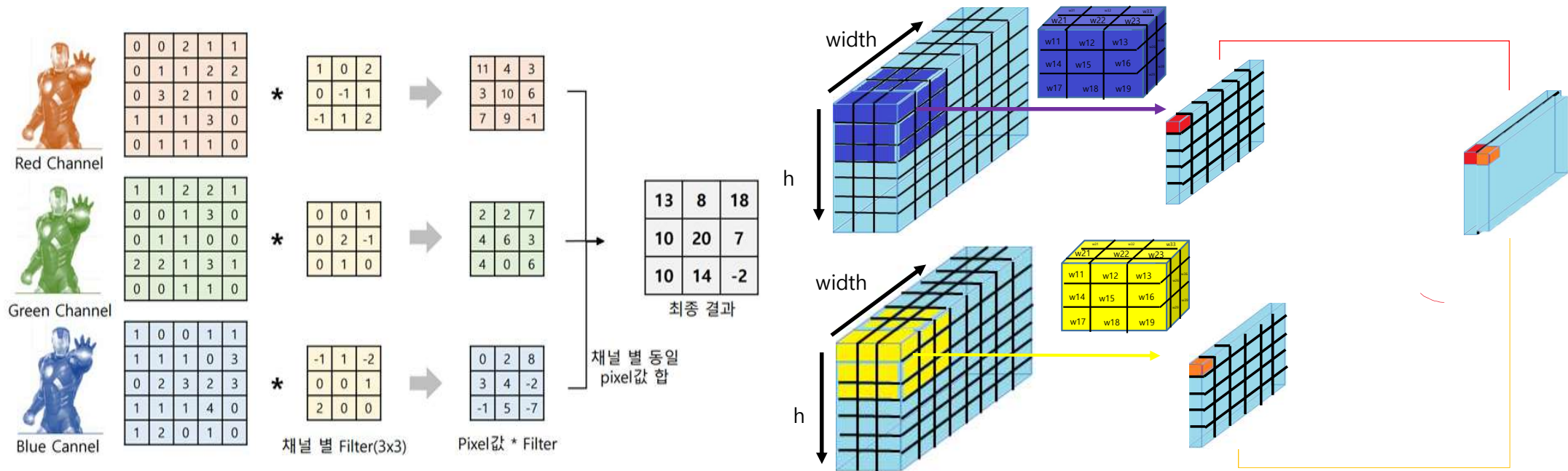
# 사전 지식

- Neural Network에 Convolution Layer를 사용
- 물체검출, 분류 등 이미지에서 좋은 성능을 나타냄



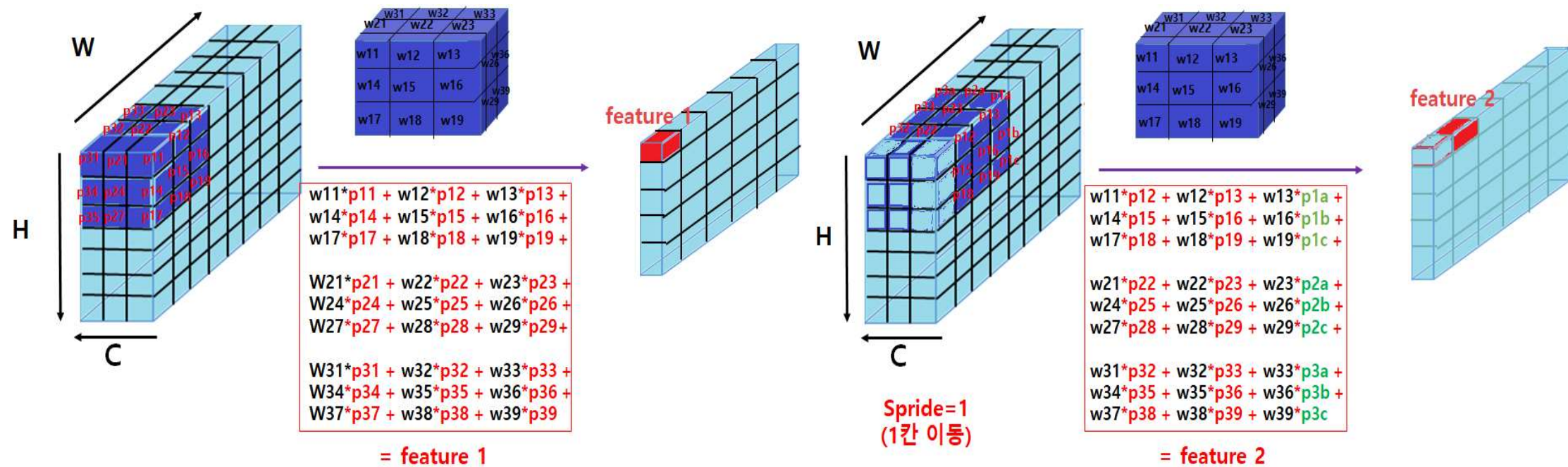
# 사전 지식

- Filter : 입력 이미지에서 feature(특징)을 찾아내기 위한 모델 파라미터
- Convolution : Filter를 이동(stride)하면서 곱한결과를 더하는 과정



# 사전 지식

- Filter : 입력 이미지에서 feature(특징)을 찾아내기 위한 모델 파라미터
- Convolution : Filter를 이동(stride)하면서 곱한결과를 더하는 과정

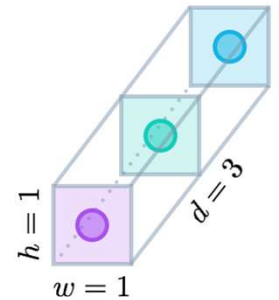
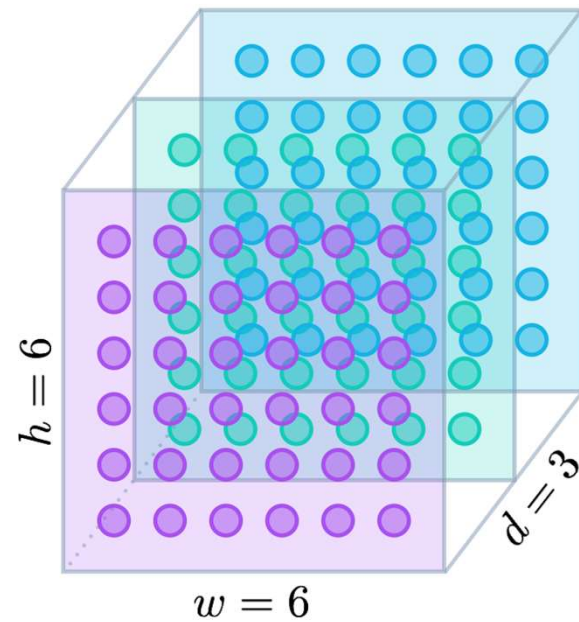
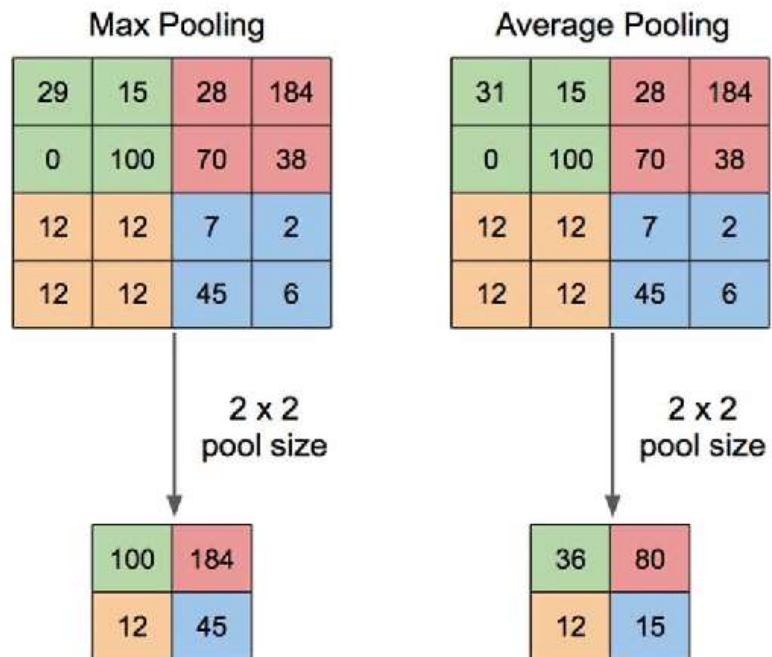




# 사전 지식

## - Pooling

- Convolution Layer의 출력값을 줄이는 목적
- Feature 값을 강조하는 (대표값) 추출하는 목적
- 학습하는 파라미터가 없음

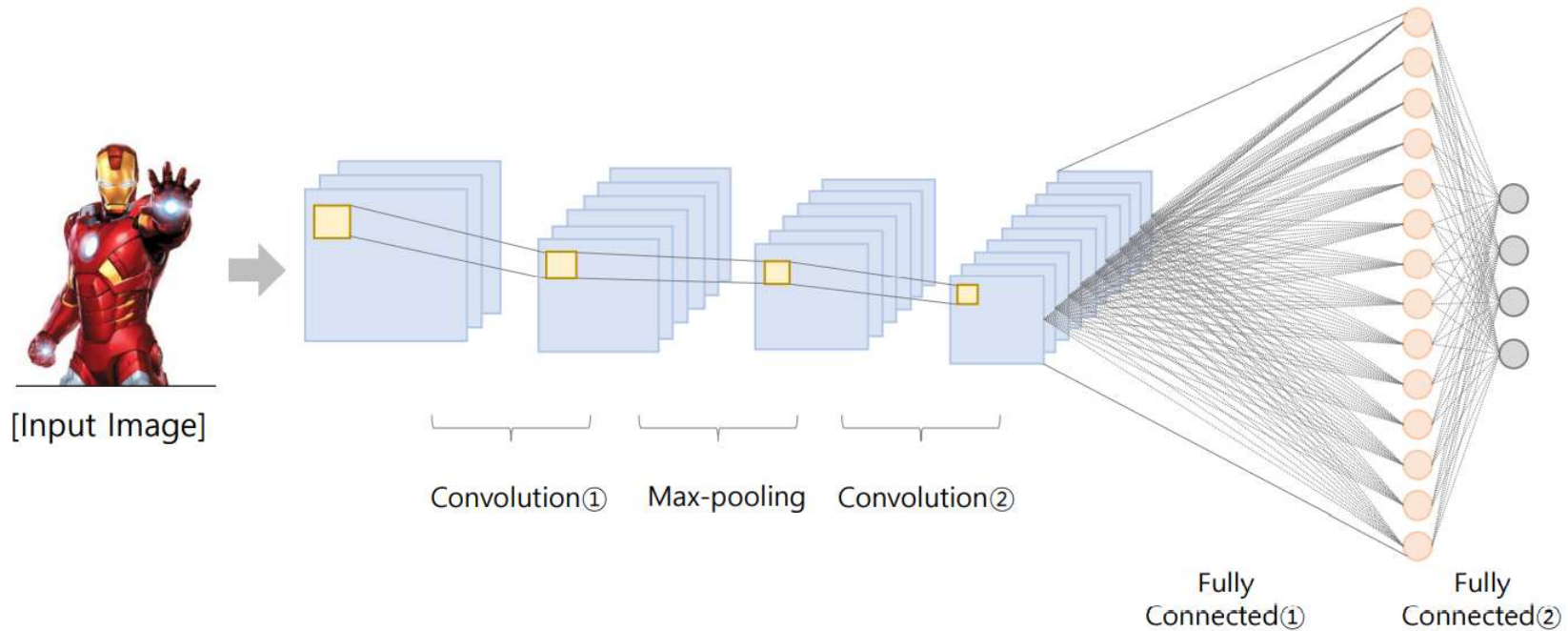


# CAM(Class Activation Map)



- CNN 모델로 예측시 입력의 어떤 부분이 CLASS 예측에 큰영향을 주었는지 확인가능
- 마지막 Convolution Layer 이후 Global Average Pooling을 사용

