Grad_CAM

발표자 : 백인성

2019.12.06.

Data Mining
Quality Analytics
hcai
Human-Centered Artificial Intelligence Center

목차

1. Introduction

2. Convolutional Neural Network(CNN)

3. Class Activation Map(CAM)

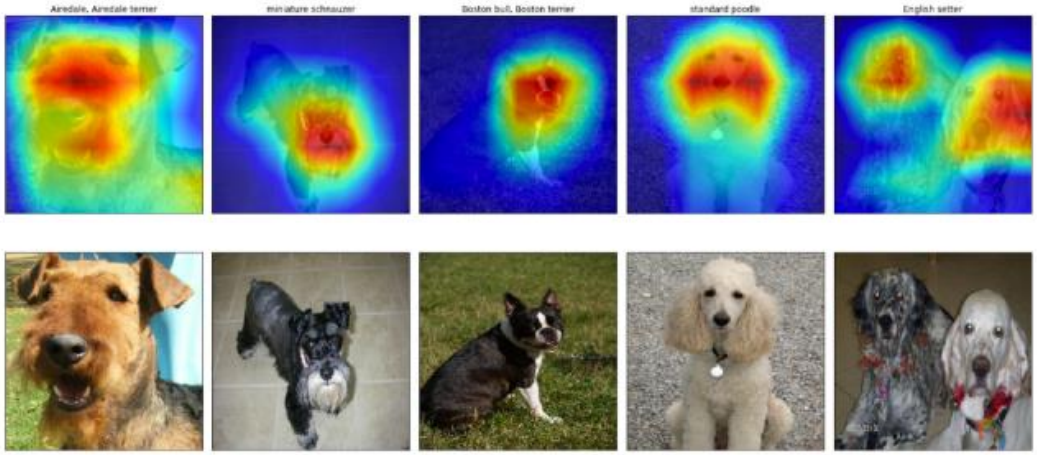
4. Grad_CAM

5. Conclusion

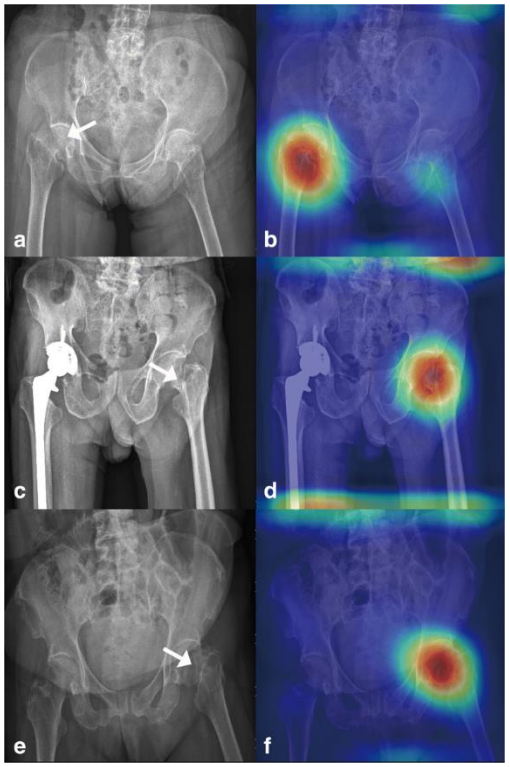
1. Introduction

Introduction

❖ 예측 모델에서 실제 CAM(Class Activation Map)을 사용한 사례



<분류 모델 원인 해석>
→ 예측 모델 결과에 대한 신뢰성



<고관절 골절 탐지>
→ 병 원인 진단

출처: https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization/
Cheng, Chi-Tung, et al. "Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs." European radiology (2019): 1-9.

Introduction

❖ 예측 모델의 해석이 필요한 이유는 아래의 3가지 이유로 요약

1. 예측 모델을 사용하는 현업자들에게 이해 시키기 쉽게

2. 구축한 예측 모델 결과가 타당함을 직관적으로 보이기 위해

3. 예측 결과에 대한 원인을 분석하고 향후 대처 할 수 있게

Introduction

- Grad-CAM(Class Activation Map) pre-view
  - CNN 구조 모델에서 Gradient를 활용해 예측 결과에 대한 원인 해석
  - CNN, 시각화, 컨셉의 핵심에 대해서 순차적으로 설명을 진행

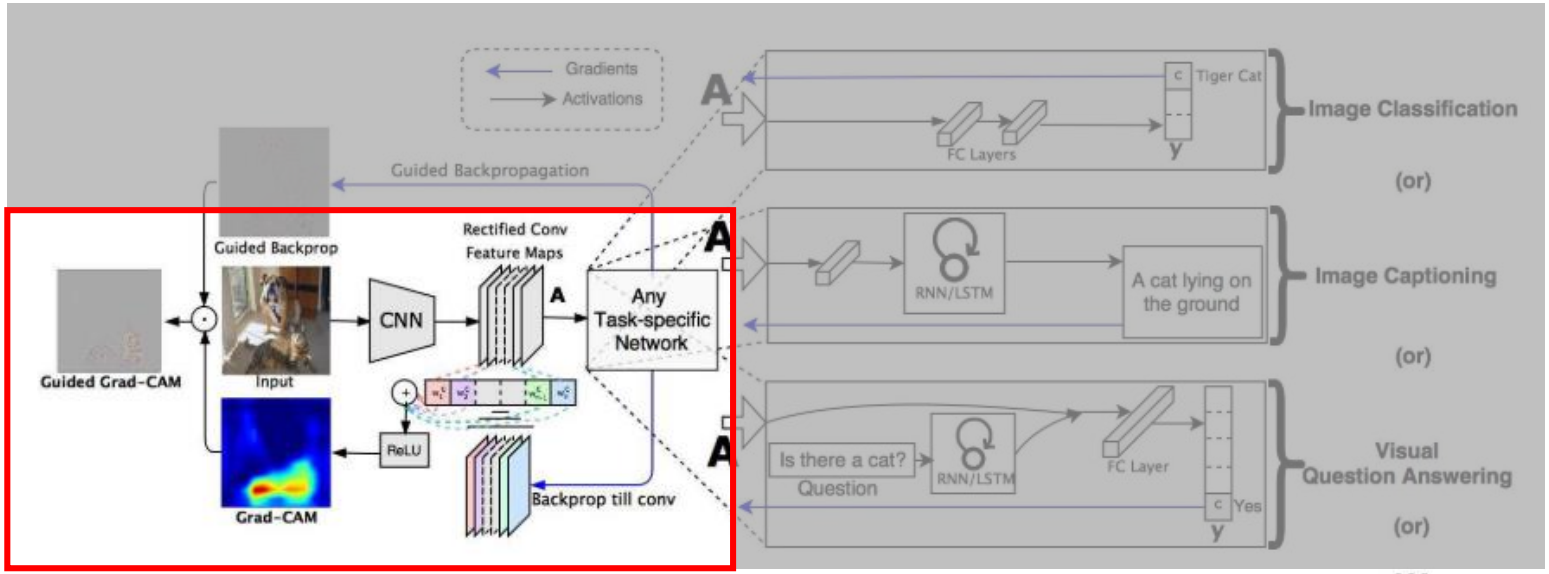
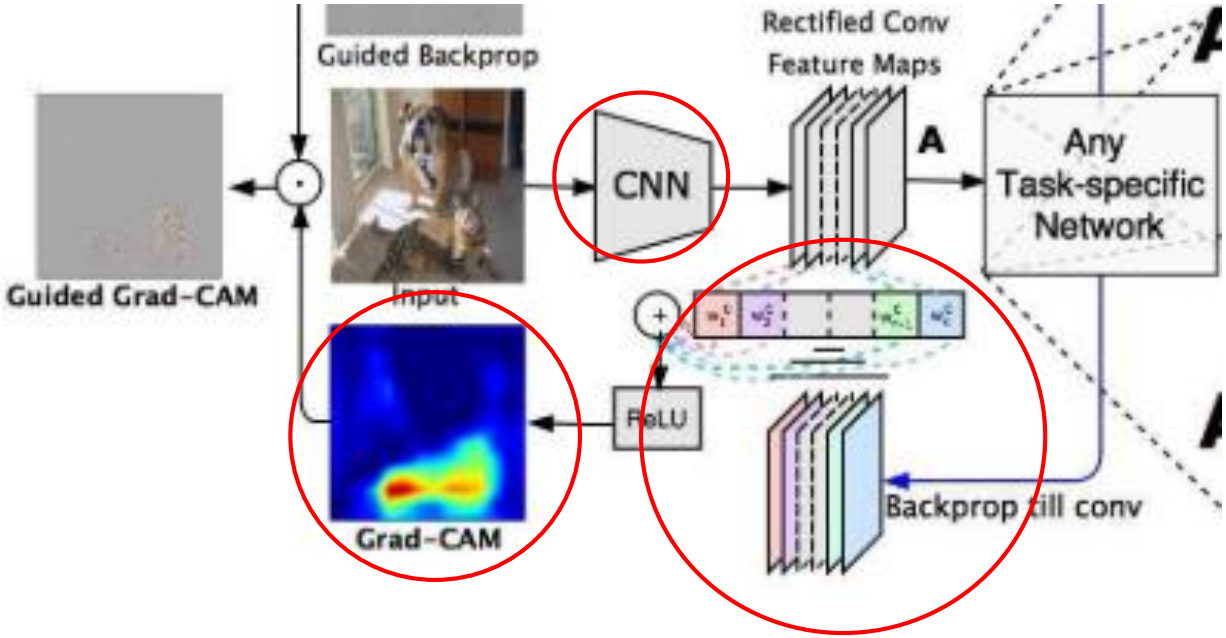


Figure 2: Grad-CAM overview: Given an image and a class of interest (*e.g.*, 'tiger cat' or any other type of differentiable output) as input, we forward propagate the image through the CNN part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to zero for all classes except the desired class (tiger cat), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which we combine to compute the coarse Grad-CAM localization (blue heatmap) which represents where the model has to look to make the particular decision. Finally, we pointwise multiply the heatmap with guided backpropagation to get Guided Grad-CAM visualizations which are both high-resolution and concept-specific.

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.

Introduction

- Grad-CAM(Class Activation Map) pre-view
  - CNN 구조 모델에서 Gradient를 활용해 예측 결과에 대한 원인 해석
  - CNN, 시각화, 컨셉의 핵심에 대해서 순차적으로 설명을 진행

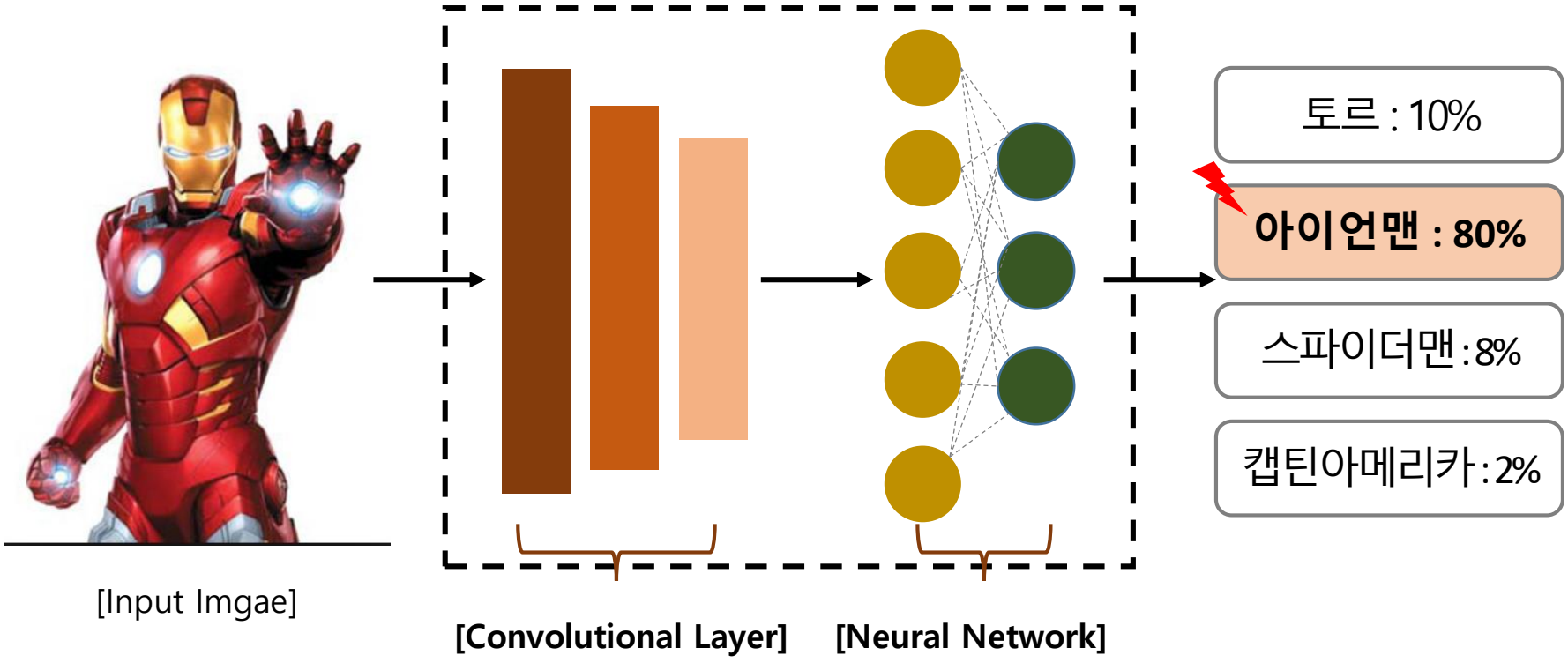


출처: Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

2. Convolutional Neural Network(CNN)

Convolutional Neural Network(CNN)