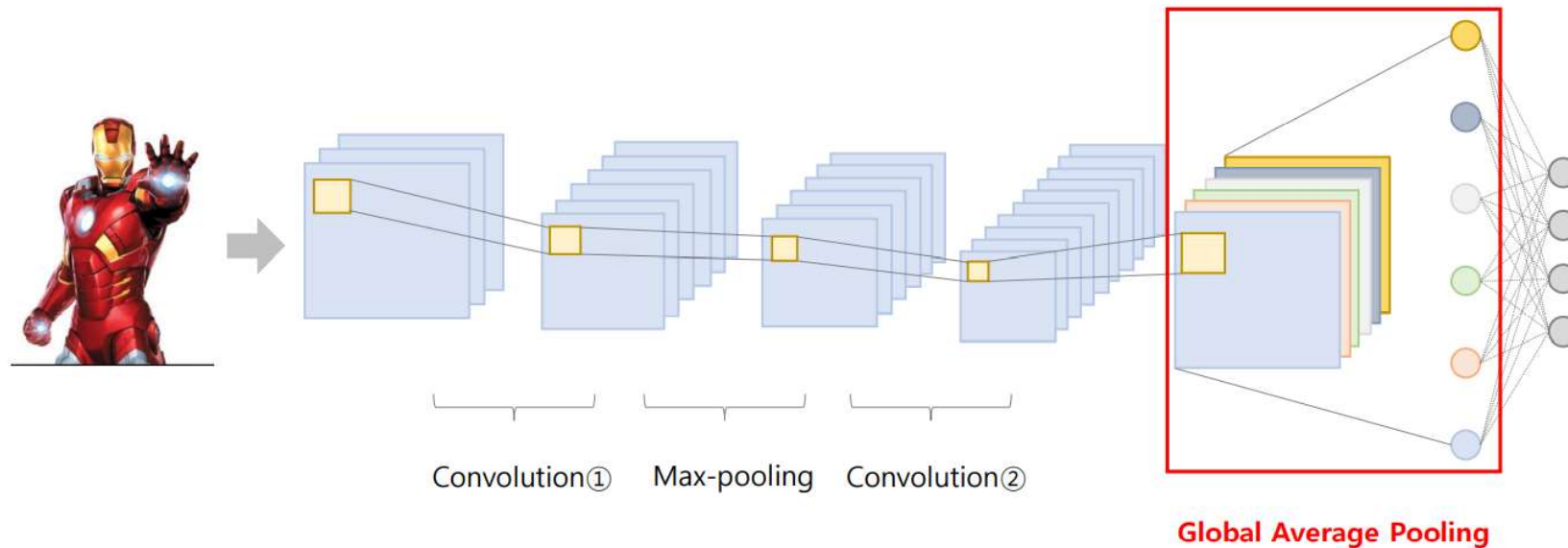
# CAM(Class Activation Map)



- 이미지를 입력으로 받아 이미지의 특징점을 추출하고 이것을 기반으로 class 분류
- Convolution Layer와 Pooling Layer는 이미지의 특징점 추출 역할
- Fully Connected Layer는 이미지의 특징점을 기반으로 분류 역할



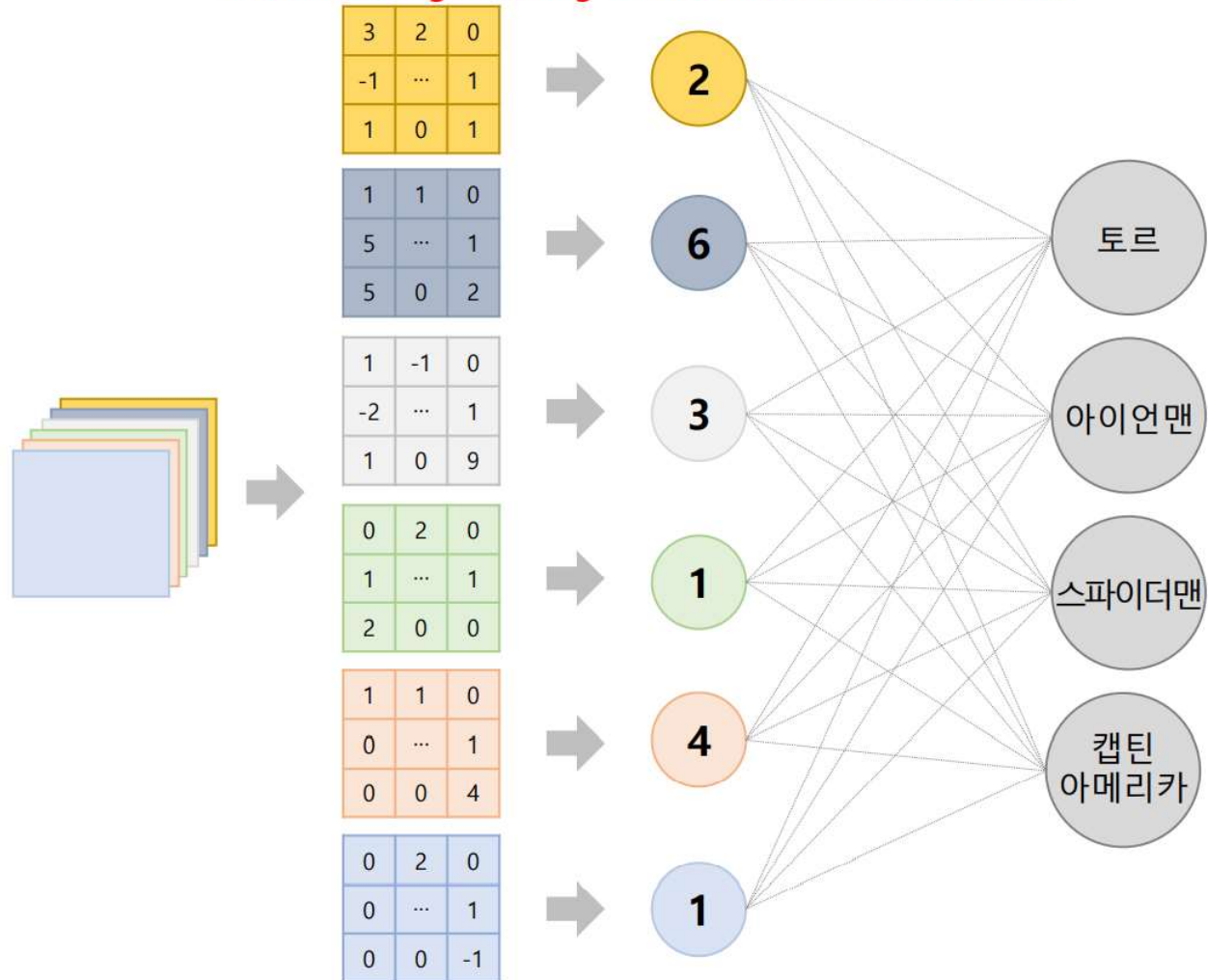
# CAM(Class Activation Map)



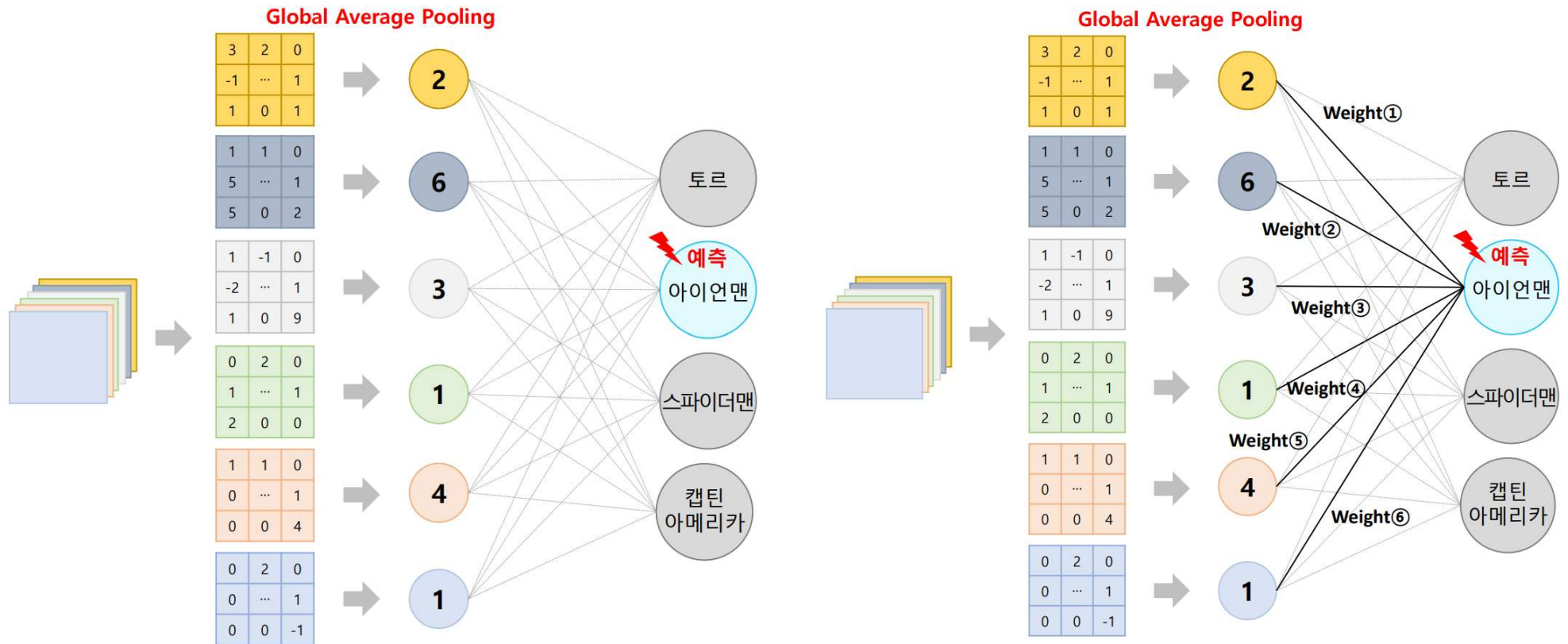
- Convolution Layer와 Pooling Layer는 이미지의 특징점 추출 역할
- ~~Fully Connected Layer는 이미지의 특징점을 기반으로 분류 역할~~
- 마지막 Convolution Layer 뒤에 Global Average Pooling 사용

# CAM(Class Activation Map)

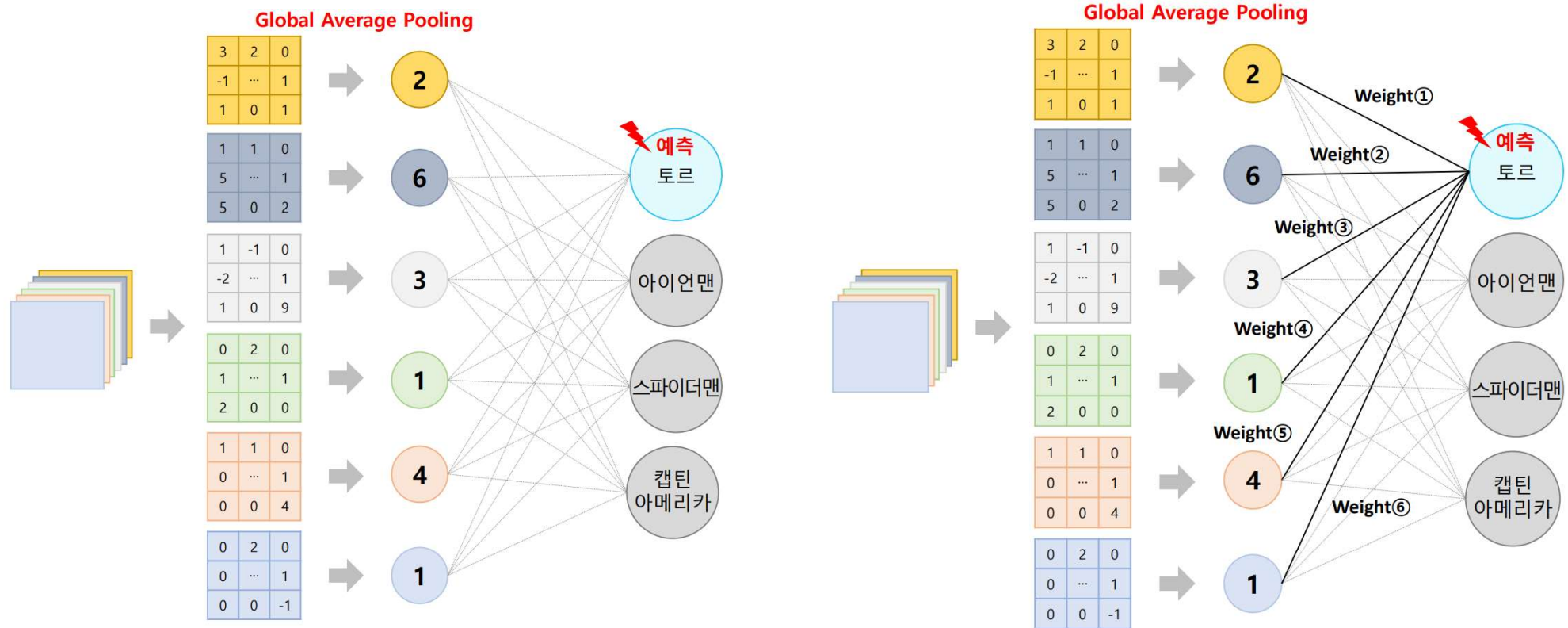
Global Average Pooling → 각 Feature별 평균 값을 구함



# CAM(Class Activation Map)

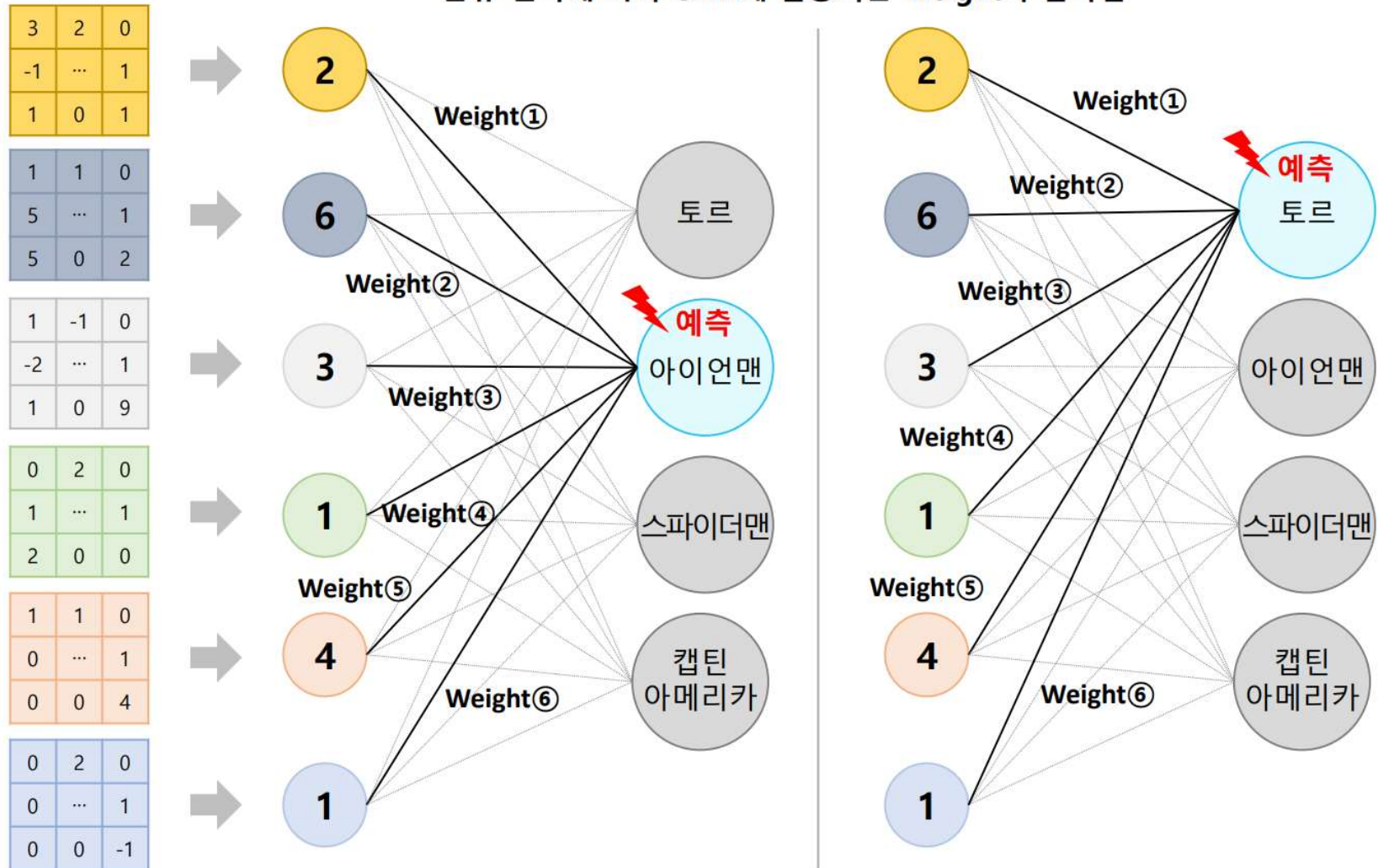


# CAM(Class Activation Map)



# CAM(Class Activation Map)

분류 결과에 따라 CAM에 활용되는 Weight가 달라짐






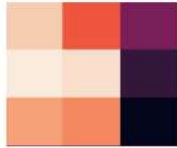
# CAM(Class Activation Map)

각 Feature 별 heatmap을 그림


3	2	0
-1	...	1
1	0	1

$$\times \text{Weight①} =$$


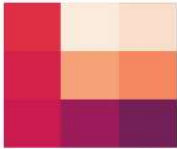
1	1	0
5	...	1
5	0	2

$$\times \text{Weight②} =$$


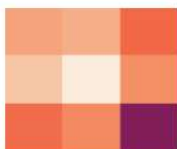
1	-1	0
-2	...	1
1	0	9

$$\times \text{Weight③} =$$



0	2	0
1	...	1
2	0	0

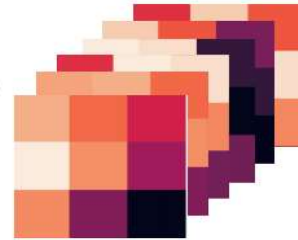
$$\times \text{Weight④} =$$


1	1	0
0	...	1
0	0	4

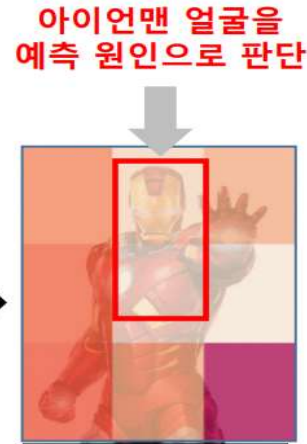
$$\times \text{Weight⑤} =$$


0	2	0
0	...	1
0	0	-1

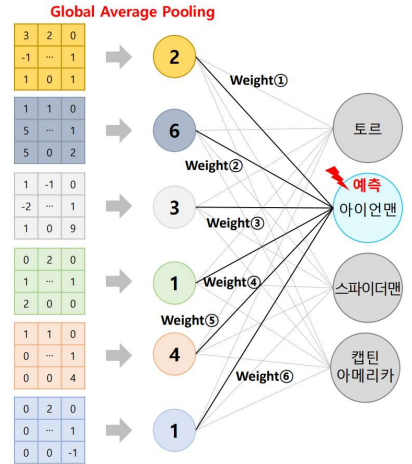
$$\times \text{Weight⑥} =$$




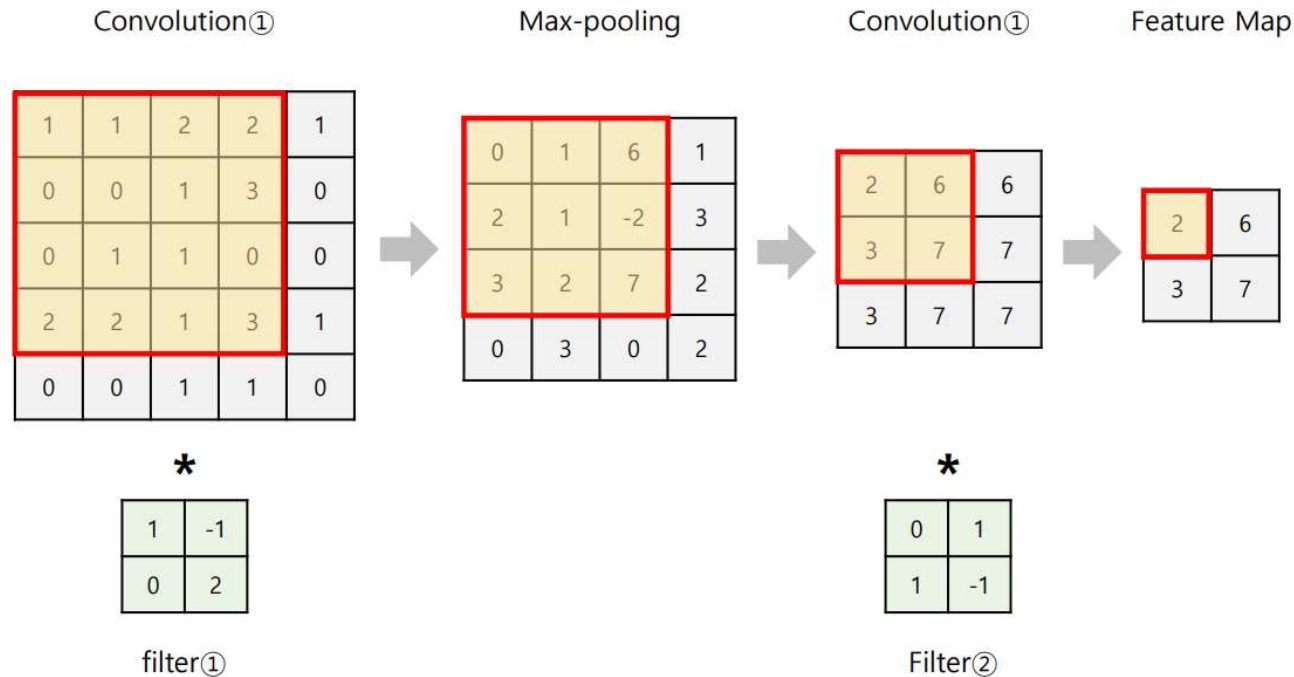
동일 위치  
Pixel 별 합



아이언맨 얼굴을  
예측 원인으로 판단

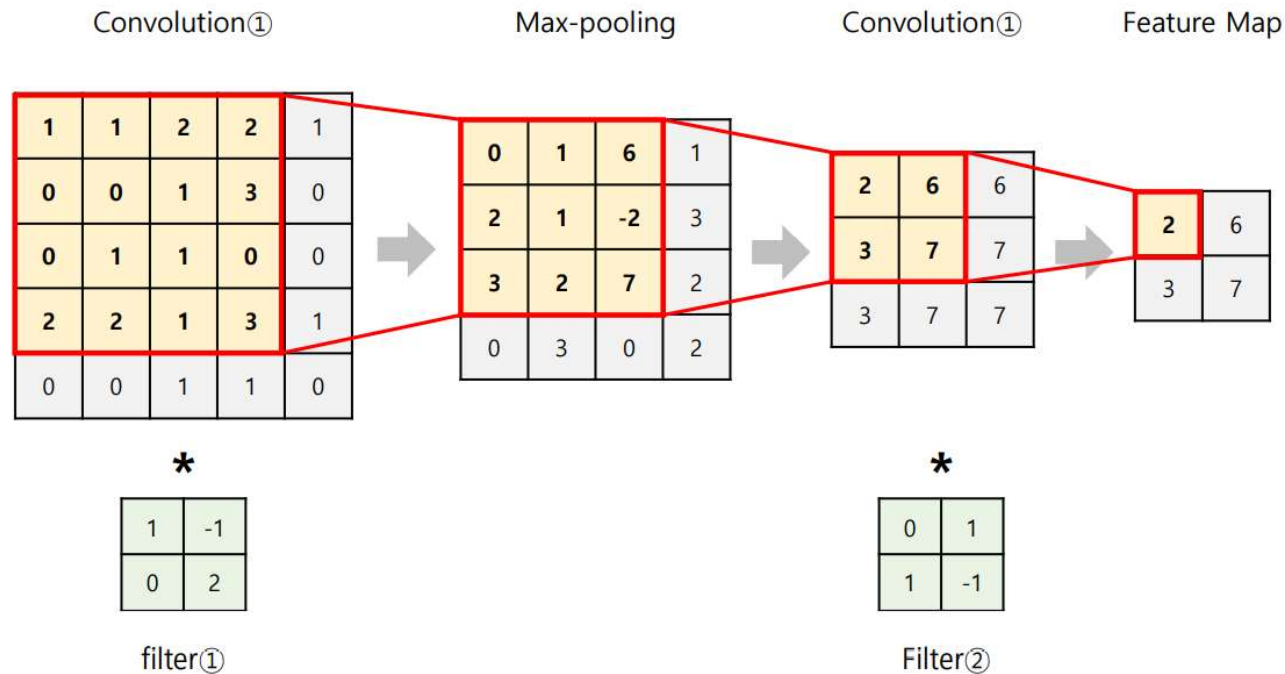


# CAM(Class Activation Map)



- Class Activation Map에서 마지막 Layer만으로도 분석이 가능한 이유?
  - 마지막 Feature Map이 가진 정보량이 많음
  - 자미가 Feature Map내 1개의 값은 원본이미지에서 많은 부분을 요약한 결과

# CAM(Class Activation Map)



- Class Activation Map에서 마지막 Layer만으로도 분석이 가능한 이유?
  - 마지막 Feature Map이 가진 정보량이 많음
  - 자미가 Feature Map내 1개의 값은 원본이미지에서 많은 부분을 요약한 결과

# Grad\_CAM

## Grad-cam: Visual explanations from deep networks via gradient-based localization

..., [R. Vedantam](#), [D. Parikh](#), [D. Batra](#) - ... Computer Vision, 2017 - openaccess.thecvf.com

... 4, some failures **are** due to ambiguities inherent in ImageNet classification ... Although the train model achieved a good validation accuracy, it **did** not generalize as well (82 ... **Grad-CAM** visualizations of the model predictions revealed that the model had learned to look at the ...