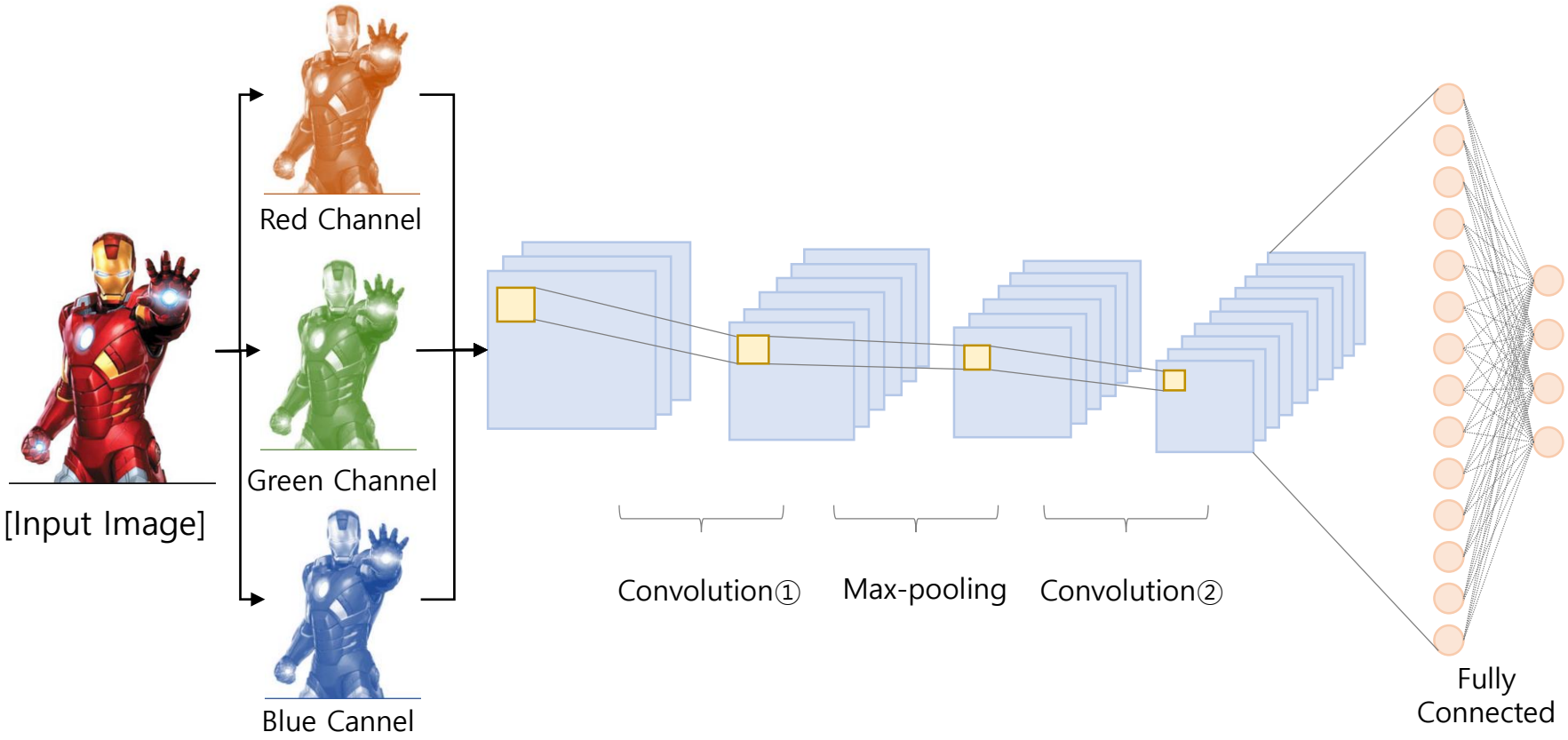
- Convolutional Neural Network 기본 구조
  - Neural Network에 Convolution Layer를 사용한 방법론
  - Object Detection, Classification 등 Visual Task에서 좋은 Performance 보임



Convolutional Neural Network(CNN)

❖ Convolutional Neural Network 세부 구조

- ➢ Input Image에 대해 Convolution & Pooling을 활용하여 특징 추출
- ➢ 분류 모델에서는 Input Image가 어떤 Class에 속하는지 예측함



Convolutional Neural Network(CNN)

❖ Filter & Convolution

- ➢ Filter : Input의 특징을 찾아내기 위한 공용 parameter
- ➢ Convolution(합성곱) : Filter를 이동 하면서 곱한 결과를 합산한 결과

Red Channel

0	0	2	1	1
0	1	1	2	2
0	3	2	1	0
1	1	1	3	0
0	1	1	1	0

*

1	0	2
0	-1	1
-1	1	2

11	4	3
3	10	6
7	9	-1

Green Channel

1	1	2	2	1
0	0	1	3	0
0	1	1	0	0
2	2	1	3	1
0	0	1	1	0

*

0	0	1
0	2	-1
0	1	0

2	2	7
4	6	3
4	0	6

Blue Cannel

1	0	0	1	1
1	1	1	0	3
0	2	3	2	3
1	1	1	4	0
1	2	0	1	0

*

-1	1	-2
0	0	1
2	0	0

채널 별 Filter(3x3)

0	2	8
3	4	-2
-1	5	-7

Pixel값 * Filter

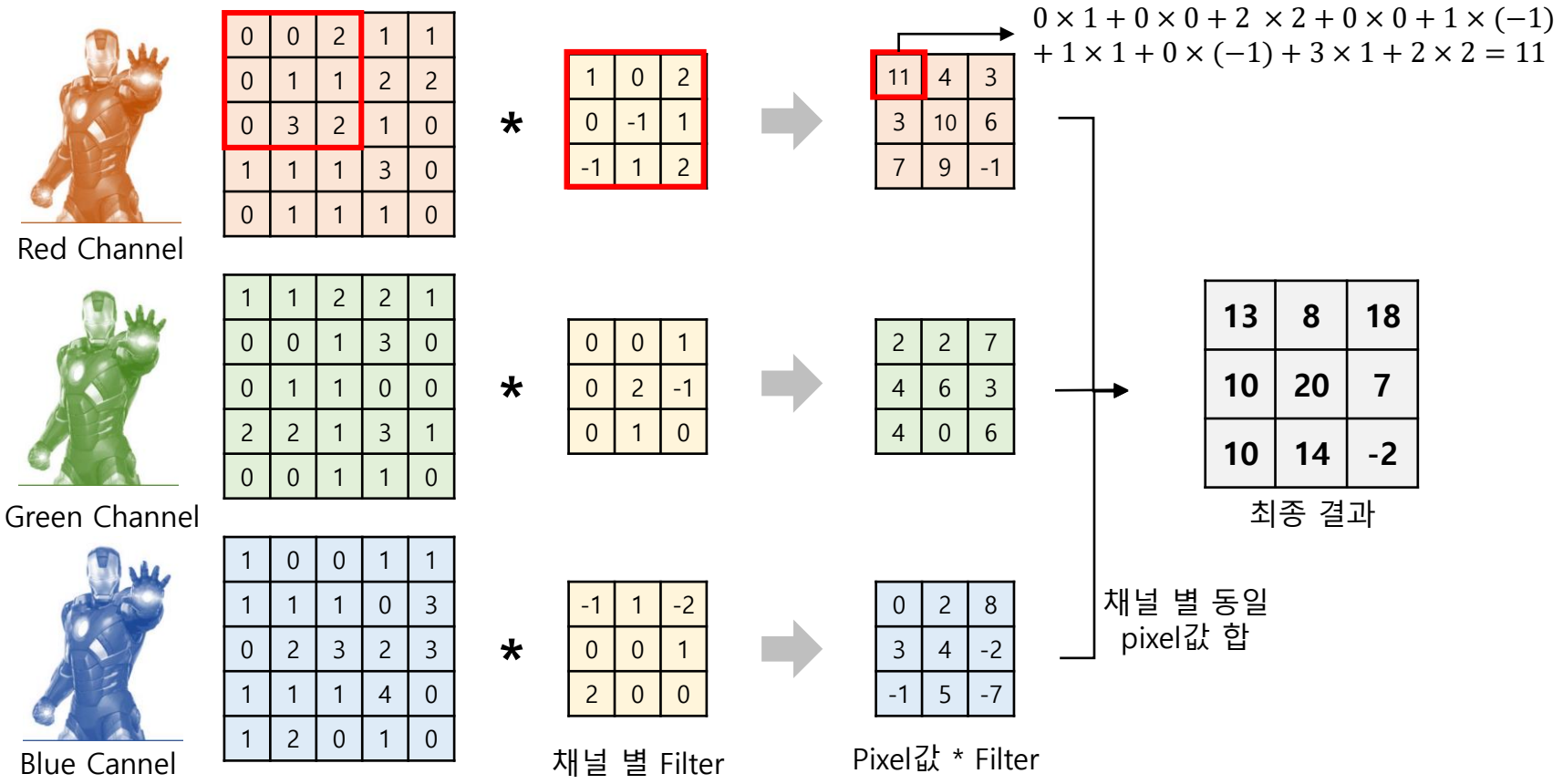
채널 별 동일 pixel값 합

13	**8**	**18**
10	**20**	**7**
10	**14**	**-2**

최종 결과

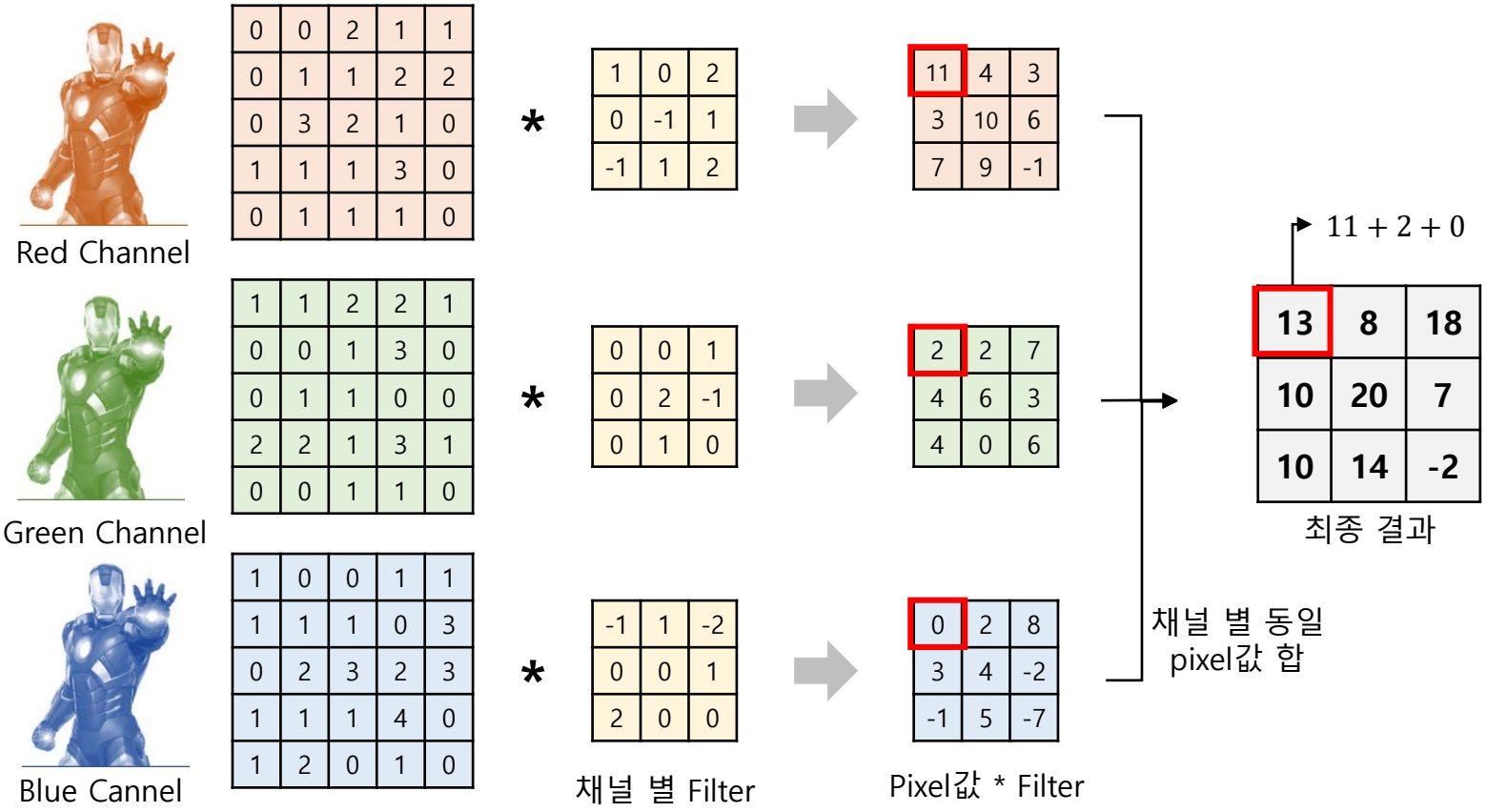
Convolutional Neural Network(CNN)

- Filter & Convolution
  - Filter : Input의 특징을 찾아내기 위한 공용 parameter
  - Convolution(합성곱) : Filter를 이동 하면서 곱한 결과를 합산한 결과



Convolutional Neural Network(CNN)

- ❖ Filter & Convolution
  - ➢ Filter : Input의 특징을 찾아내기 위한 공용 parameter
  - ➢ Convolution(합성곱) : Filter를 이동 하면서 곱한 결과를 합산한 결과

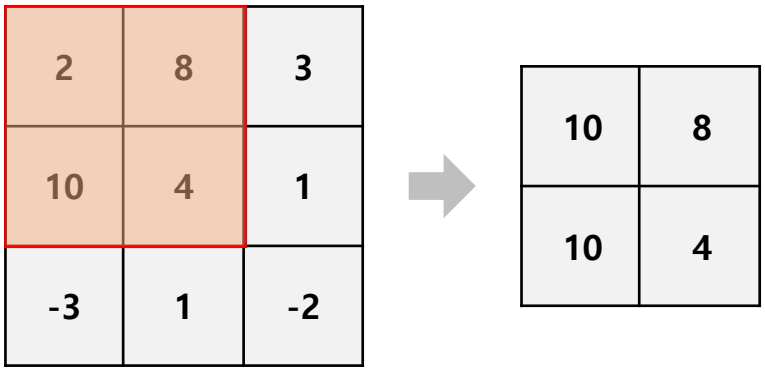


Convolutional Neural Network(CNN)

❖ Pooling(Max, Average)

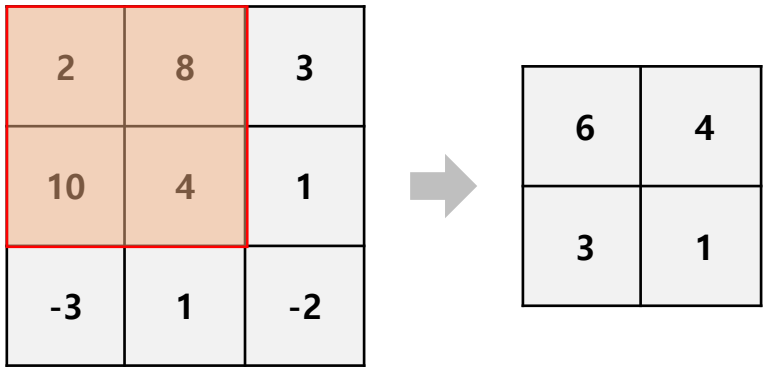
➢ Pooling : Convolution Layer의 출력 값을 입력으로 받아 크기를 줄이거나 특정 pixel 값을 강조하는 용도로 사용됨, Filter와 달리 학습 대상이 아님

<Max Pooling>



→ Pooling 크기 내 **최대값**으로 요약

<Average Pooling>



→ Pooling 크기 내 **평균값**으로 요약

3. Class Activation Map(CAM)

Class Activation Map(CAM)

- Class Activation Map(CAM)(2017)
  - 딥러닝 프레임 워크에서 예측 원인을 파악하기 위해 등장한 알고리즘
  - 2016년도 CVPR(Computer Vision and Pattern Recognition)에서 등장

Learning deep features for discriminative localization

B Zhou, A Khosla, A Lapedriza, A Oliva... - Proceedings of the ..., 2016 - cv-foundation.org

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 image. Despite the apparent simplicity of global average pooling, we are able to achieve ...