☆ 99 1246회 인용 관련 학술자료 전체 6개의 버전



This ICCV paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the version available on IEEE Xplore.

## Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Ramprasaath R. Selvaraju<sup>1\*</sup> Michael Cogswell<sup>1</sup> Abhishek Das<sup>1</sup> Ramakrishna Vedantam<sup>1\*</sup>

Devi Parikh<sup>1,2</sup> Dhruv Batra<sup>1,2</sup>

<sup>1</sup>Georgia Institute of Technology <sup>2</sup>Facebook AI Research

{ramprs, cogswell, abhshkdz, vrama, parikh, dbatra}@gatech.edu

### Abstract

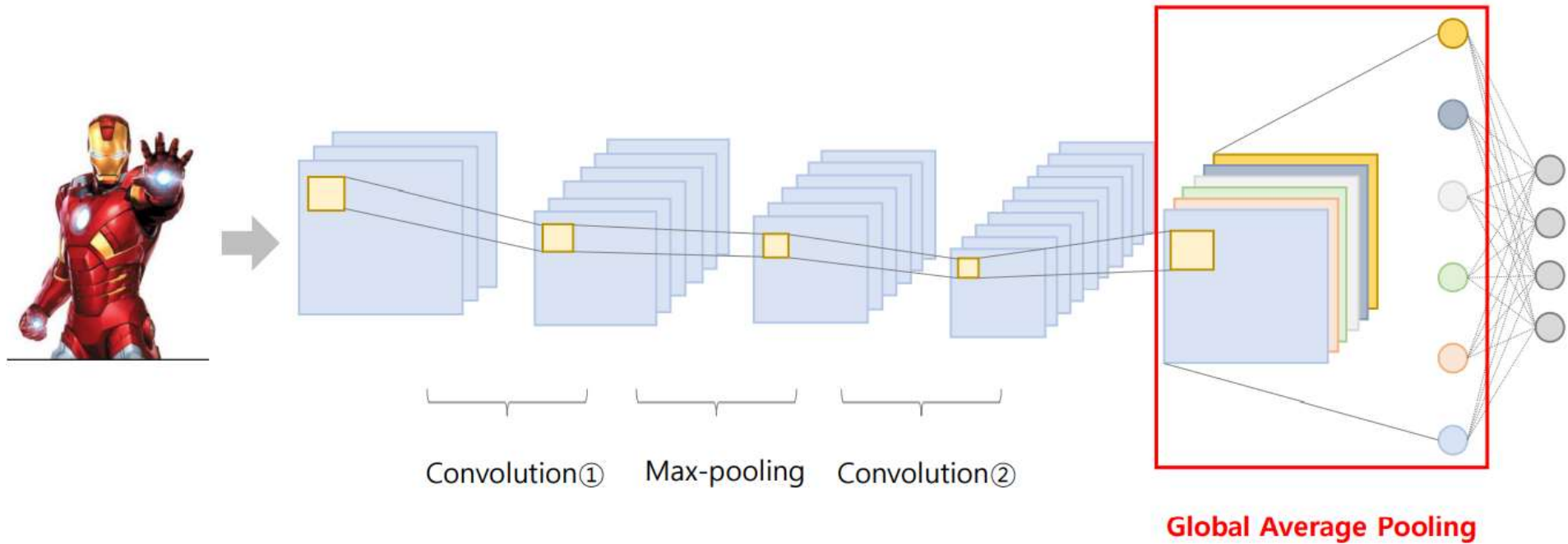
We propose a technique for producing 'visual explanations' for decisions from a large class of Convolutional Neural Network (CNN)-based models, making them more transparent. Our approach – Gradient-weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept (say logits for 'dog' or even a caption), flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. Unlike previous approaches, Grad-CAM is applicable to a wide variety of CNN model-families: (1) CNNs with fully-connected layers (e.g. VGG), (2) CNNs used for structured outputs (e.g. captioning), (3) CNNs used in tasks with multi-modal inputs (e.g. visual question an-

### 1. Introduction

Convolutional Neural Networks (CNNs) and other deep networks have enabled unprecedented breakthroughs in a variety of computer vision tasks, from image classification [24, 16] to object detection [15], semantic segmentation [27], image captioning [43, 6, 12, 21], and more recently, visual question answering [3, 14, 32, 36]. While these deep neural networks enable superior performance, their lack of decomposability into intuitive and understandable components makes them hard to interpret [26]. Consequently, when today's intelligent systems fail, they fail spectacularly disgracefully, without warning or explanation, leaving a user staring at an incoherent output, wondering why.

출처: Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

# Grad\_CAM

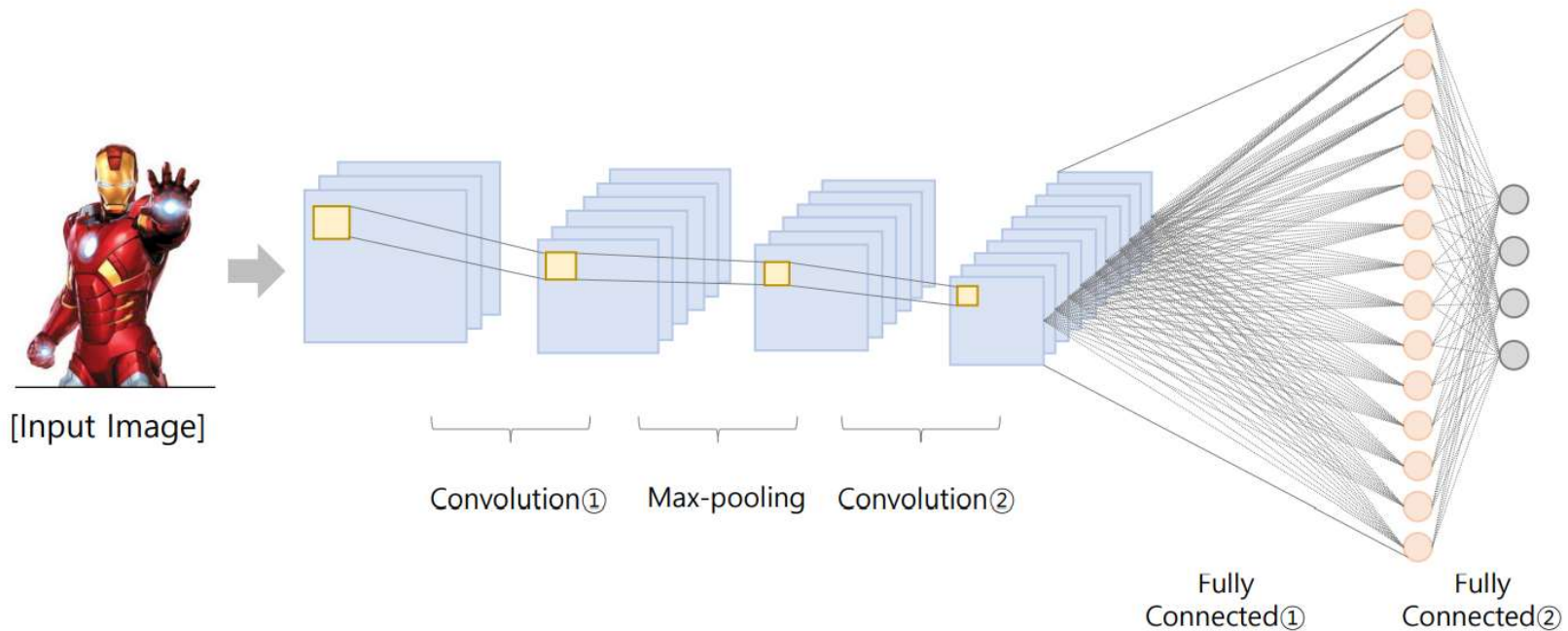


## - CNN + Class Activation Map(CAM) 구조

- 마지막 Convolution Layer 뒤에 Global Average Pooling 구조를 사용
- CNN에서 꼭 GAP를 사용해야하는 한계 발생



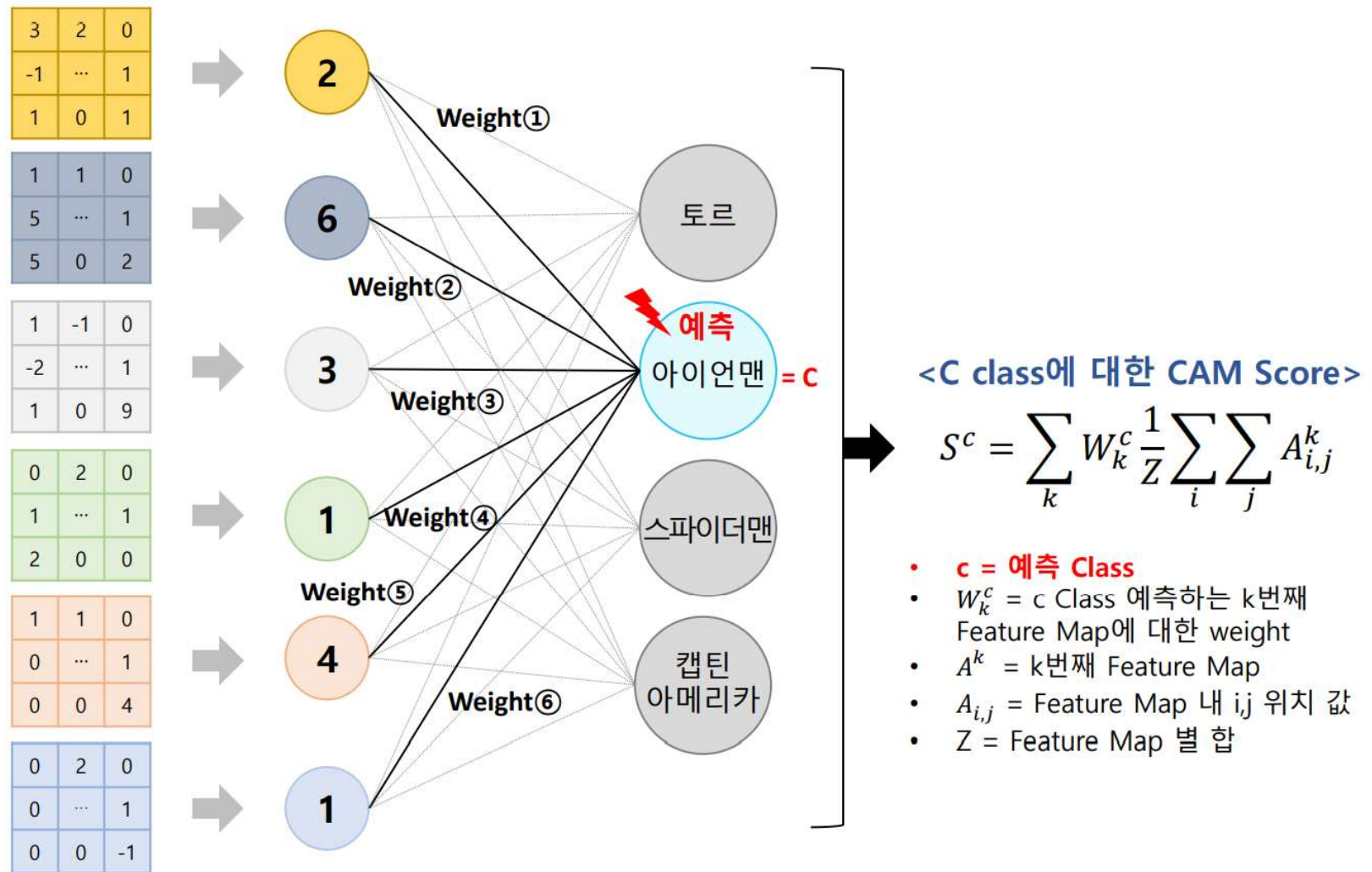
# Grad\_CAM



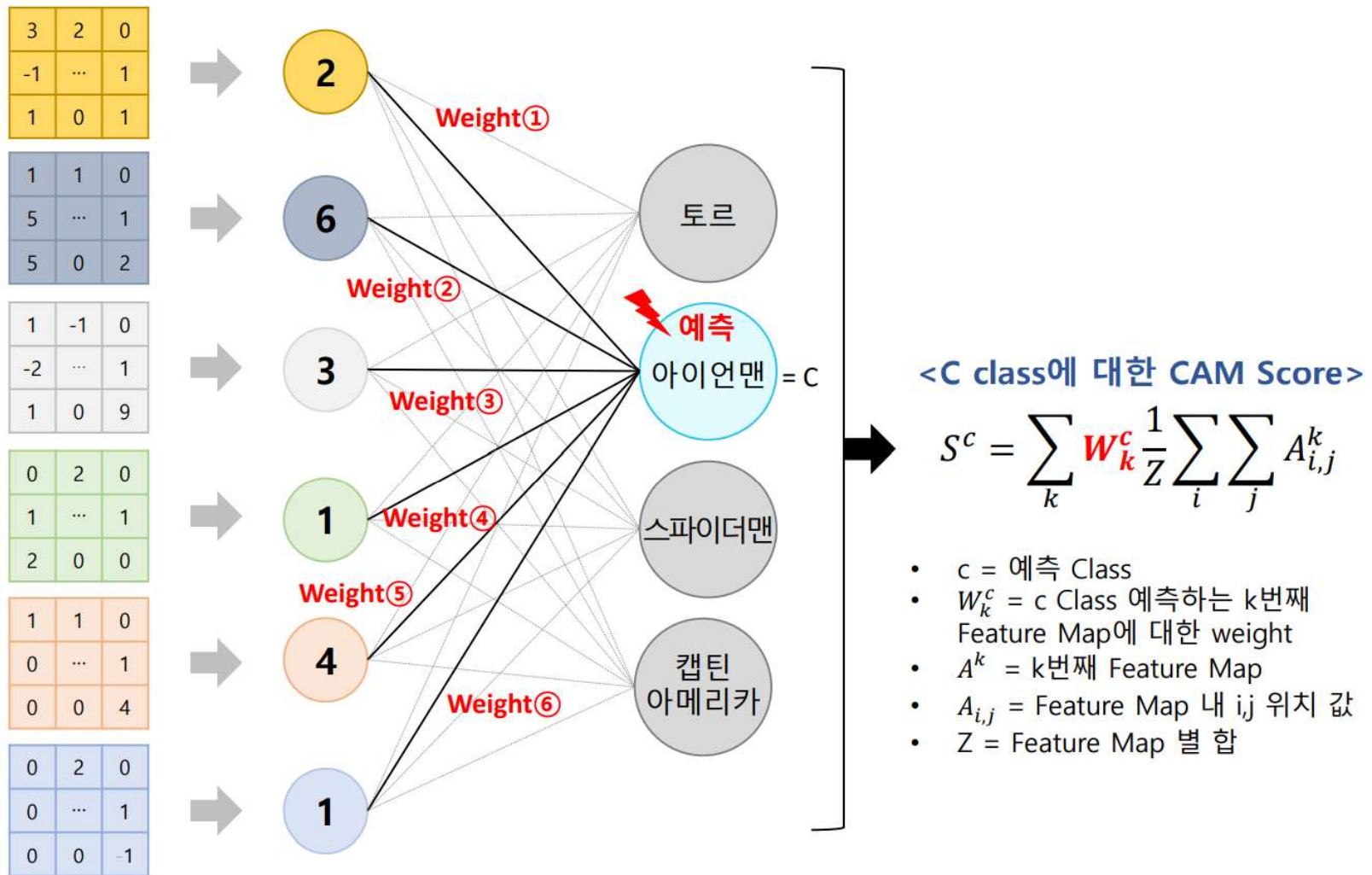
## - CNN + Grad\_CAM구조

- CAM 마지막 Convolution Layer 뒤 GAP 사용을 하지 않음
- CNN 기본 구조를 변형하지 않고 그대로 사용.