☆ 99 1606회 인용 관련 학술자료 전체 16개의 버전 »

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

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[1].

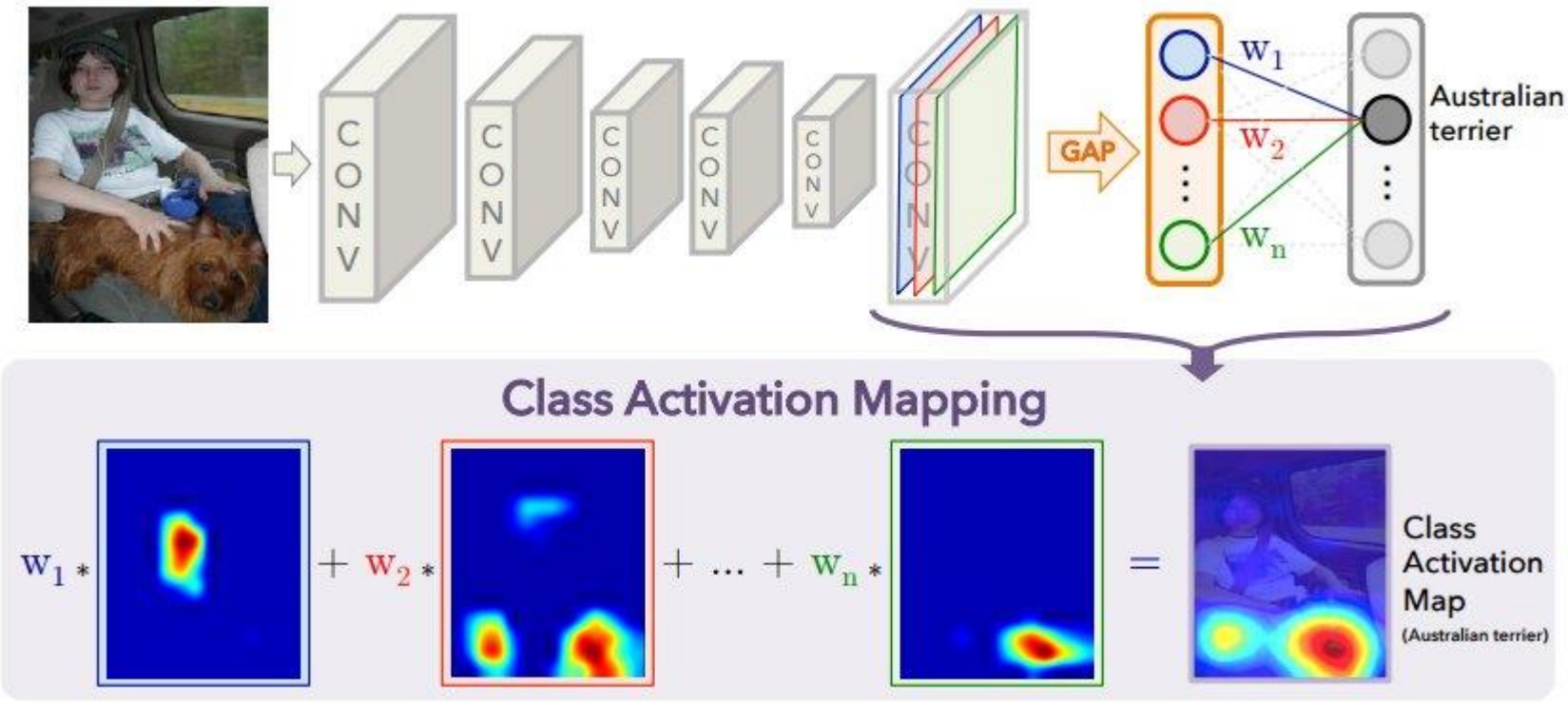
Figure 1. A simple modification of the global average pooling layer combined with our class activation mapping (CAM) technique allows the classification-trained CNN to both classify the image and localize class-specific image regions in a single forward-pass e.g., the toothbrush for *brushing teeth*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

출처: Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Class Activation Map(CAM)

❖ Class Activation Map 구조

- ➢ CNN 모델로 예측 시, 어떤 부분이 Class 예측에 큰 영향을 주었는지 확인 가능
- ➢ 마지막 Convolution Layer 이후 Global Average Pooling 사용

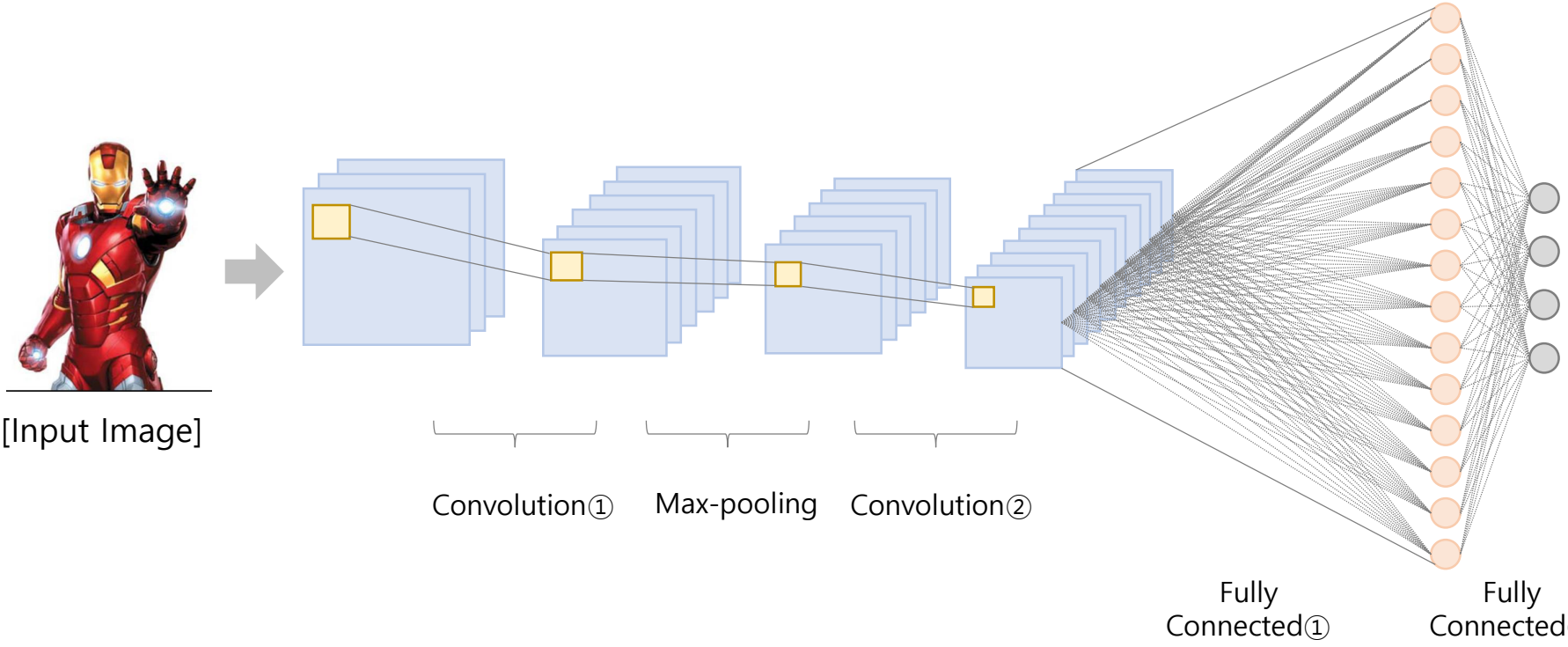


출처: Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Class Activation Map(CAM)

❖ Convolutional Neural Network(CNN) 구조

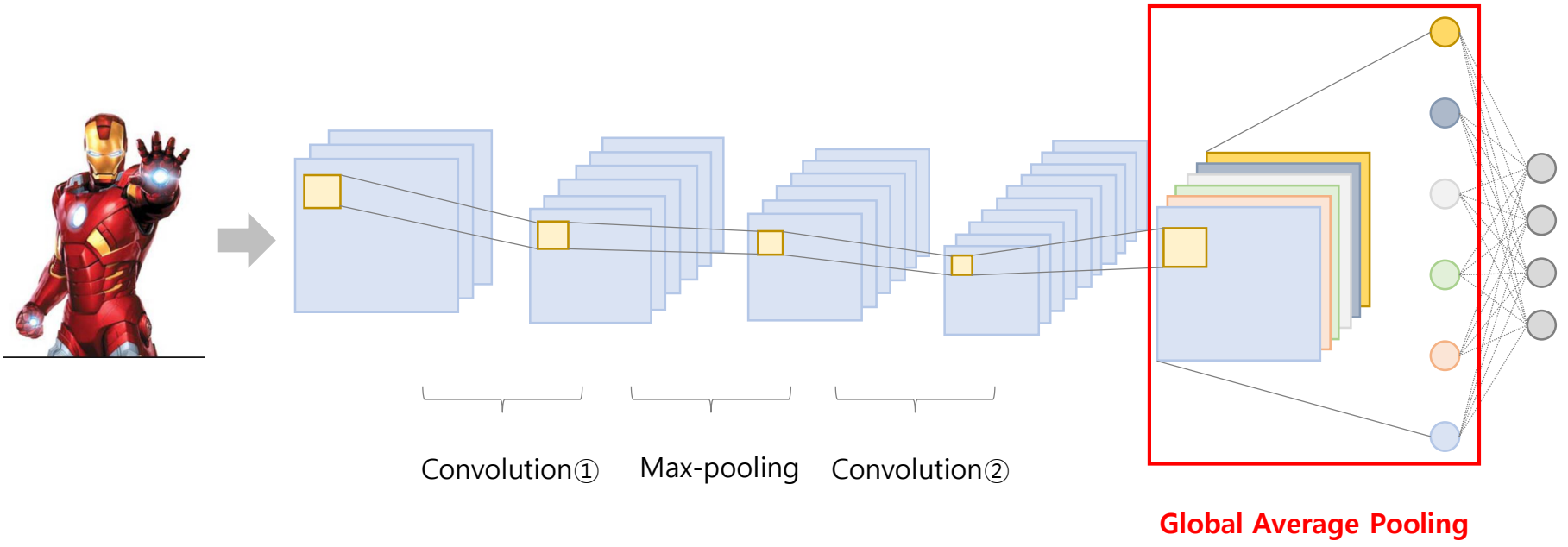
- ➢ 이미지를 입력 변수로 활용하고 이미지의 Class를 맞추는 분류 모델 구조
- ➢ Convolution Layer와 Pooling Layer를 활용해서 이미지 내 정보를 요약
- ➢ 최종 분류 예측 전에 Fully Connected Layer 활용



Class Activation Map(CAM)

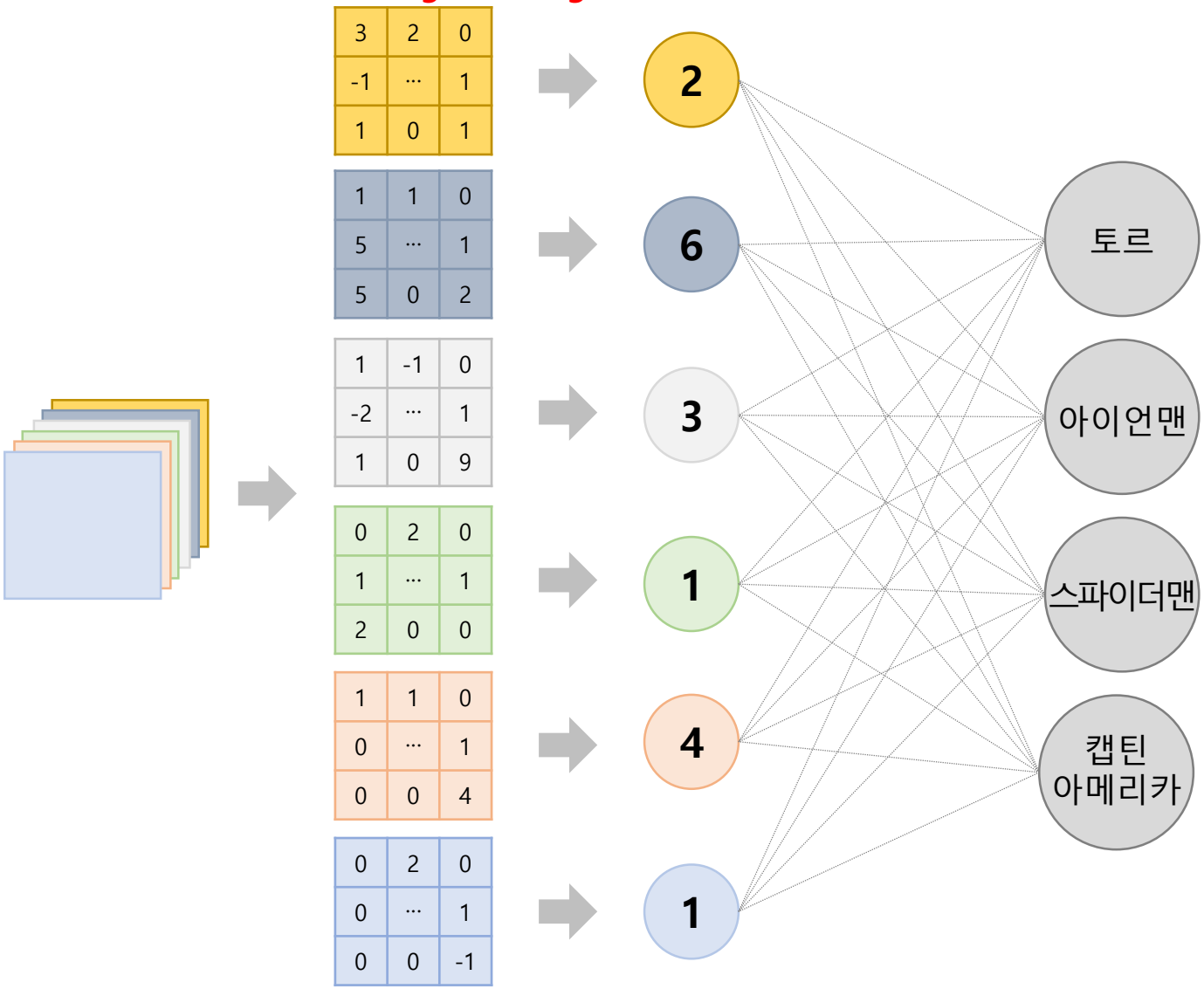
❖ CNN + Class Activation Map(CAM) 구조

- ➢ Convolution Layer와 Pooling Layer를 활용해서 이미지 내 정보를 요약
- ➢ 마지막 Convolutional layer 뒤에 Global Average Pooling 구조를 사용