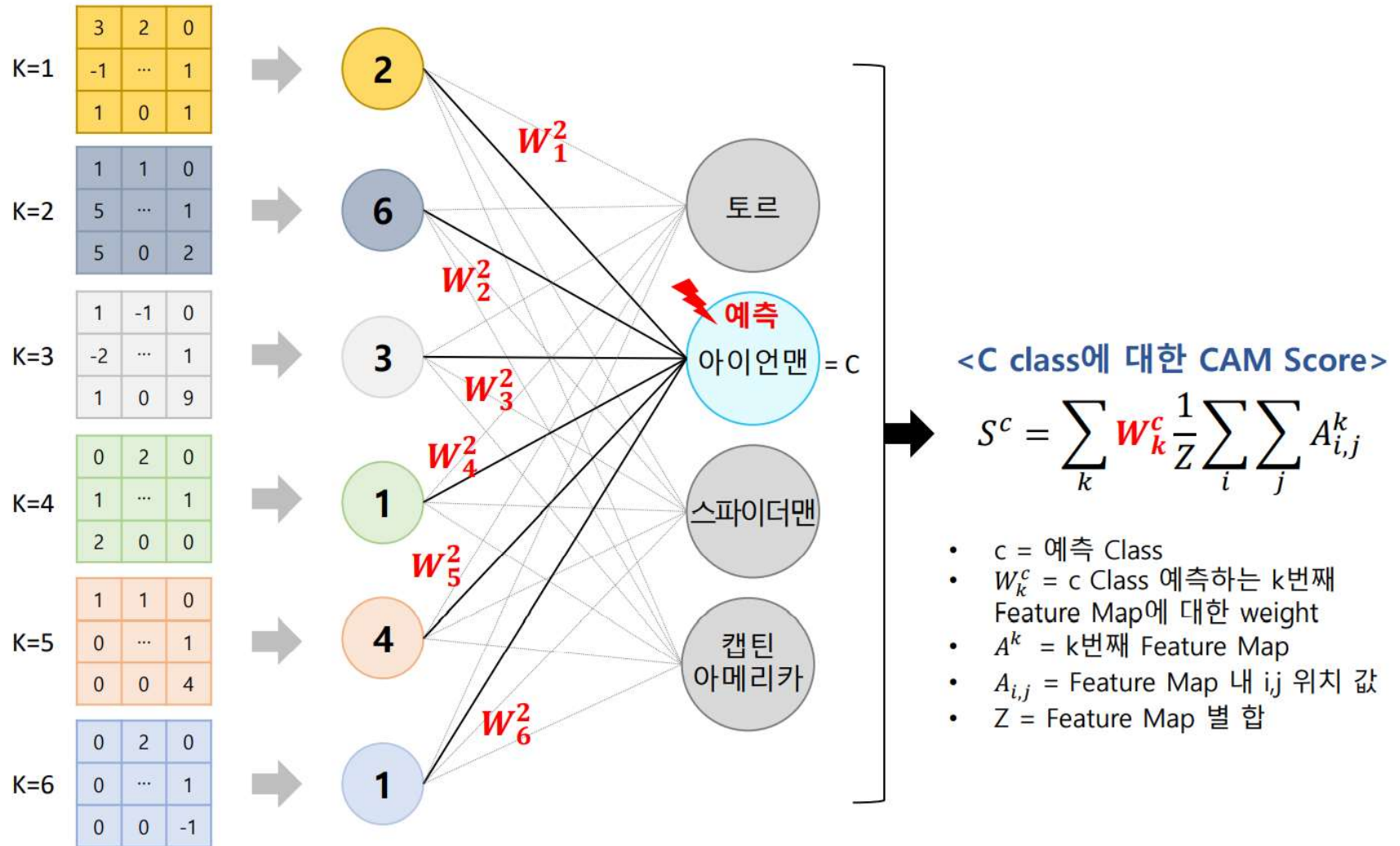
# Grad\_CAM



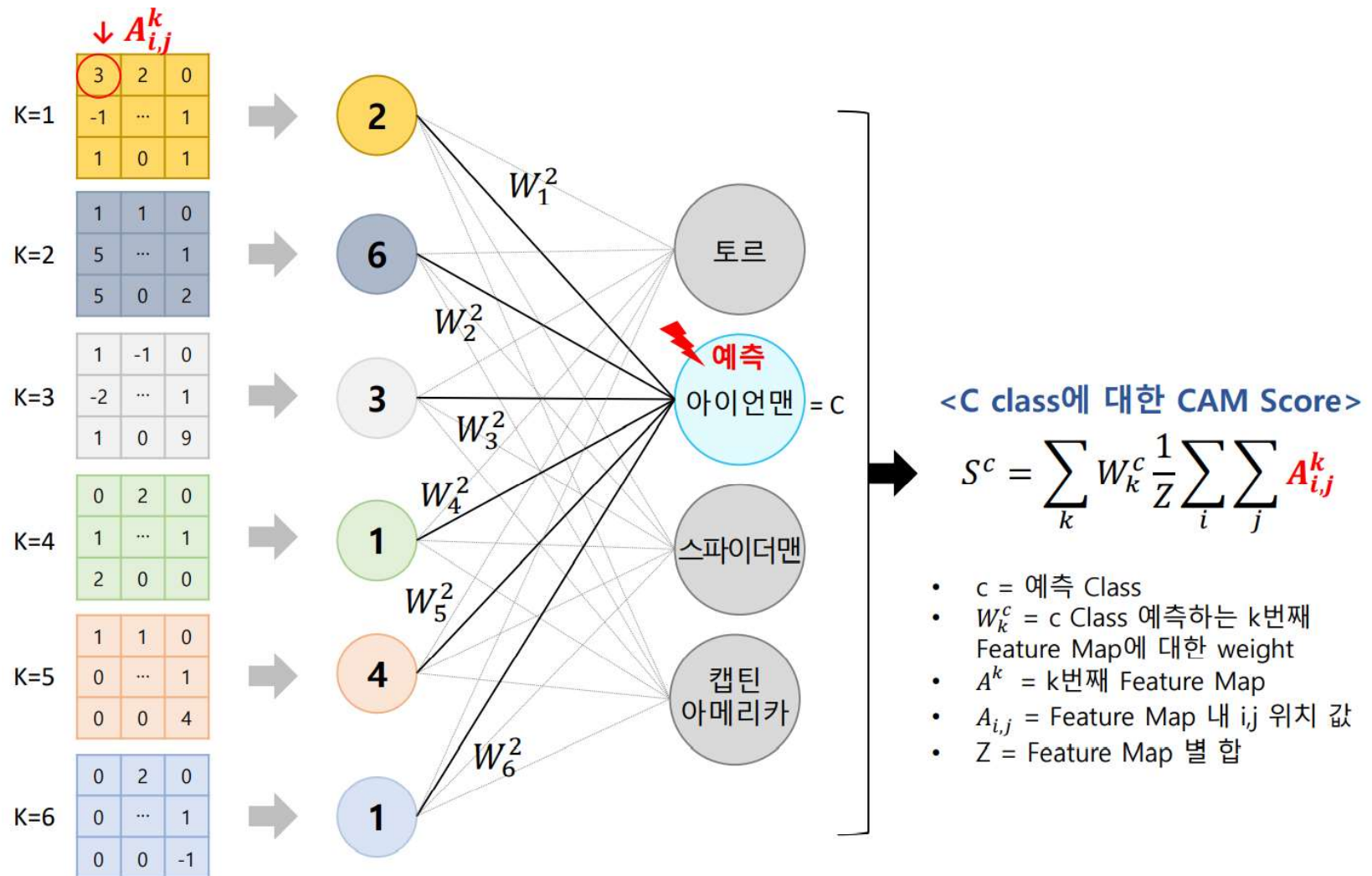
# Grad\_CAM



# Grad\_CAM

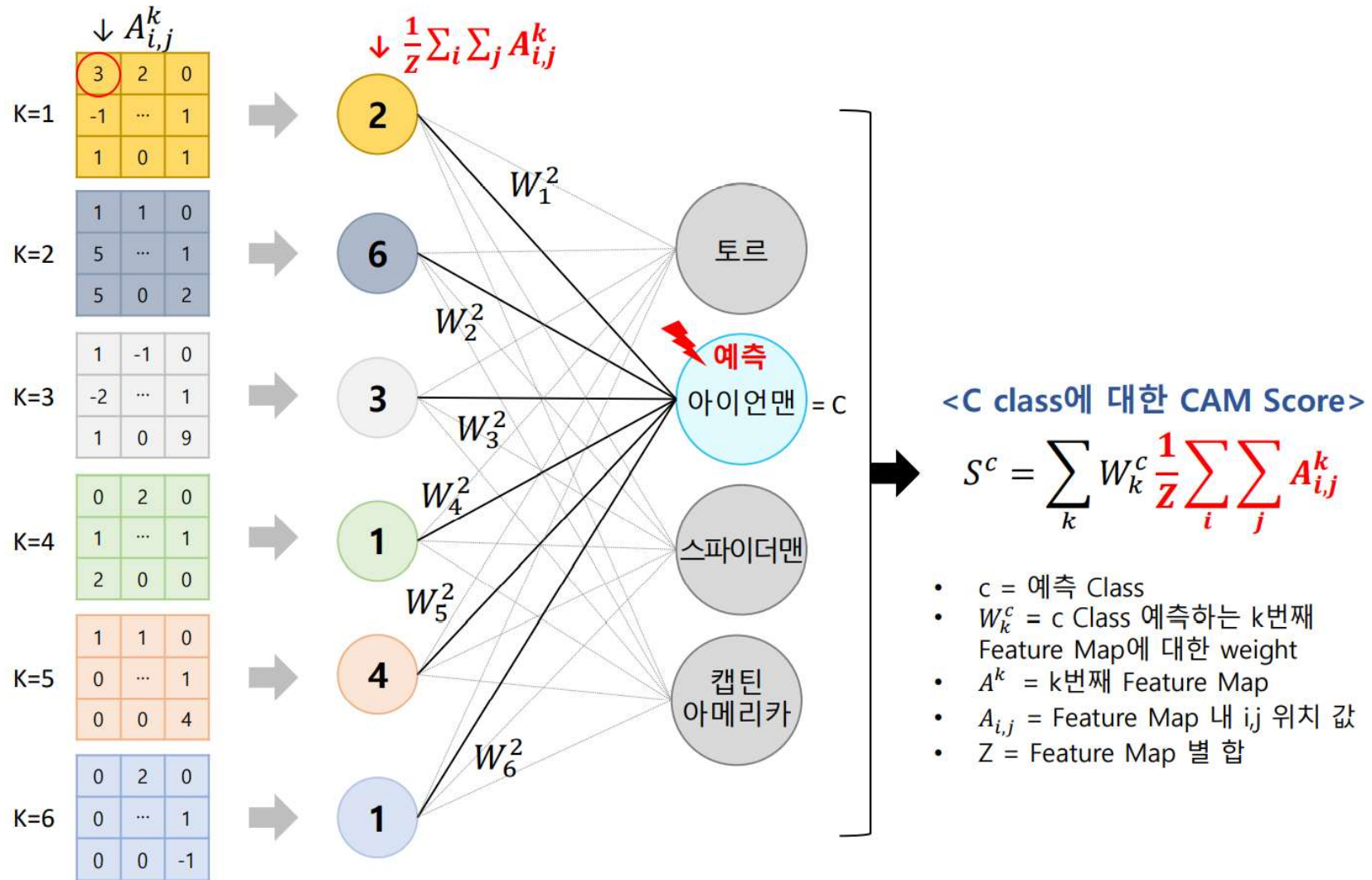


# Grad\_CAM

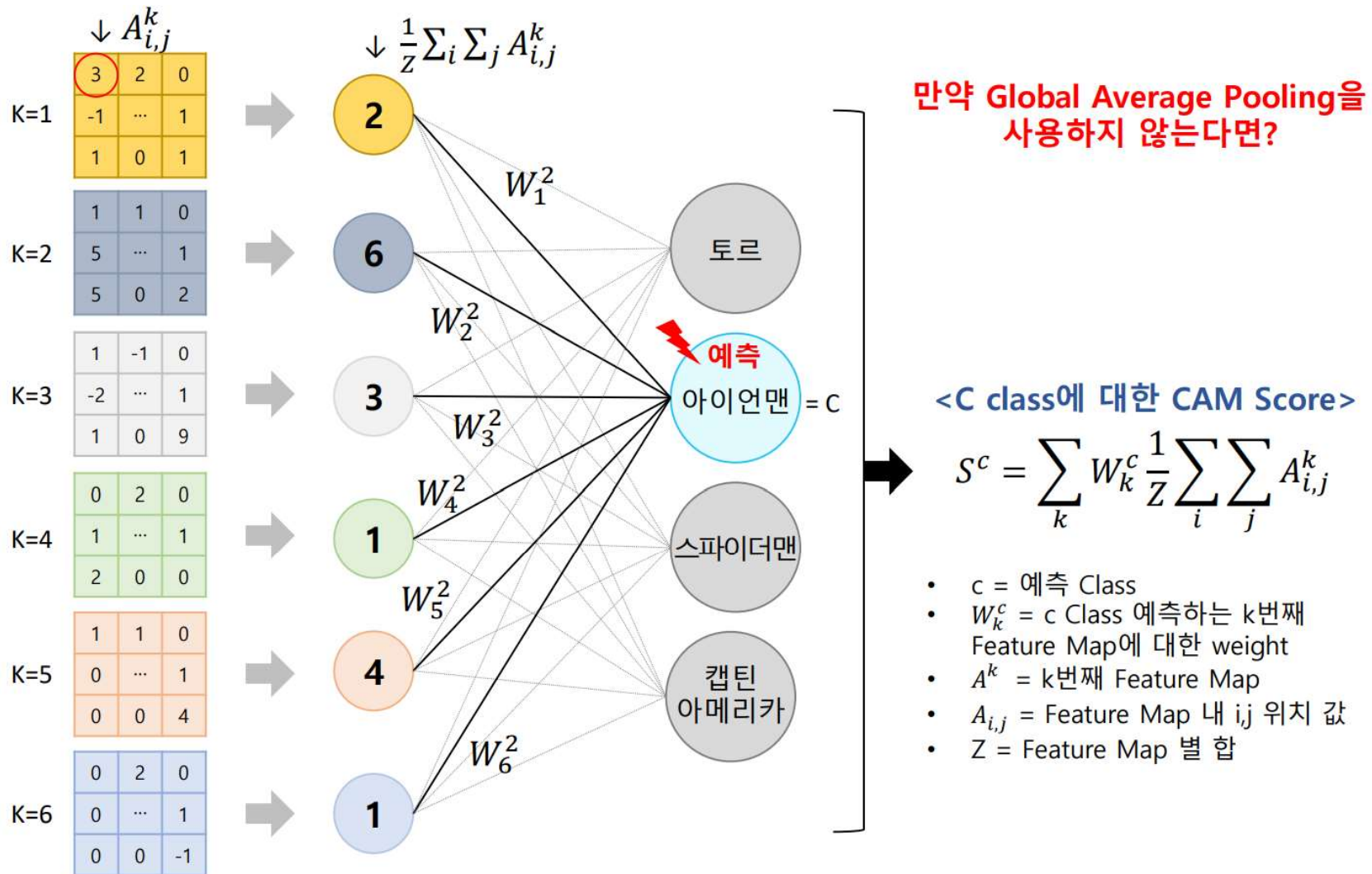




# Grad\_CAM



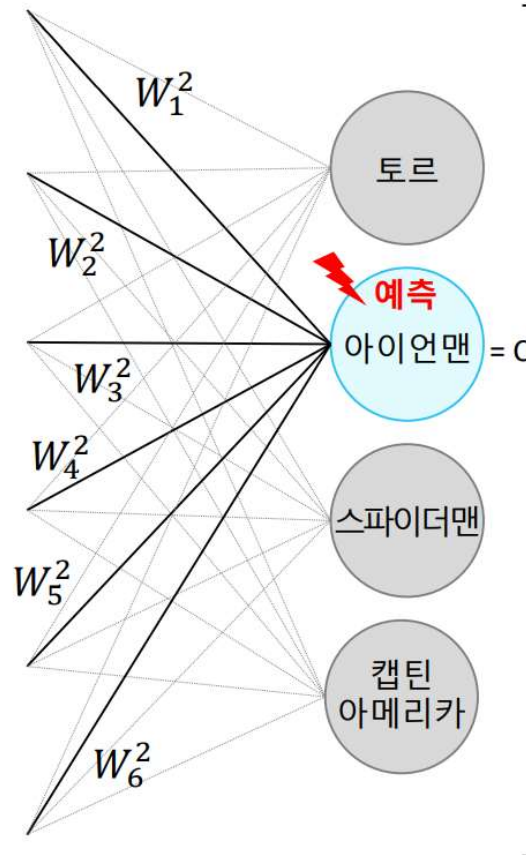
# Grad\_CAM



# Grad\_CAM

↓  $A_{i,j}^k$

K=1	<div>3</div>	2	0
	-1	...	1
	1	0	1
K=2	1	1	0
	5	...	1
	5	0	2
K=3	1	-1	0
	-2	...	1
	1	0	9
K=4	0	2	0
	1	...	1
	2	0	0
K=5	1	1	0
	0	...	1
	0	0	4
K=6	0	2	0
	0	...	1
	0	0	-1



만약 **Global Average Pooling**을 사용하지 않는다면?

- Feature Map 별 Weight도 사용하지 못함
- Weight를 정의하는 방식을 바꾸자

<C class에 대한 CAM Score>

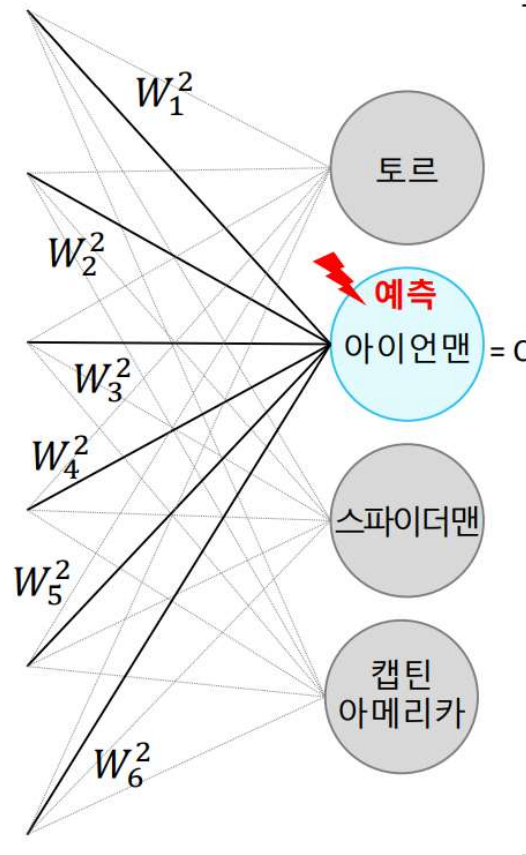
$$S^c = \sum_k \underbrace{W_k^c}_{???} \frac{1}{Z} \sum_i \sum_j A_{i,j}^k$$

- $c$  = 예측 Class
- $W_k^c$  =  $c$  Class 예측하는  $k$ 번째 Feature Map에 대한 weight
- $A^k$  =  $k$ 번째 Feature Map
- $A_{i,j}$  = Feature Map 내  $i,j$  위치 값
- $Z$  = Feature Map 별 합

# Grad\_CAM

↓  $A_{i,j}^k$

K=1	<div>3</div>	2	0
	-1	...	1
	1	0	1
K=2	1	1	0
	5	...	1
	5	0	2
K=3	1	-1	0
	-2	...	1
	1	0	9
K=4	0	2	0
	1	...	1
	2	0	0
K=5	1	1	0
	0	...	1
	0	0	4
K=6	0	2	0
	0	...	1
	0	0	-1



만약 **Global Average Pooling**을 사용하지 않는다면?