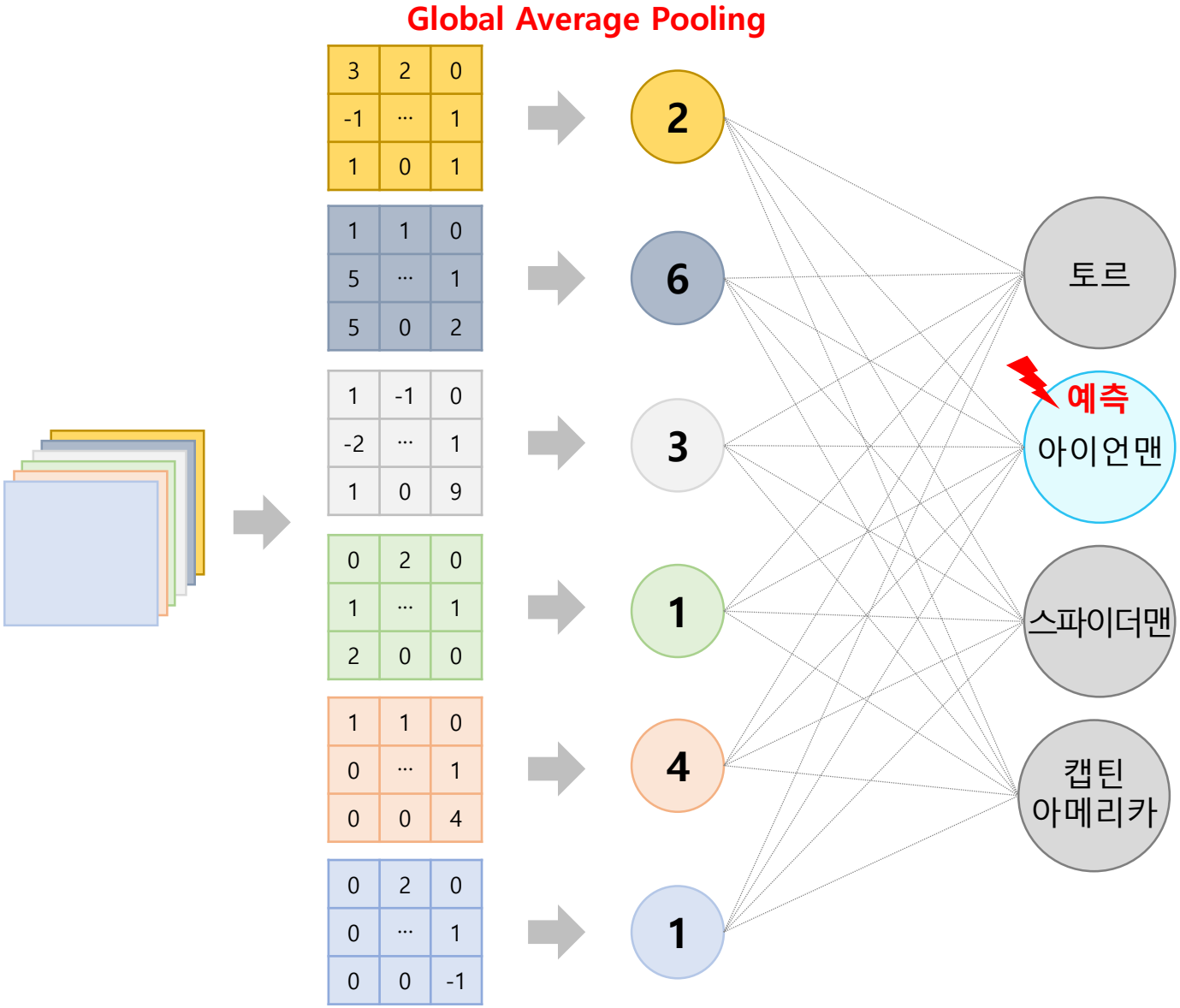


Class Activation Map(CAM)

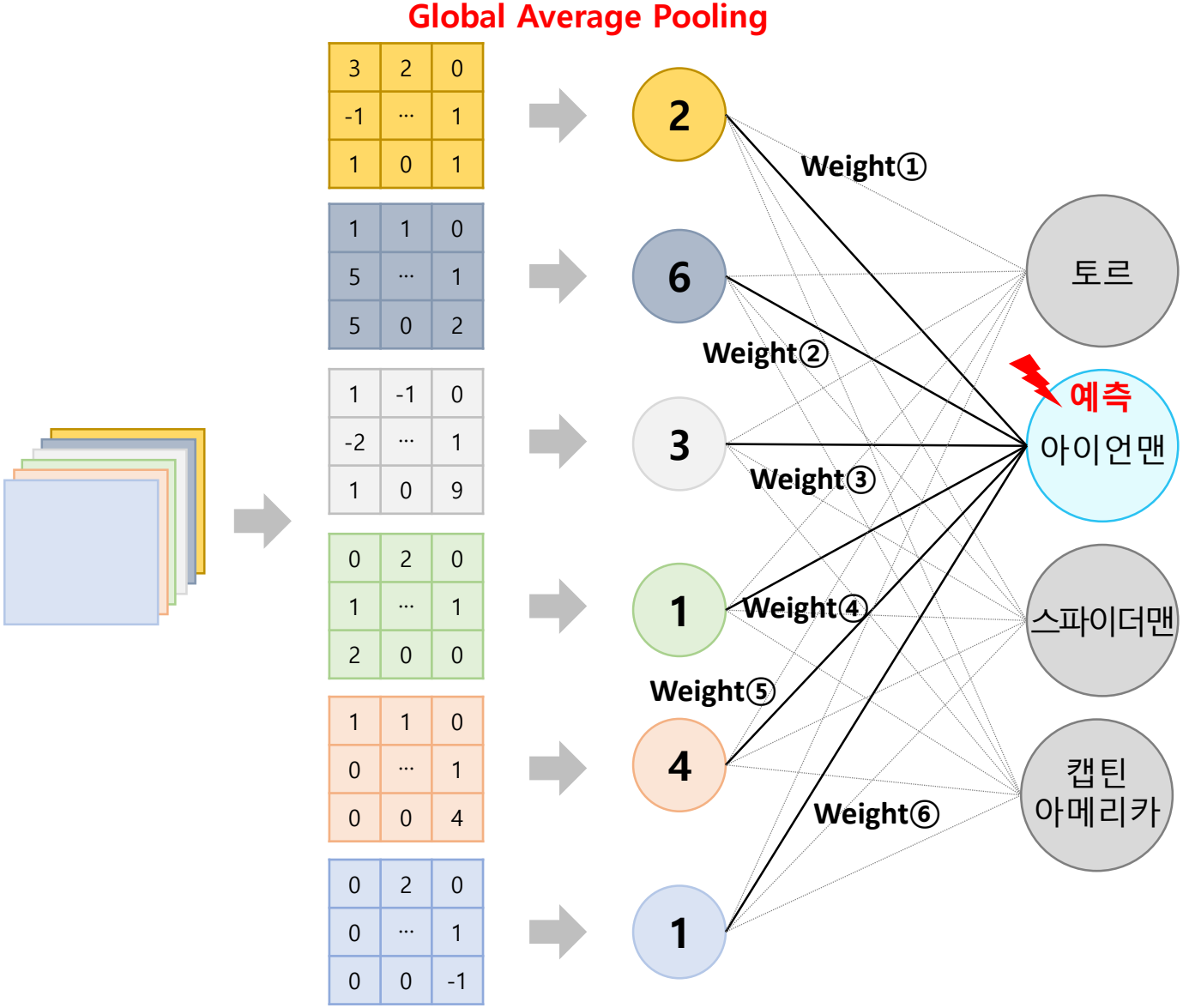
Global Average Pooling → 각 Feature별 평균 값을 구함



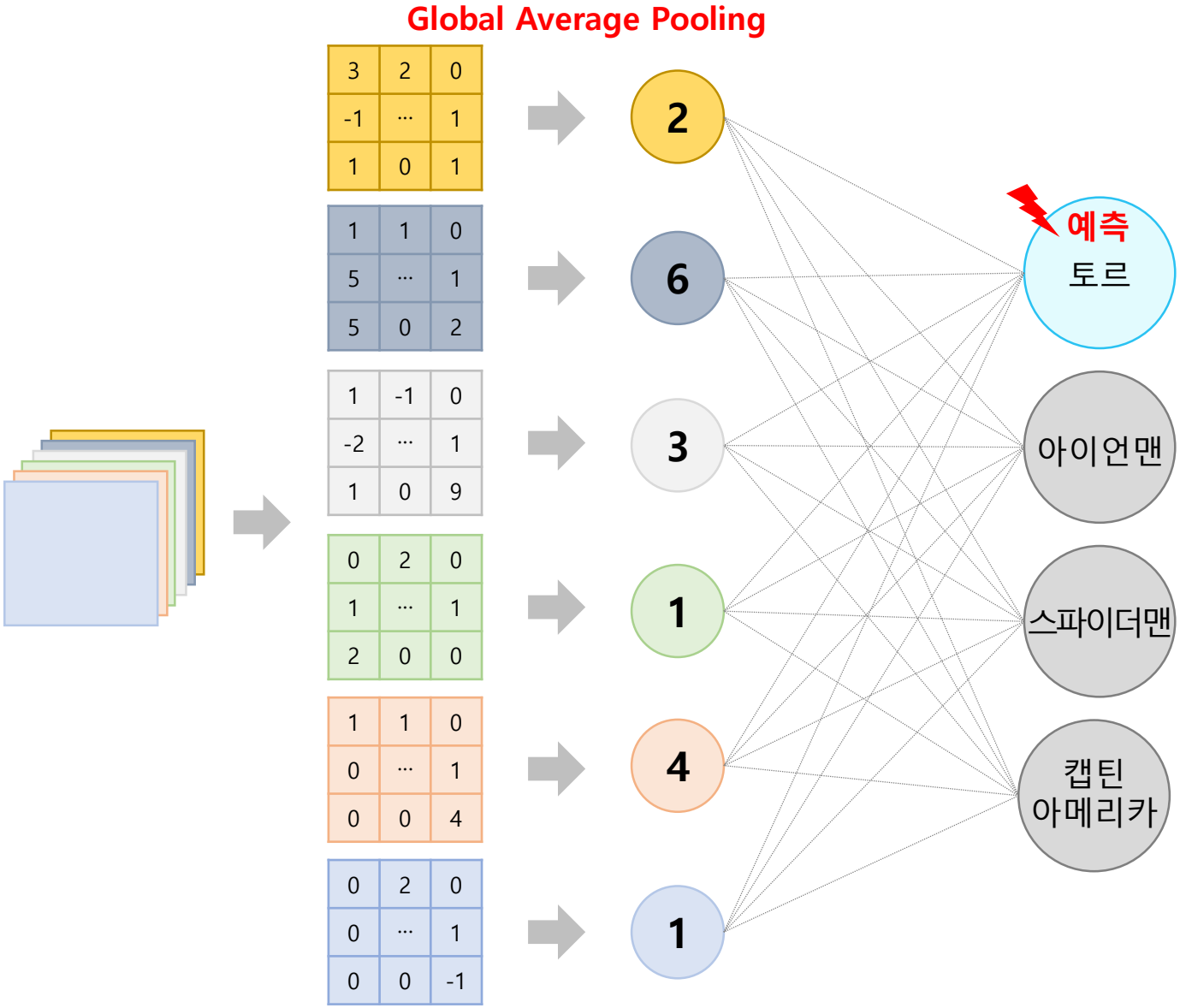
Class Activation Map(CAM)



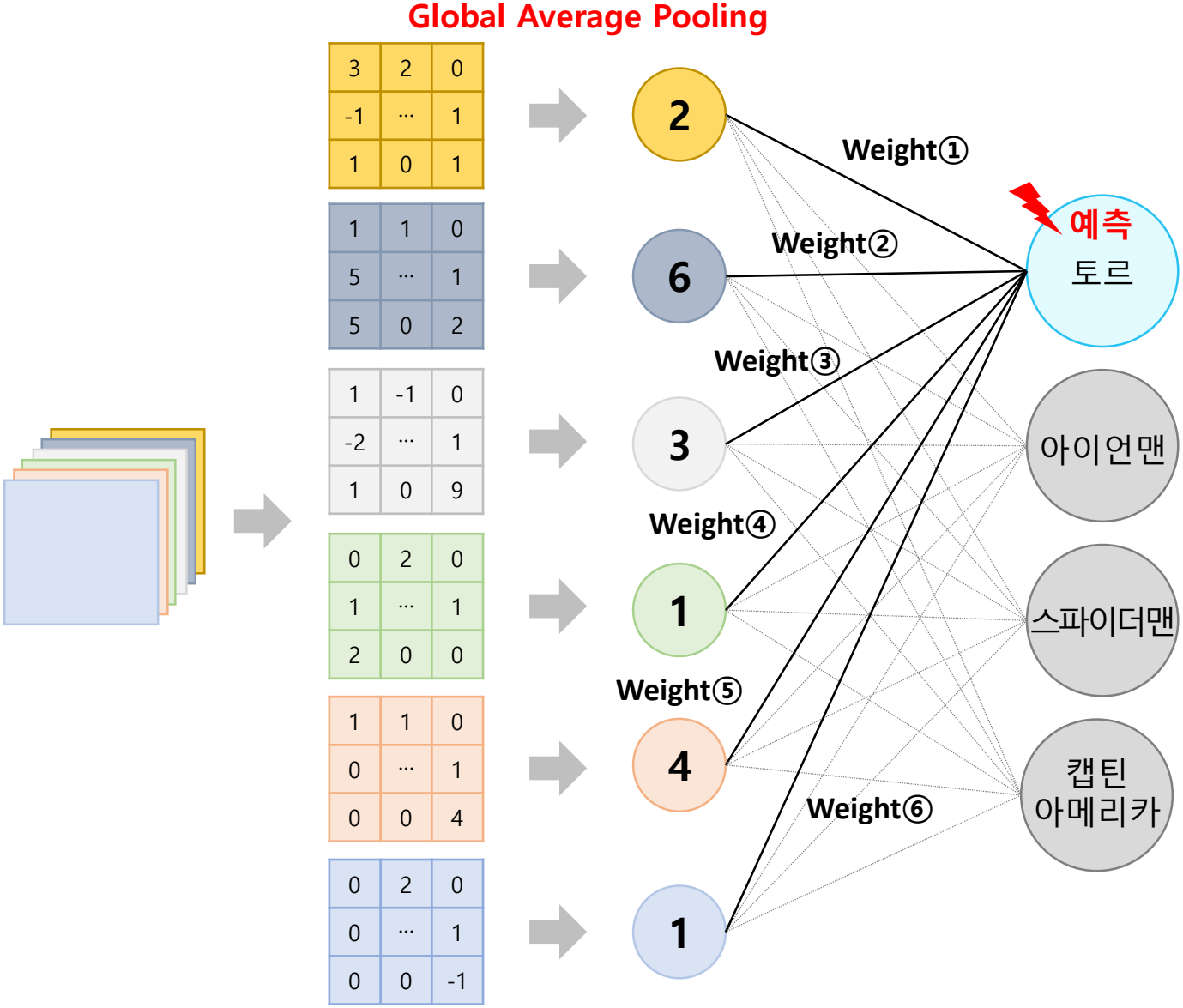
Class Activation Map(CAM)



Class Activation Map(CAM)

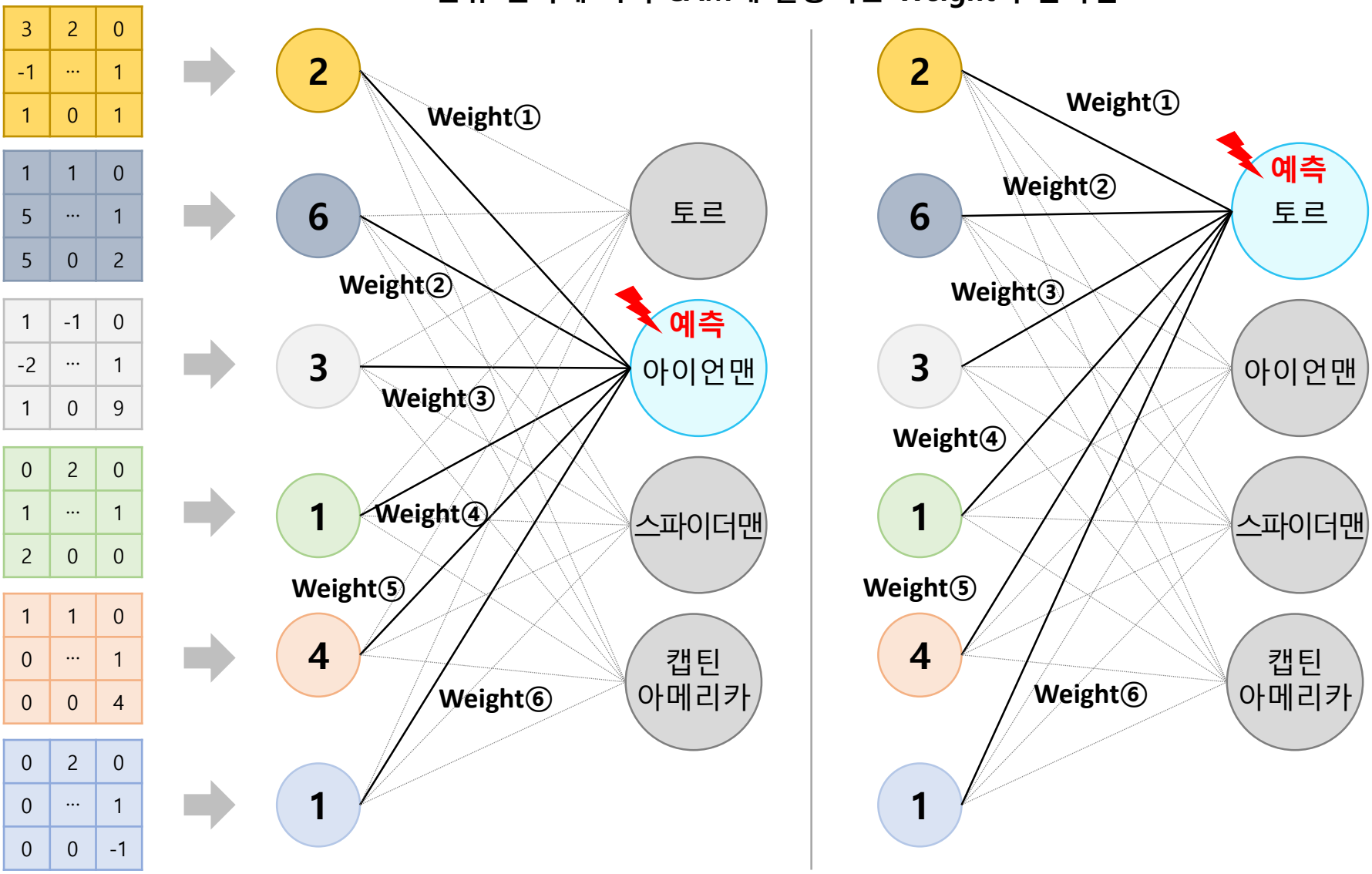


Class Activation Map(CAM)

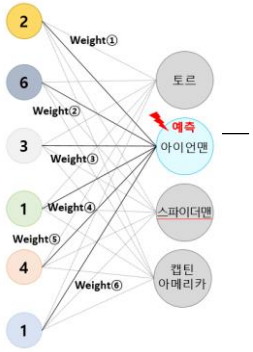


Class Activation Map(CAM)

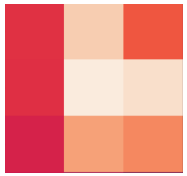
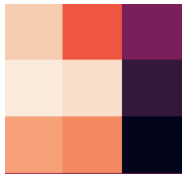
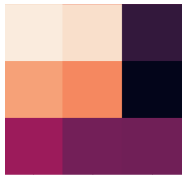
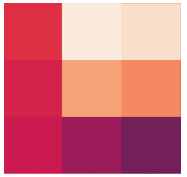
분류 결과에 따라 CAM에 활용되는 Weight가 달라짐



Class Activation Map(CAM)

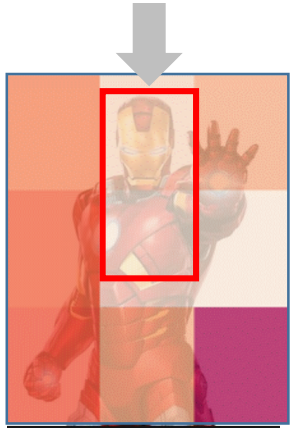


각 Feature 별 heatmap을 그림

3	2	0	x	Weight①	=	
-1	···	1				
1	0	1				
1	1	0	x	Weight②	=	
5	···	1				
5	0	2				
1	-1	0	x	Weight③	=	
-2	···	1				
1	0	9				
0	2	0	x	Weight④	=	
1	···	1				
2	0	0				
1	1	0	x	Weight⑤	=	
0	···	1				
0	0	4				
0	2	0	x	Weight⑥	=	
0	···	1				
0	0	-1				

동일 위치
Pixel별 합

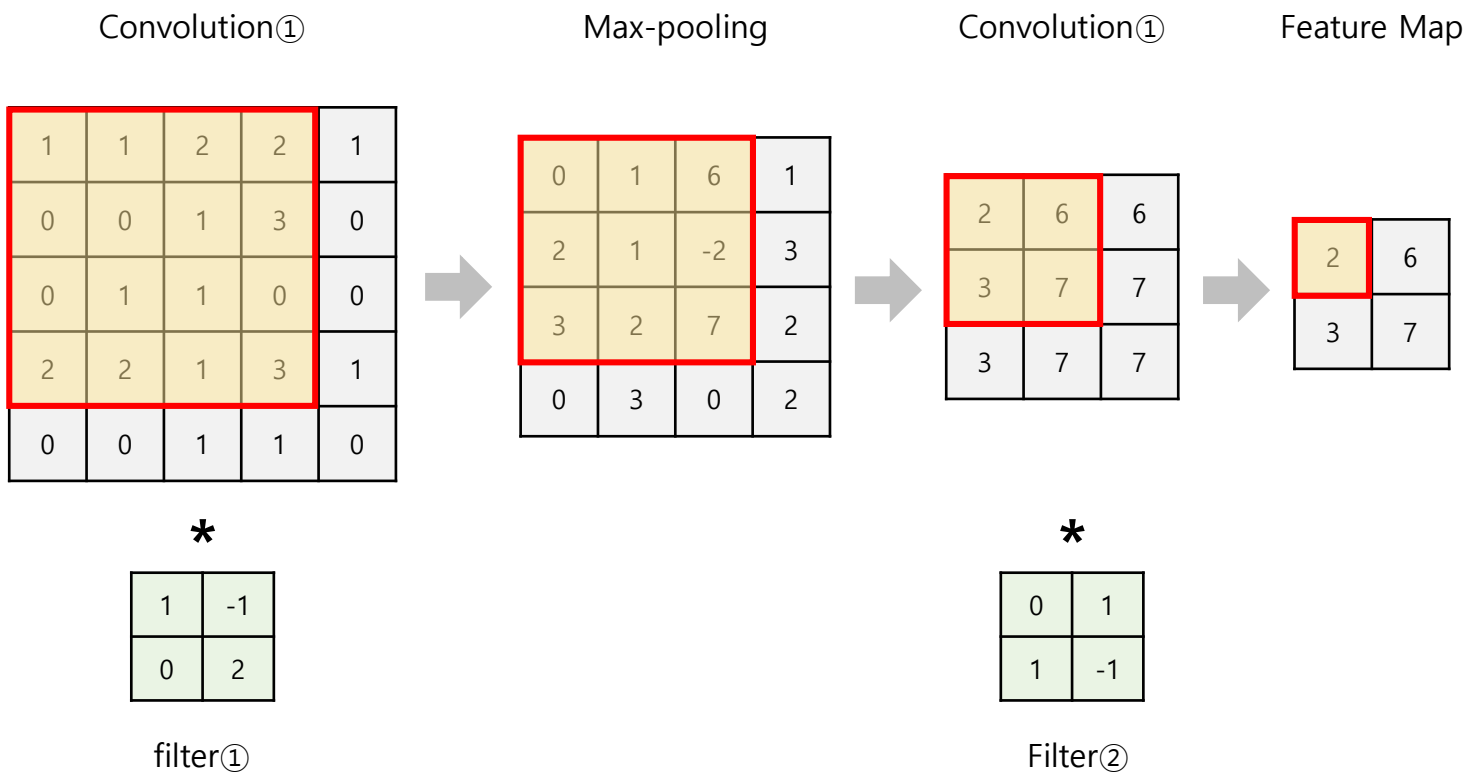
아이언맨 얼굴을
예측 원인으로 판단



Class Activation Map(CAM)

❖ Class Activation Map에서 마지막 Layer만으로도 원인 분석이 가능한 이유

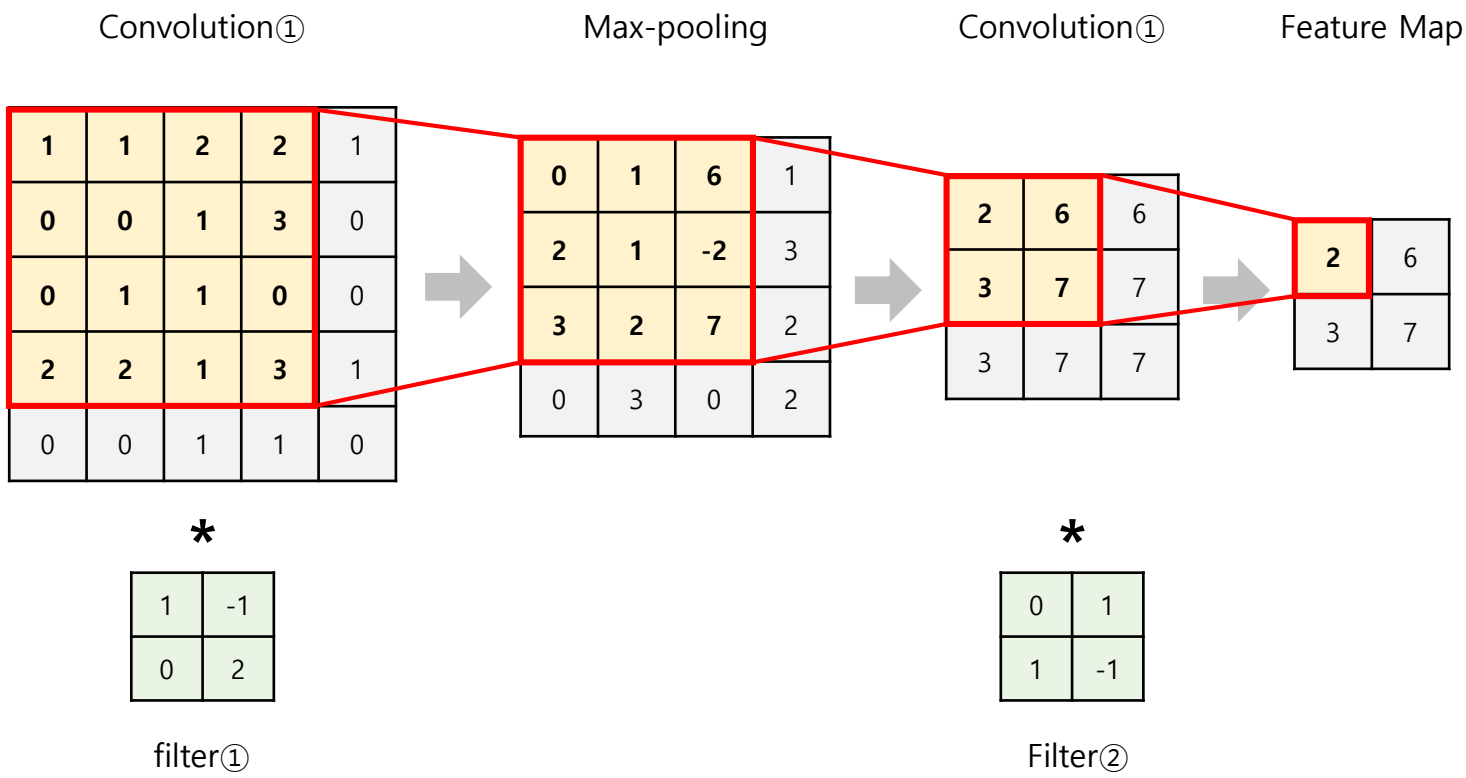
- ➢ 마지막 Convolution Feature Map이 가진 정보량이 많음
- ➢ 마지막 Feature Map 내 1개 값은 원본 이미지에서 많은 부분을 요약한 결과임



Class Activation Map(CAM)

❖ Class Activation Map에서 마지막 Layer만으로도 원인 분석이 가능한 이유

- ➢ 마지막 Convolution Feature Map이 가진 정보량이 많음
- ➢ 마지막 Feature Map 내 1개 값은 원본 이미지에서 많은 부분을 요약한 결과임



4. Grad_CAM

Grad_CAM

- Grad-CAM(2017)
  - Class Activation Map(CAM) 방법론 이후 1년 뒤에 등장
  - 2017년도 ICCV(International Conference on Computer Vision)에서 소개 됨

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 valida- tion accuracy, it **did** not generalize as well (82 ... **Grad-CAM** visualizations of the model predictions re- vealed that the model had learned to look at the ...