- Feature Map 별 Weight도 사용하지 못함
- Weight를 정의하는 방식을 바꾸자

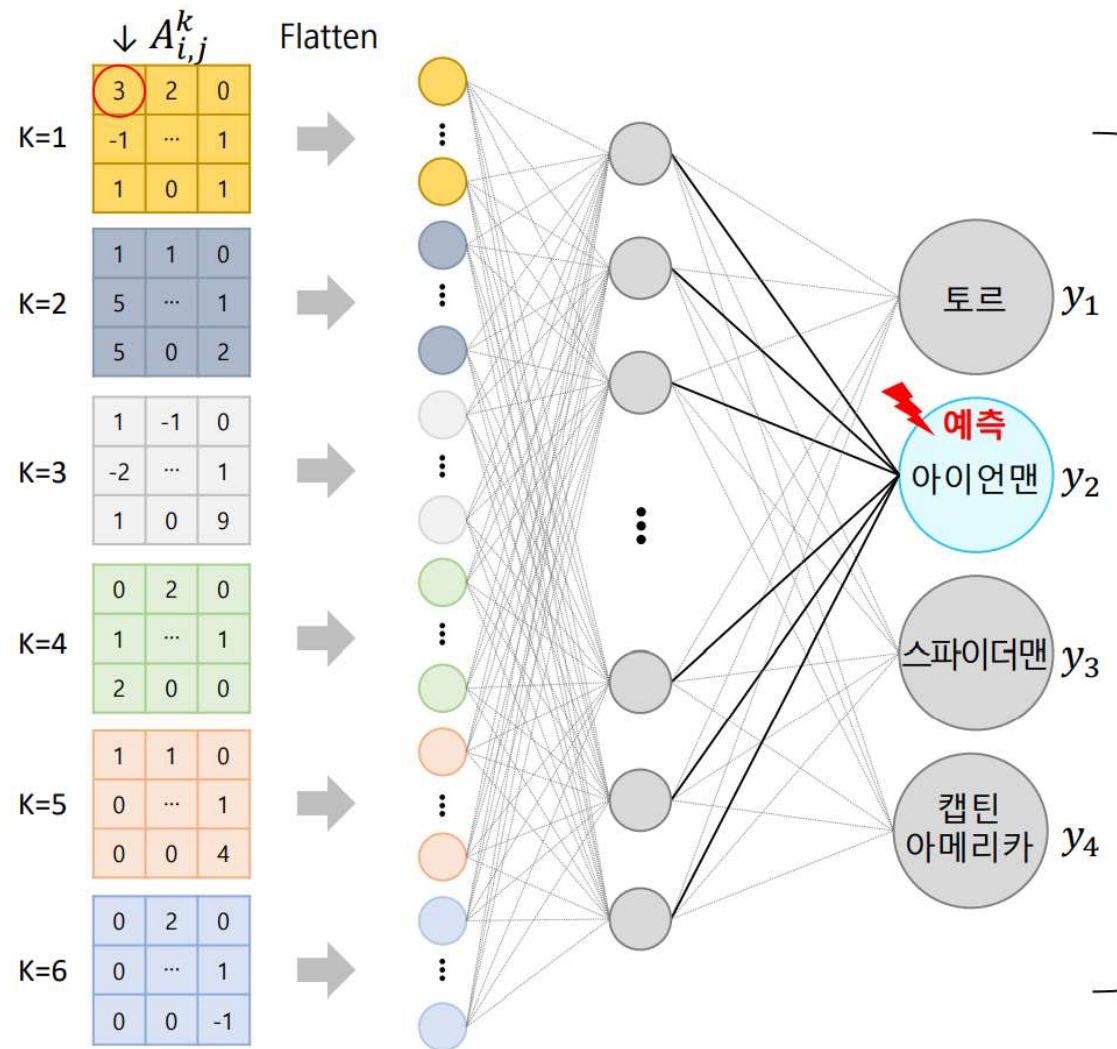
<C class에 대한 CAM Score>

$$S^c = \sum_k \underbrace{W_k^c}_{???} \frac{1}{Z} \sum_i \sum_j A_{i,j}^k$$

- $c$  = 예측 Class
- $W_k^c$  =  $c$  Class 예측하는  $k$ 번째 Feature Map에 대한 weight
- $A^k$  =  $k$ 번째 Feature Map
- $A_{i,j}$  = Feature Map 내  $ij$  위치 값
- $Z$  = Feature Map 별 합