☆ 99 1246회 인용 관련 학술자료 전체 6개의 버전 »

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

Ramprasaath R. Selvaraju[1*] Michael Cogswell[1] Abhishek Das[1] Ramakrishna Vedantam[1*]
Devi Parikh[1,2] Dhruv Batra[1,2]
[1]Georgia Institute of Technology [2]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-*

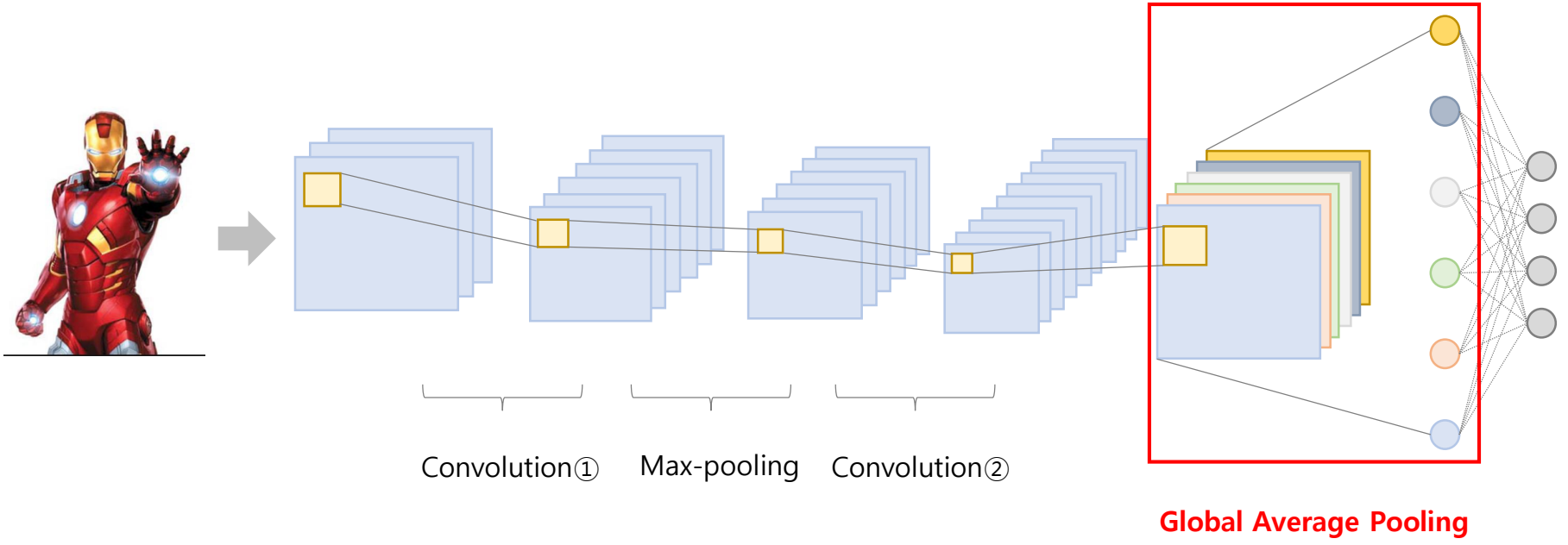
1. Introduction

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

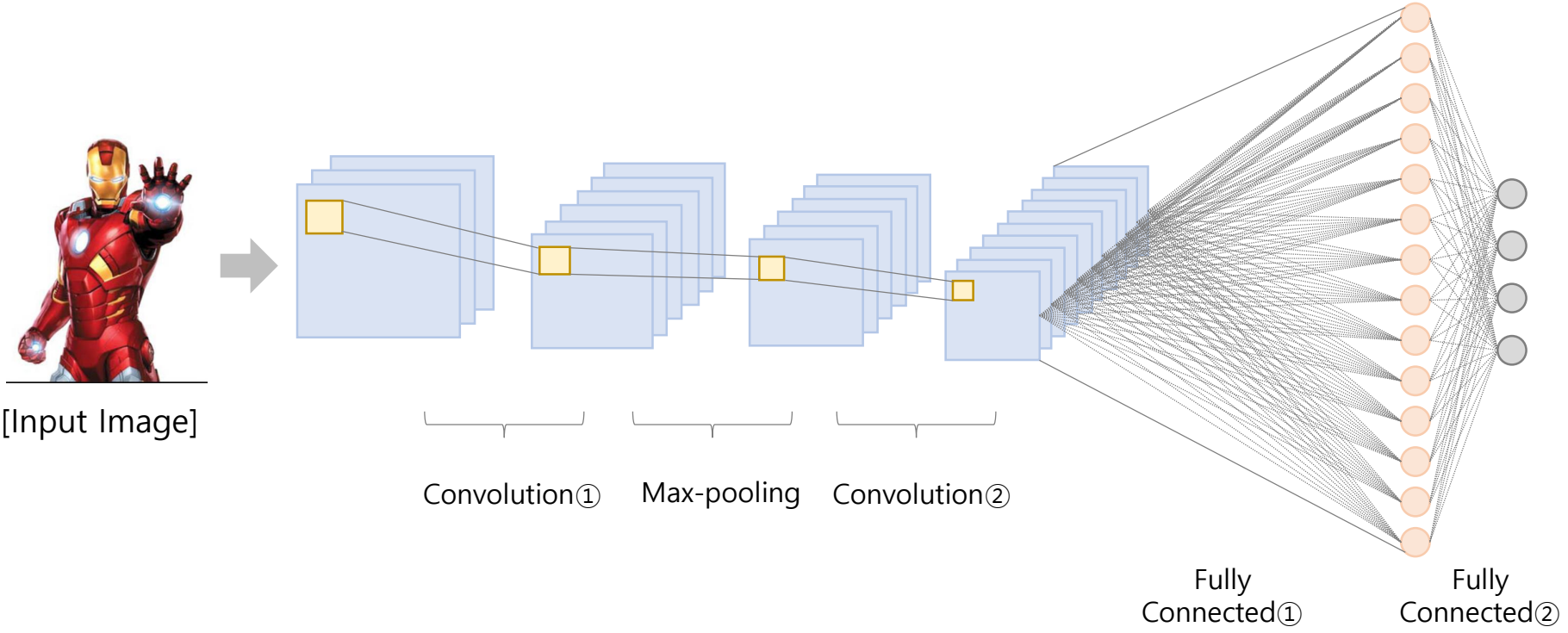
- CNN + Class Activation Map(CAM) 구조
  - 마지막 Convolutional layer 뒤에 Global Average Pooling 구조를 사용
  - 하지만 CNN 구조에서 GAP 부분을 꼭 넣어줘야 하기 때문에 한계가 발생



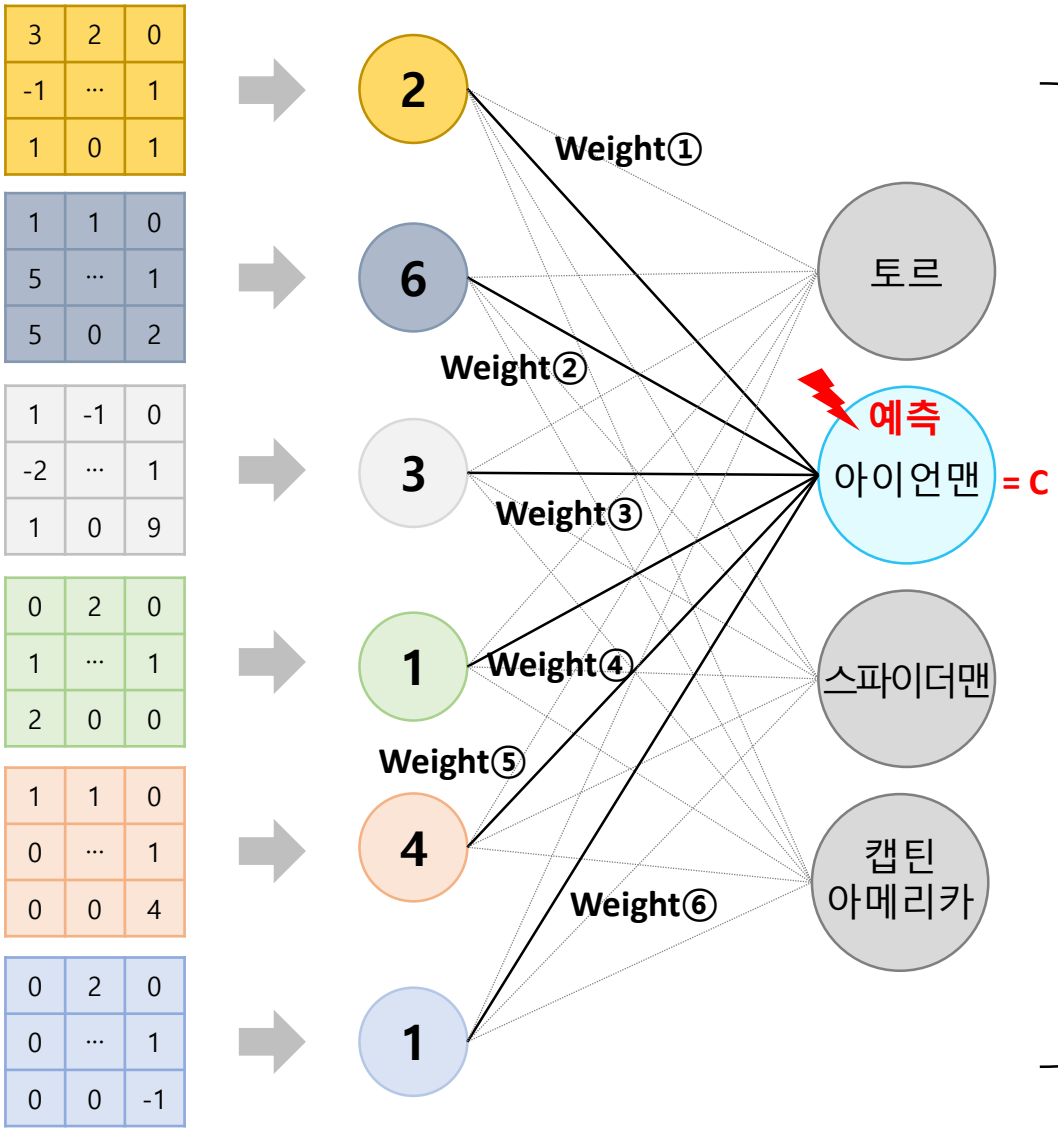
Grad_CAM

❖ CNN + Grad_CAM 구조

- ➢ Class Activation Map(CAM)에서 마지막 Convolution Layer 뒤 GAP 사용 안함
- ➢ CNN 기본 구조를 변형하지 않고 그대로 사용함