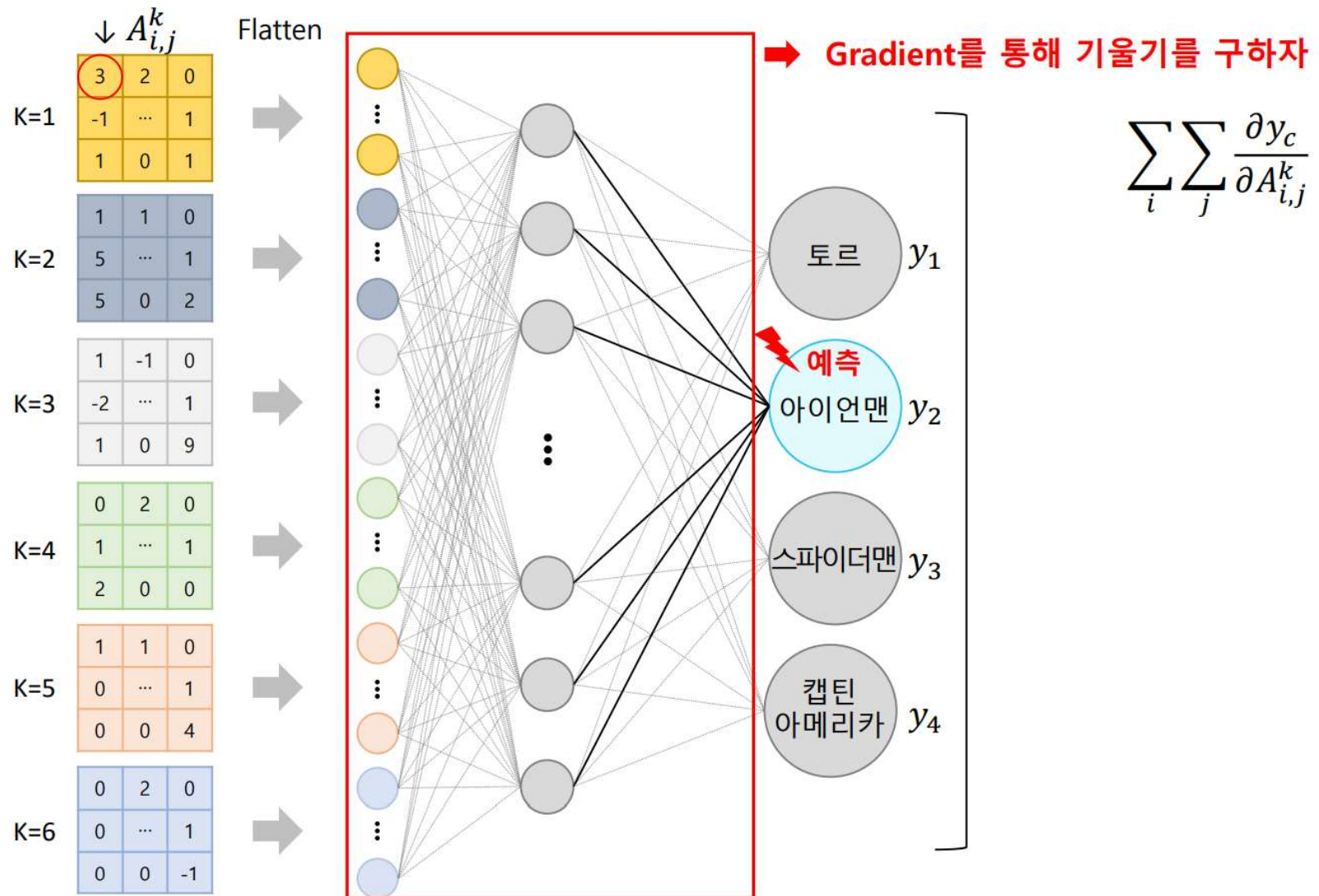


# Grad\_CAM

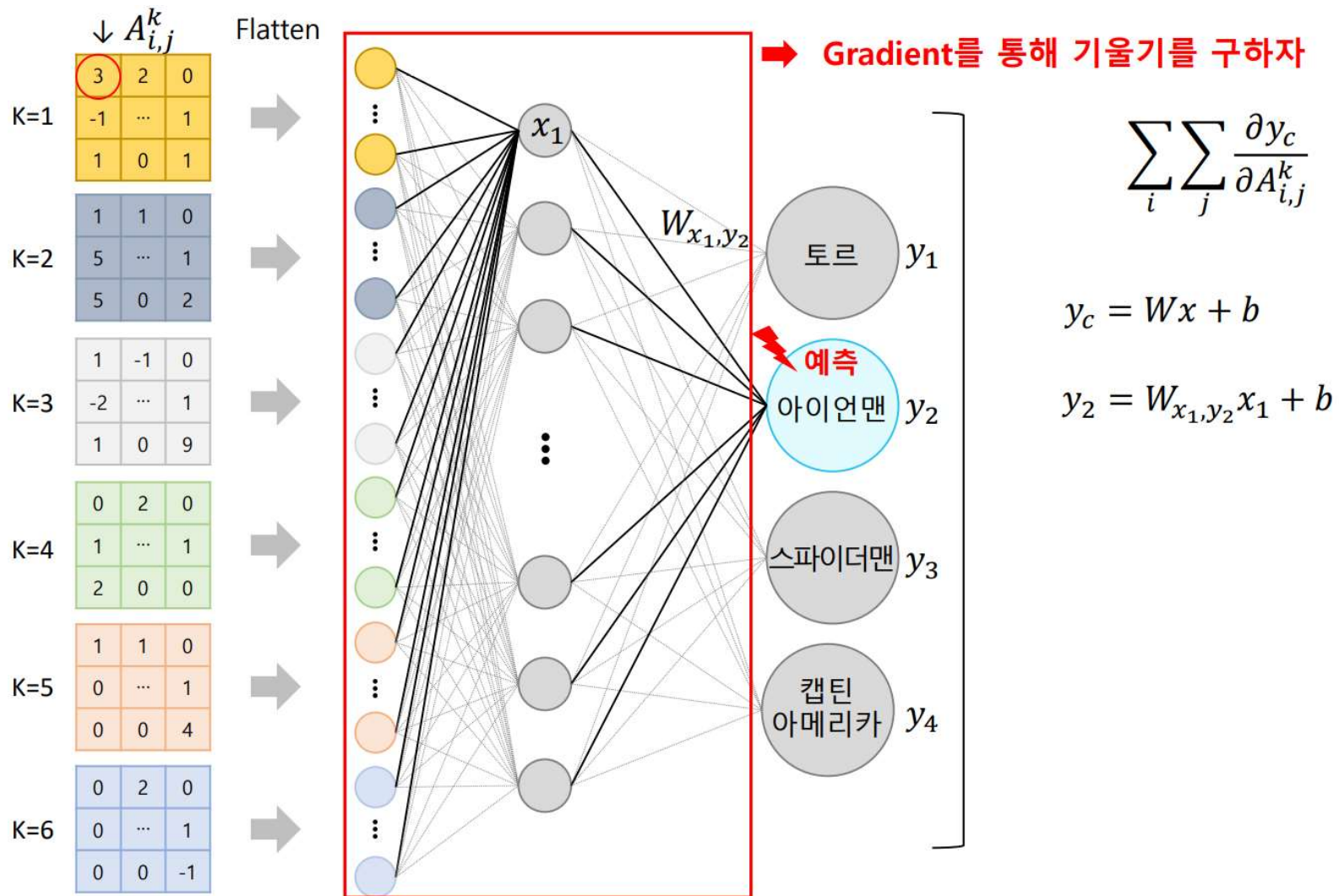




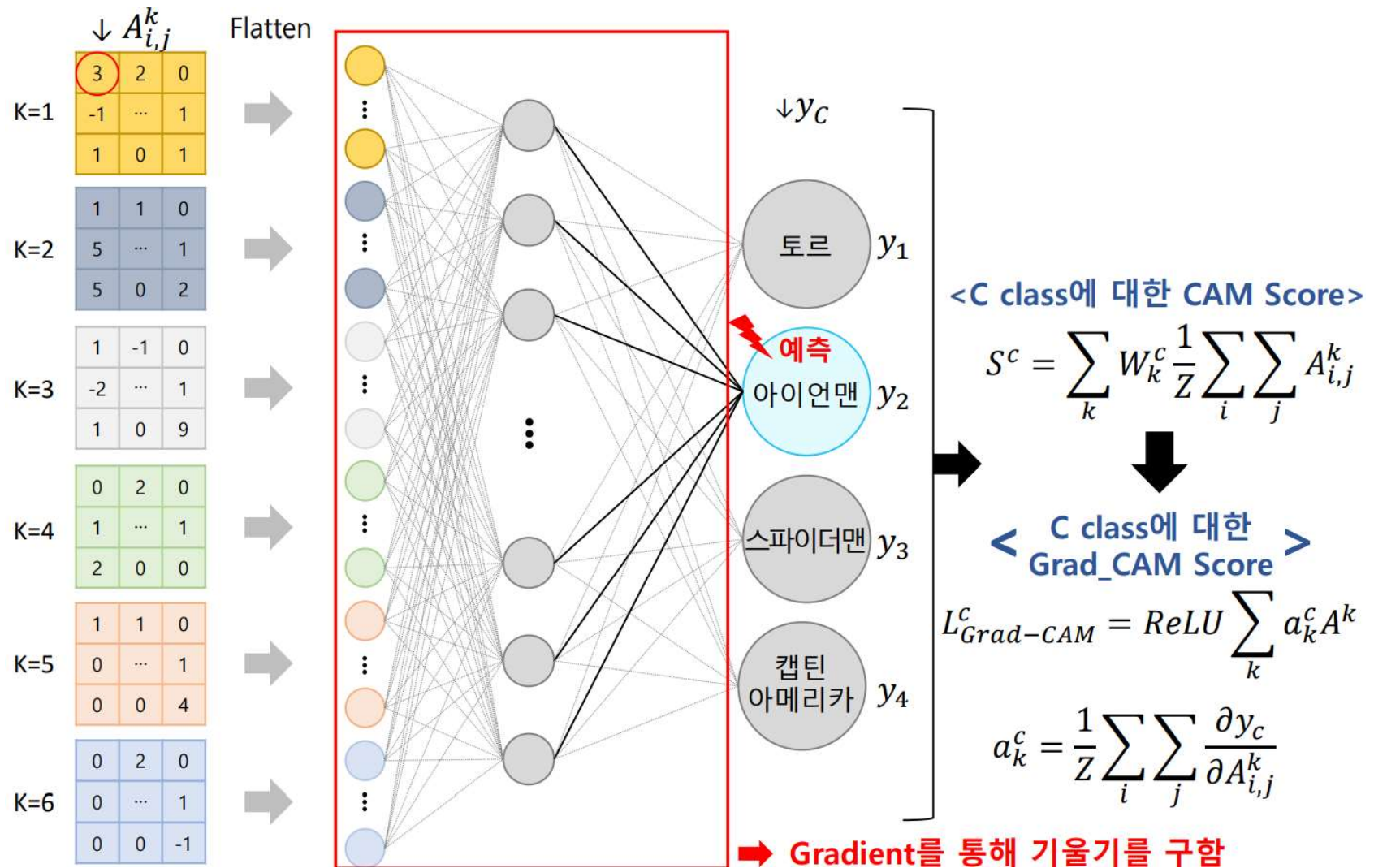
# Grad\_CAM



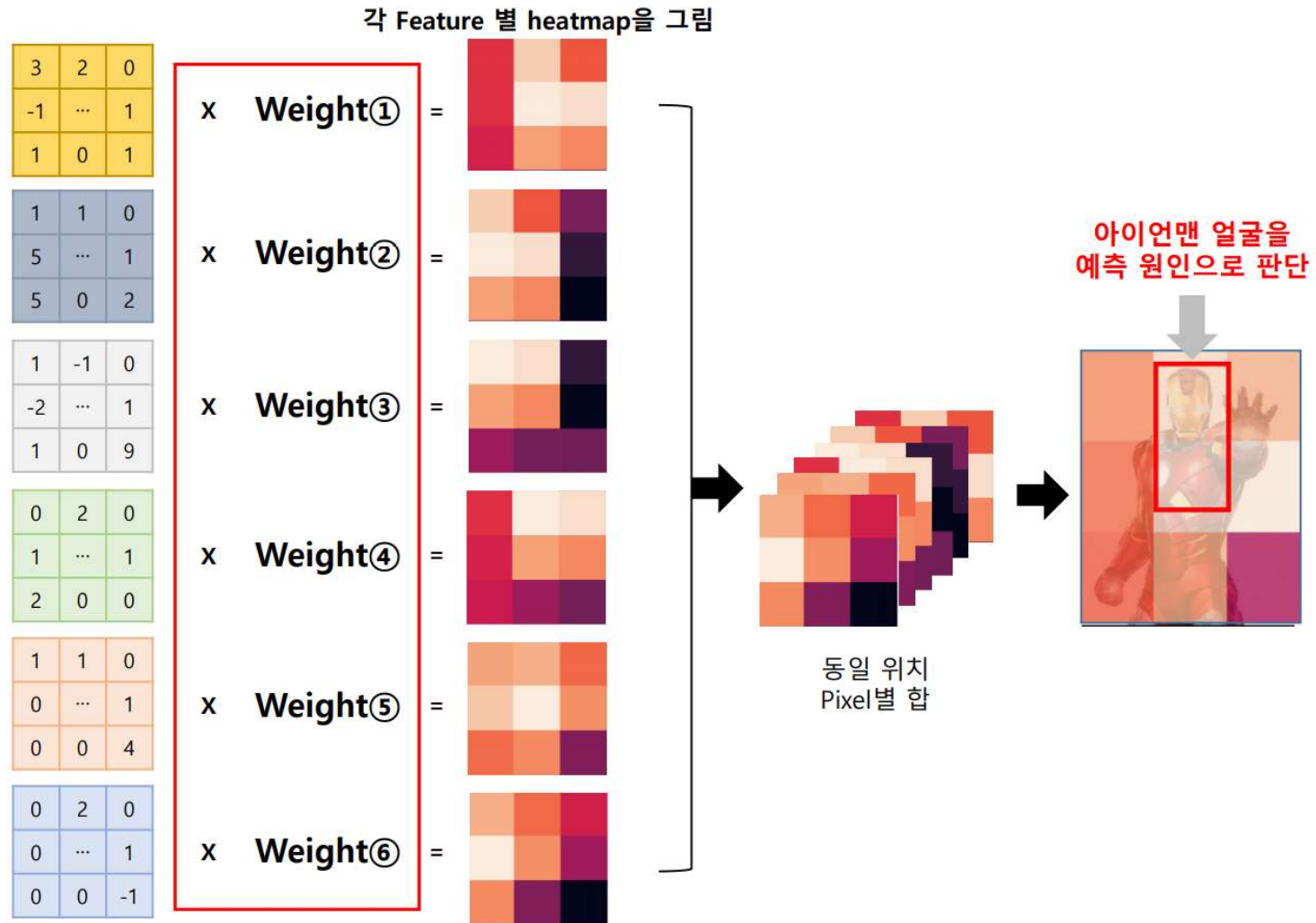
# Grad\_CAM



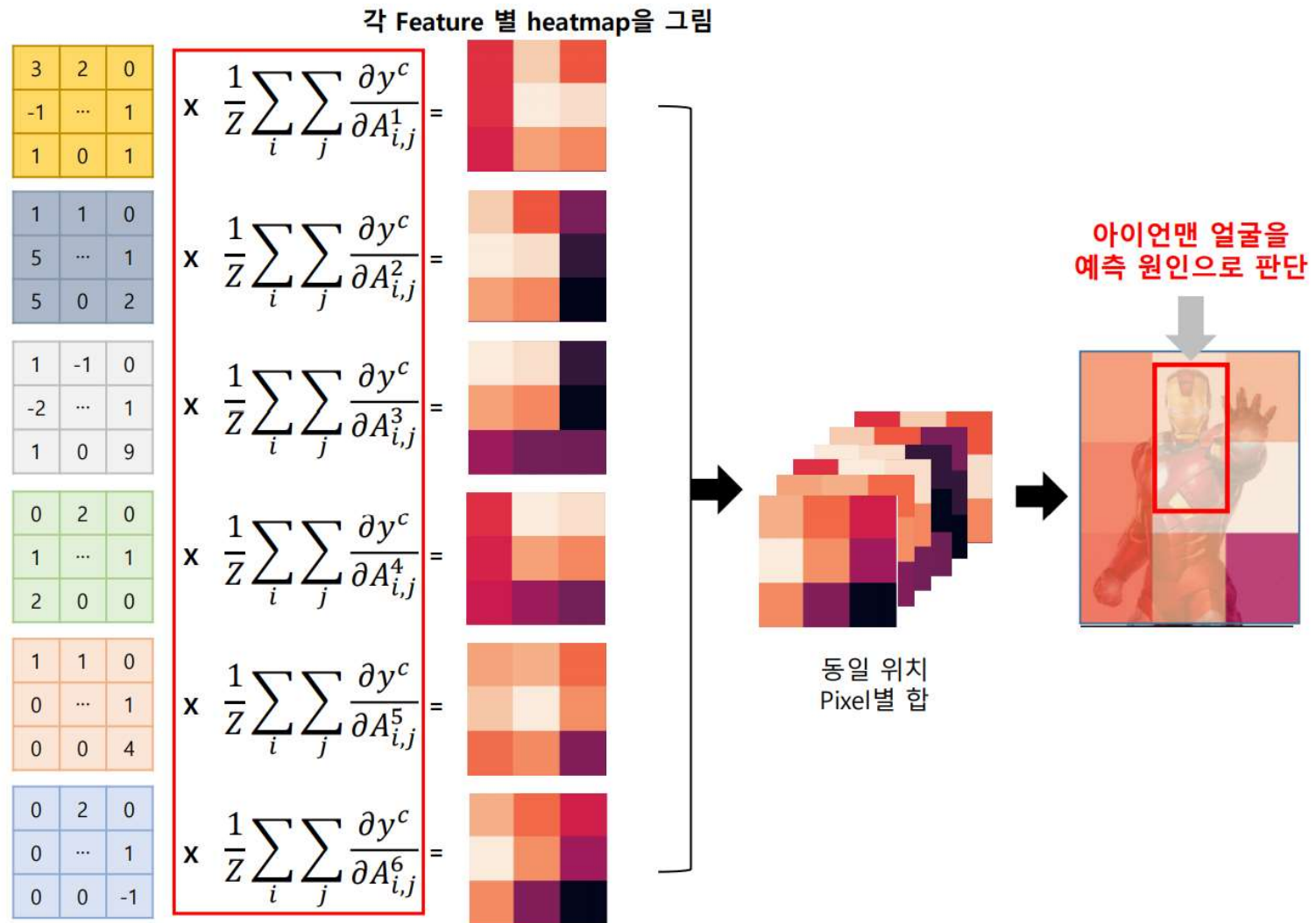
# Grad\_CAM



# Grad\_CAM

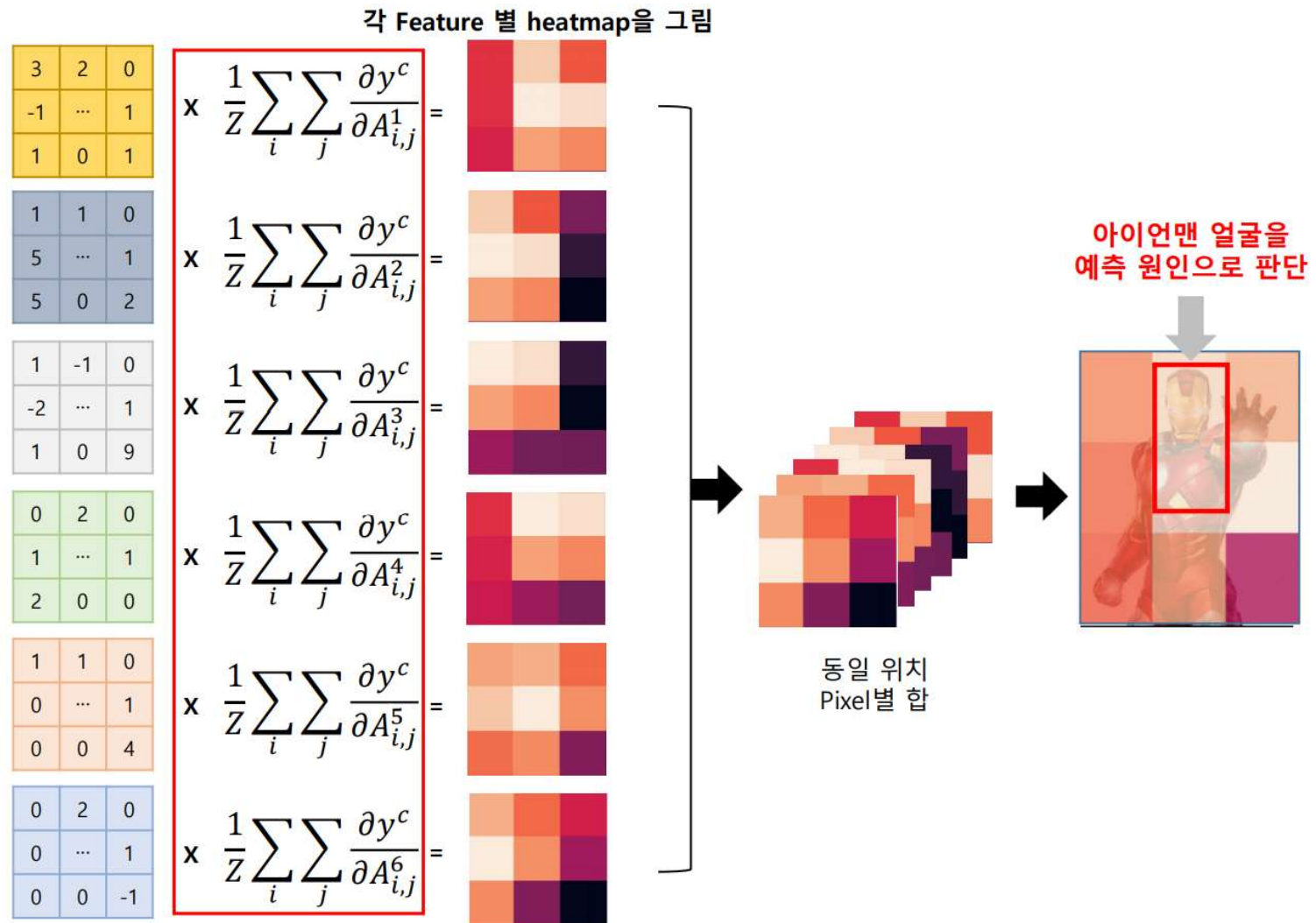


# Grad\_CAM





# Grad\_CAM



# Grad\_CAM

ResNet50 Grad-CAM: input\_1



## - Grad\_CAM 장점

- Gradient 로 Weight를 구하다 보니 마지막 Layer 아니아도 적용 가능

# 감사합니다.

Reference : <http://dmqm.korea.ac.kr/uploads/seminar>