Grad_CAM

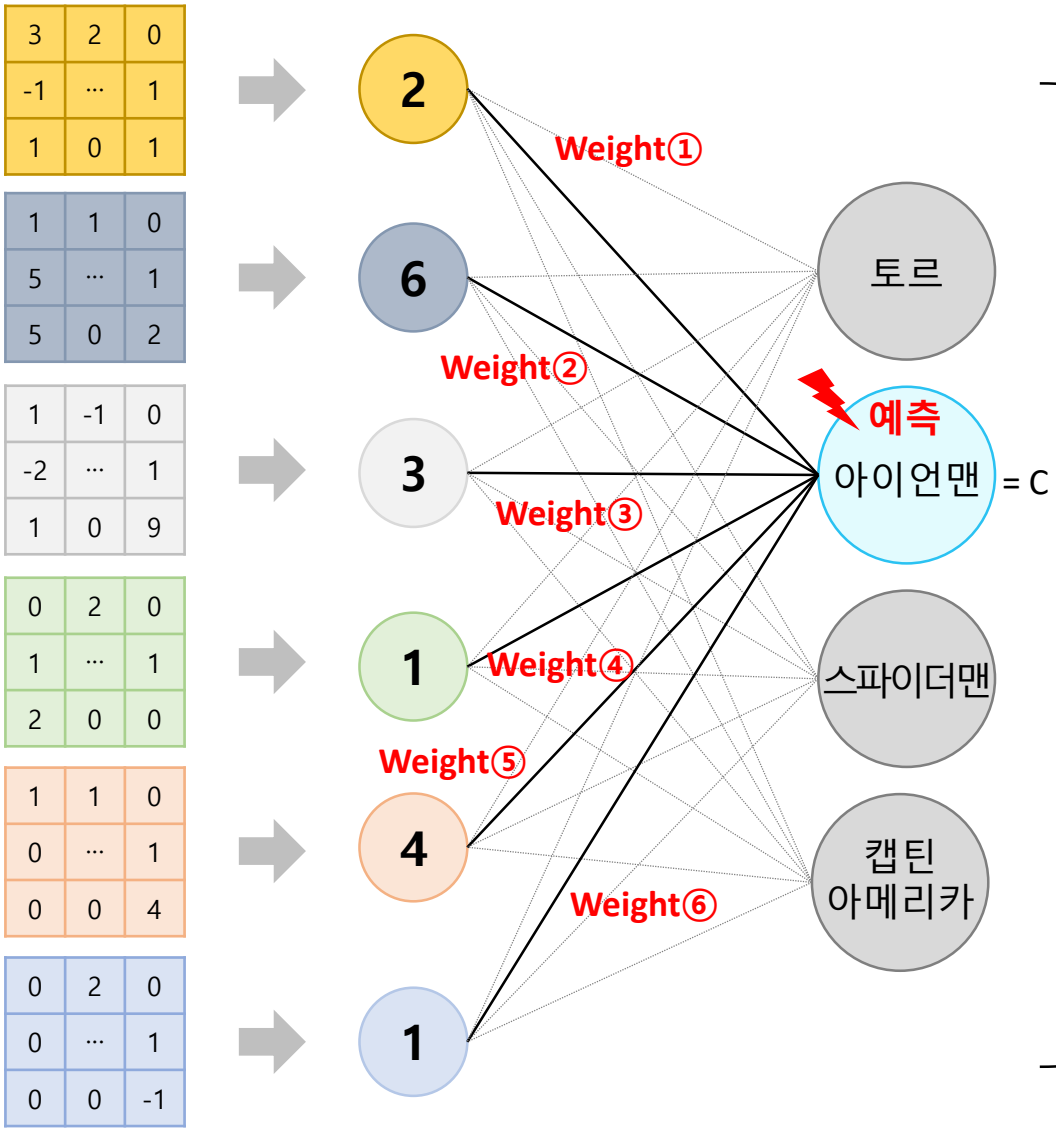


<C class에 대한 CAM Score>

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

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

Grad_CAM

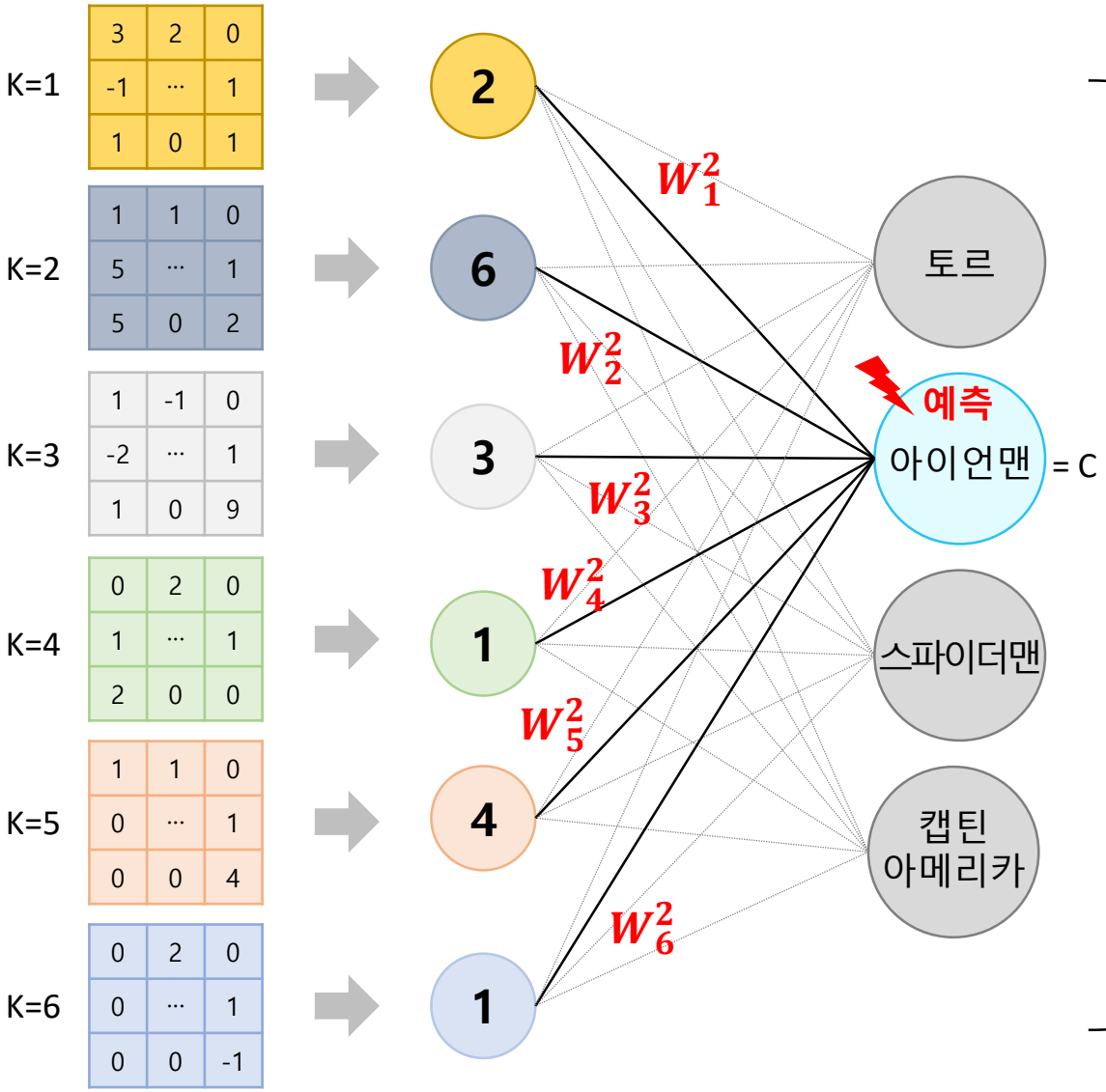


<C class에 대한 CAM Score>

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

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

Grad_CAM

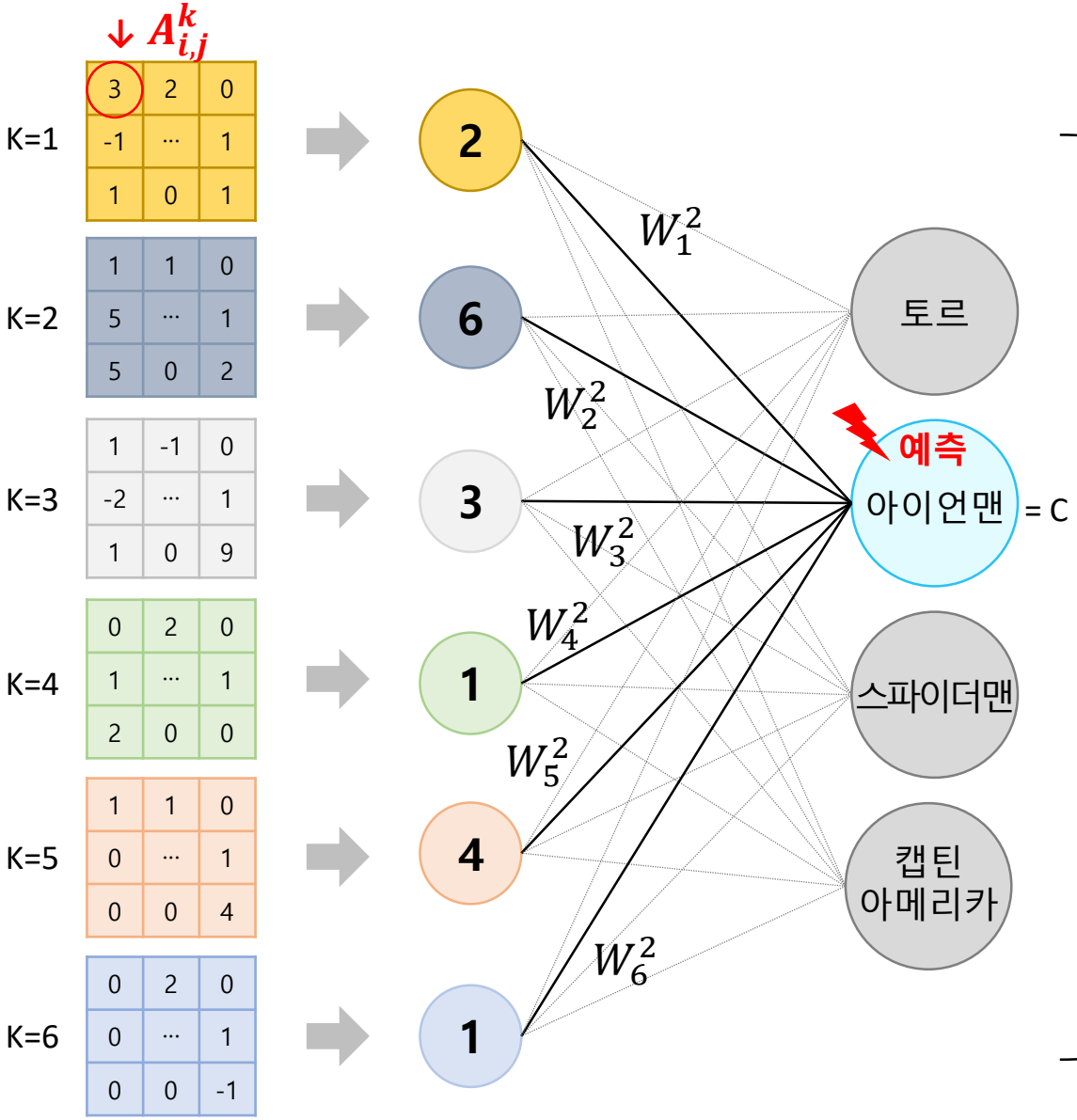


<C class에 대한 CAM Score>

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

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

Grad_CAM

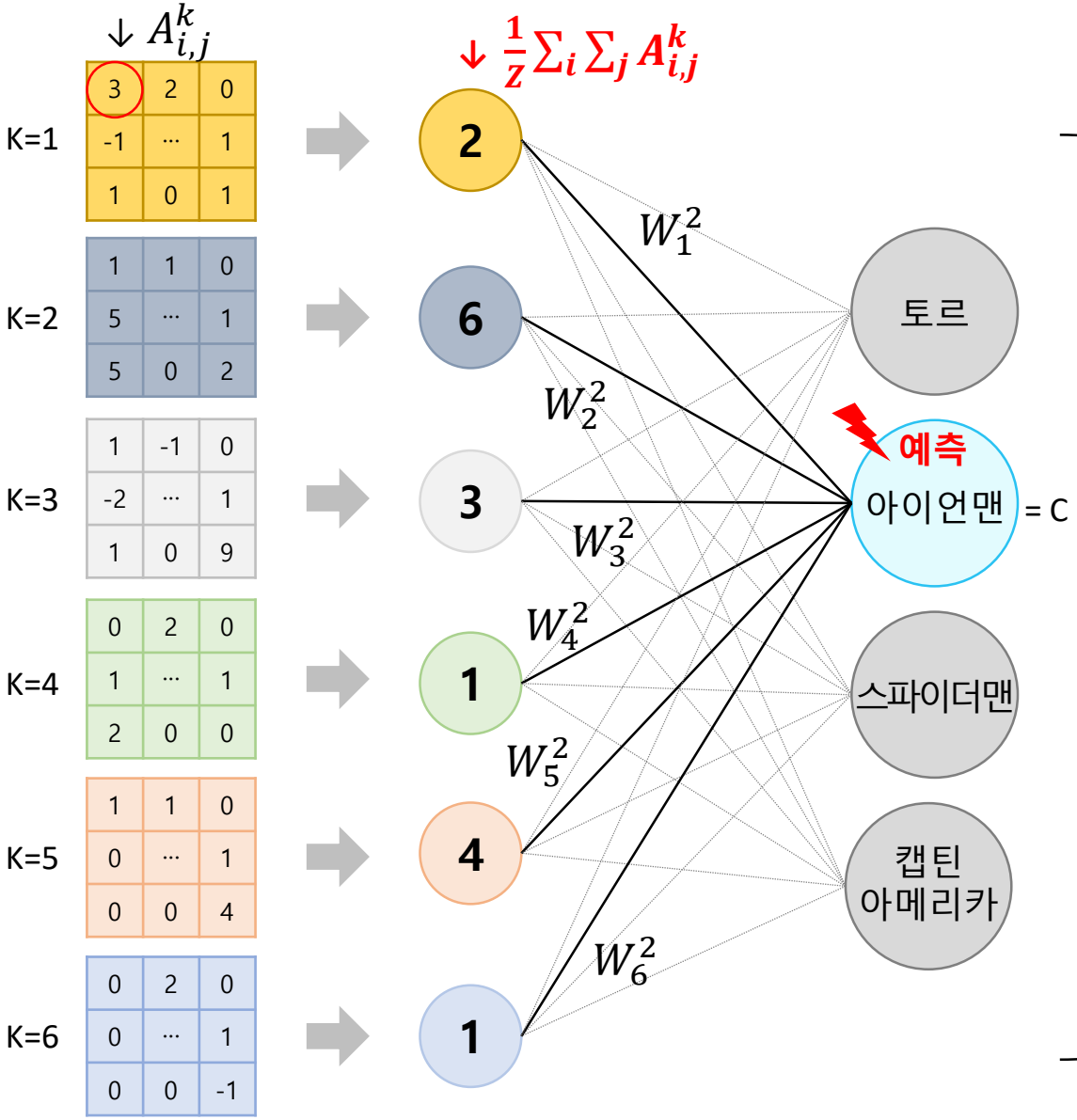


<C class에 대한 CAM Score>

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

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

Grad_CAM

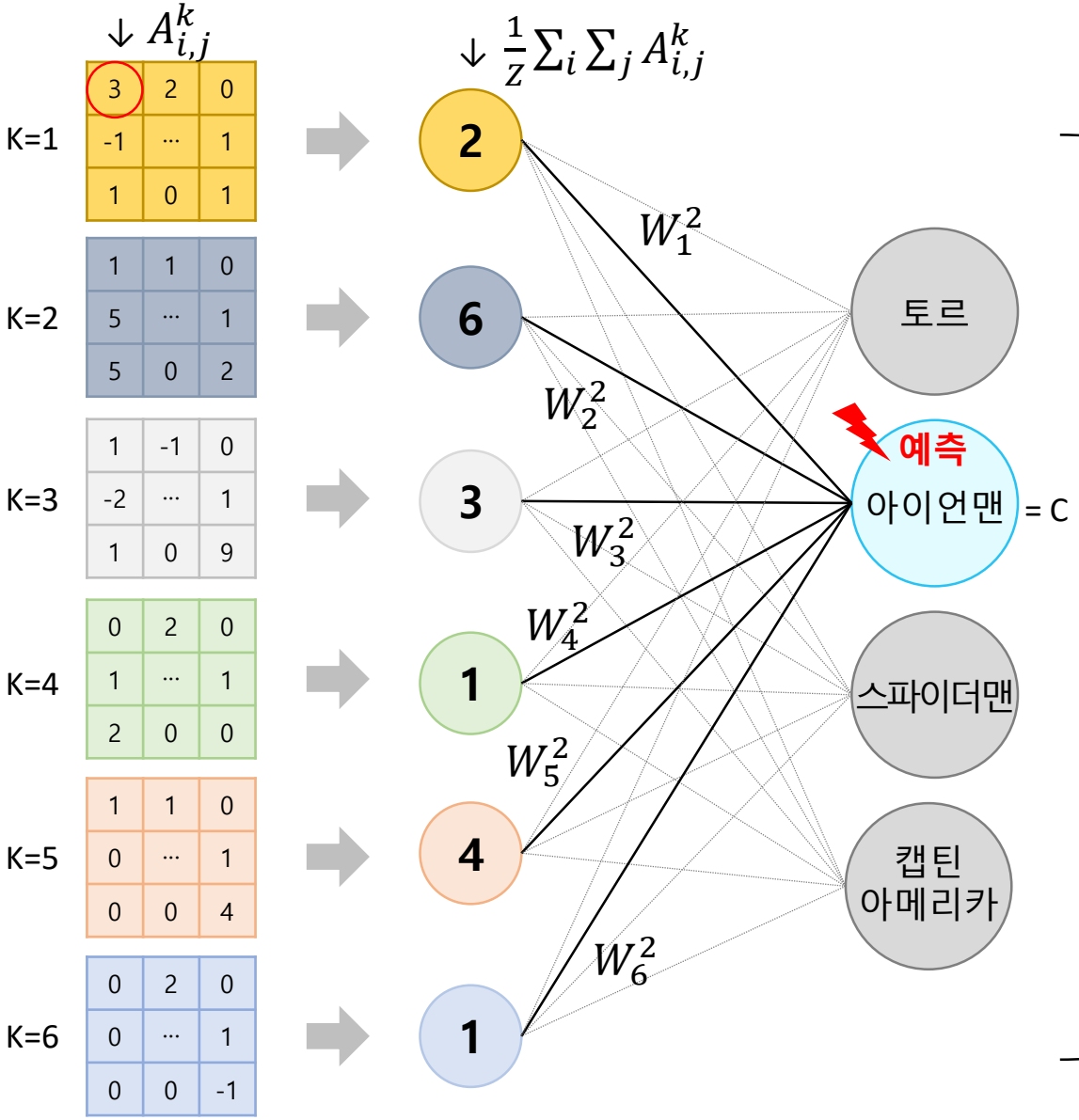


<C class에 대한 CAM Score>

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

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

Grad_CAM



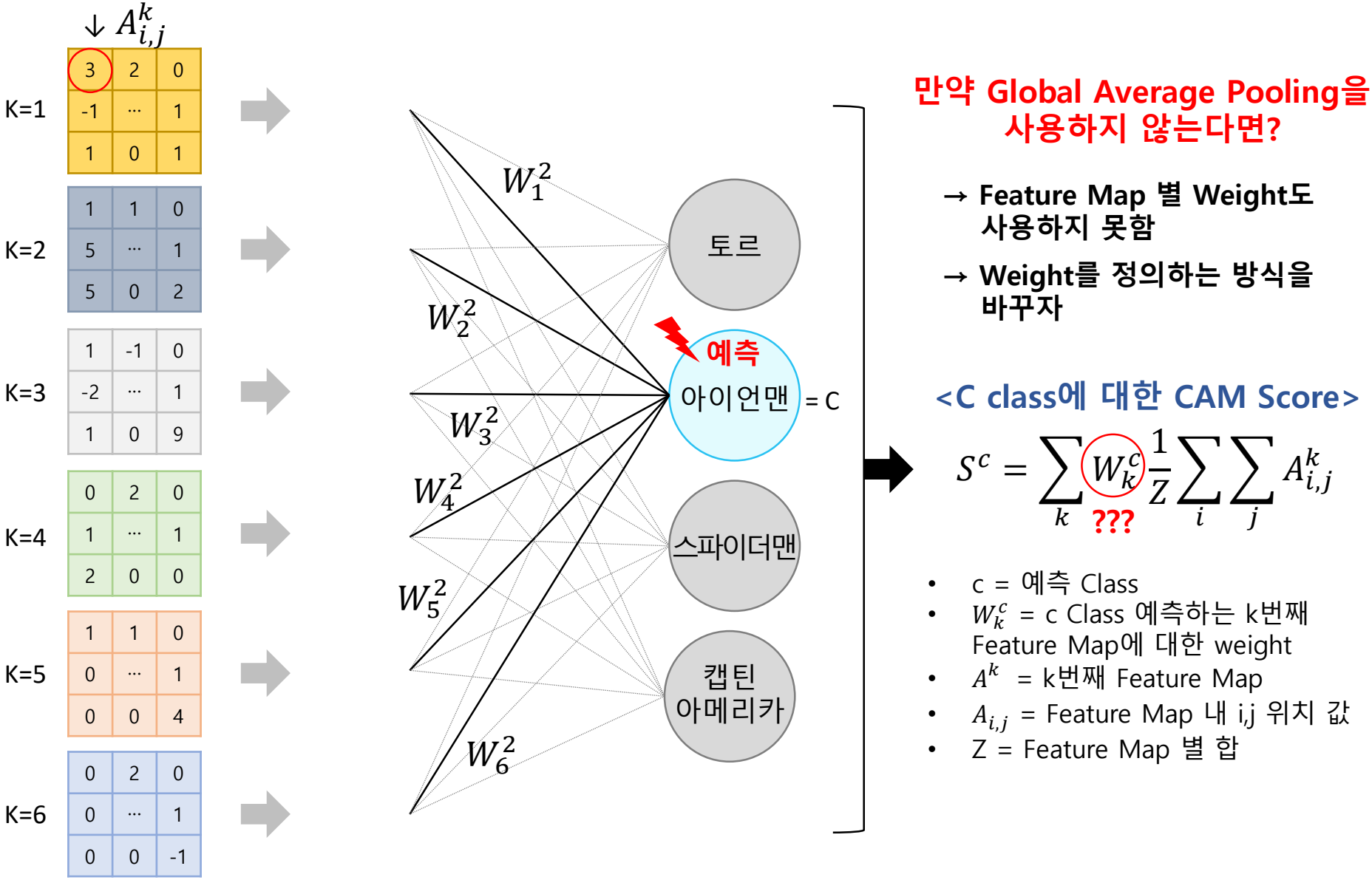
만약 Global Average Pooling을 사용하지 않는다면?

<C class에 대한 CAM Score>

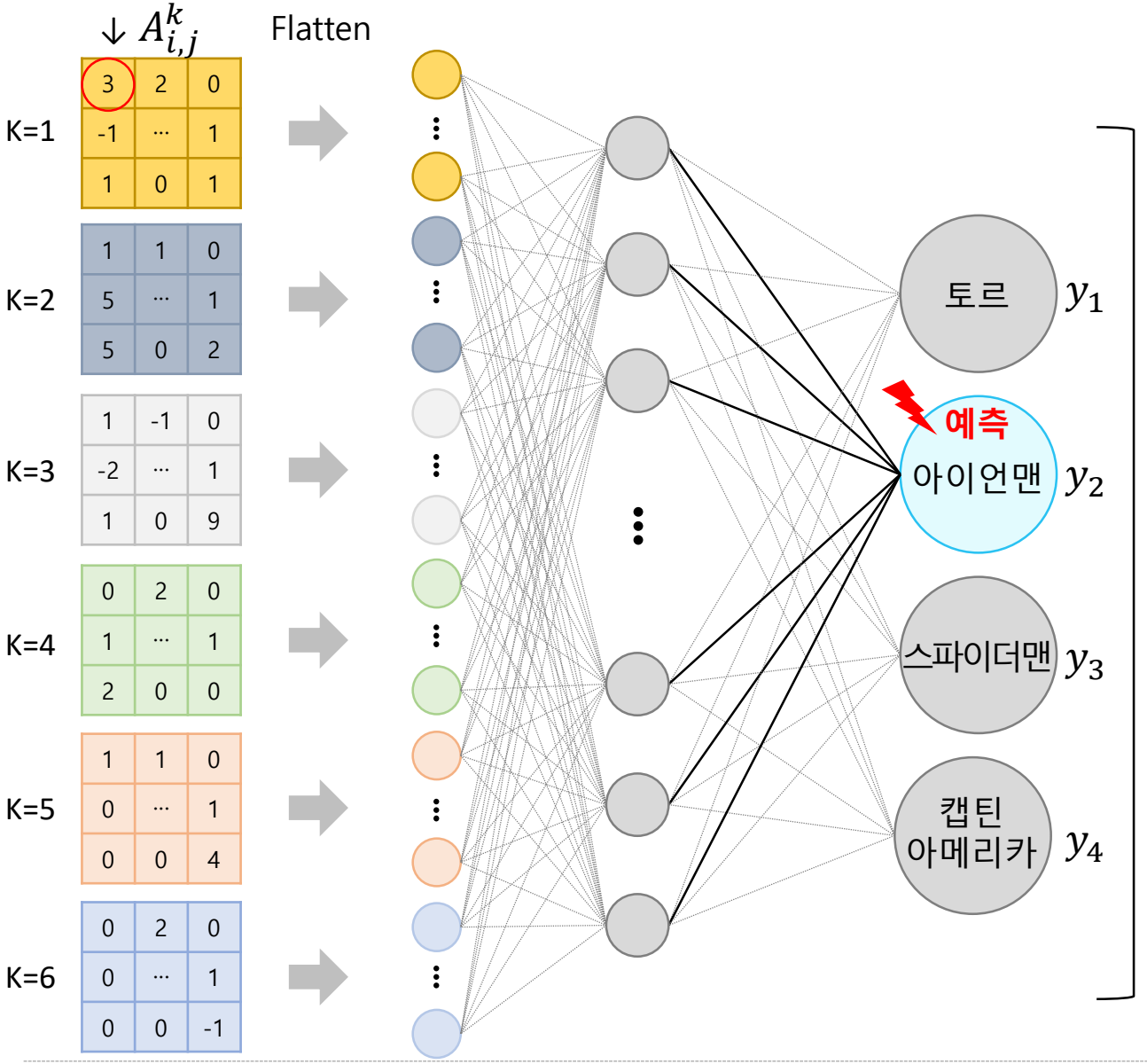
$$S^c = \sum_k W_k^c \frac{1}{Z} \sum_i \sum_j A_{i,j}^k$$

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

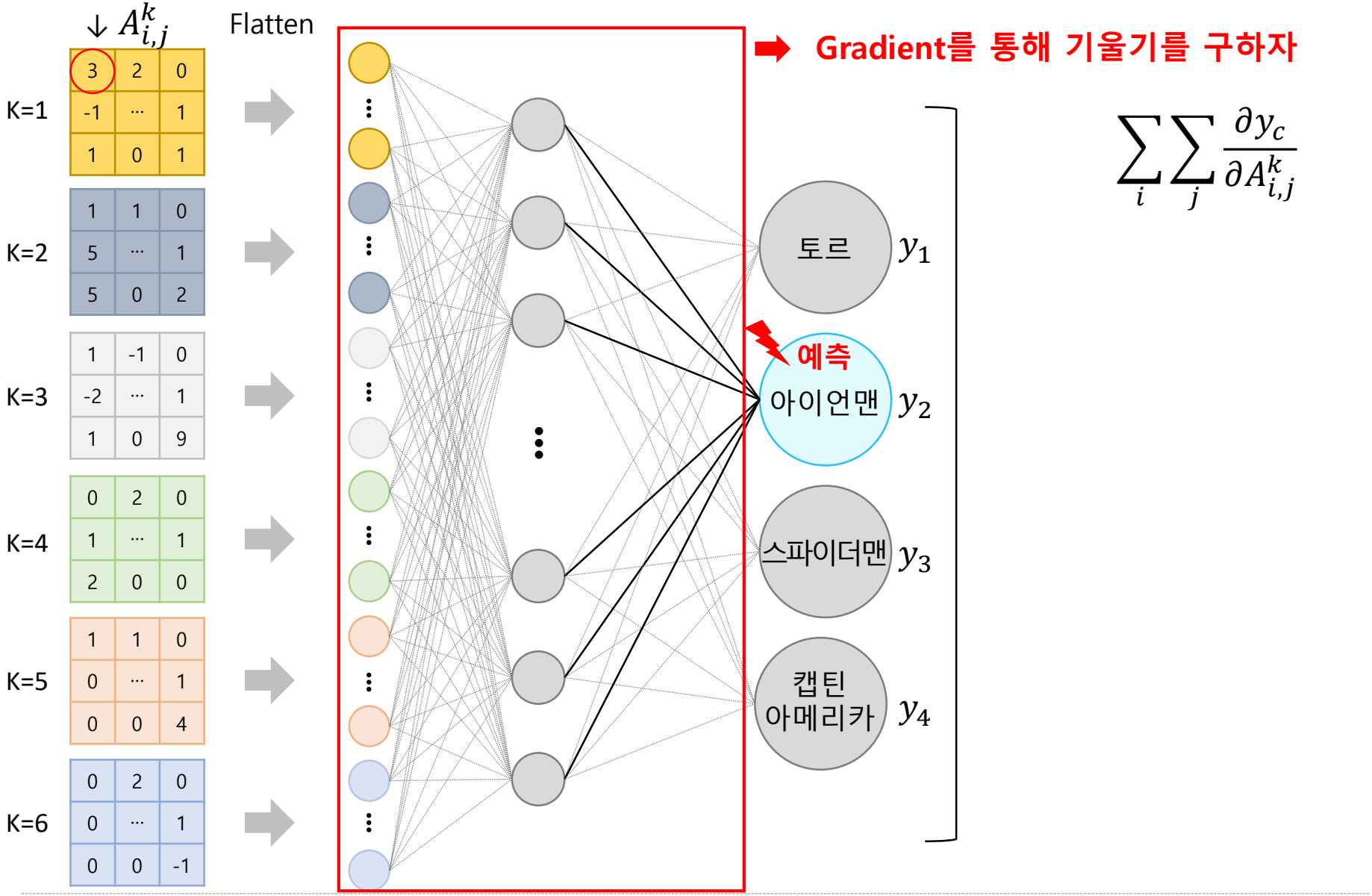
Grad_CAM



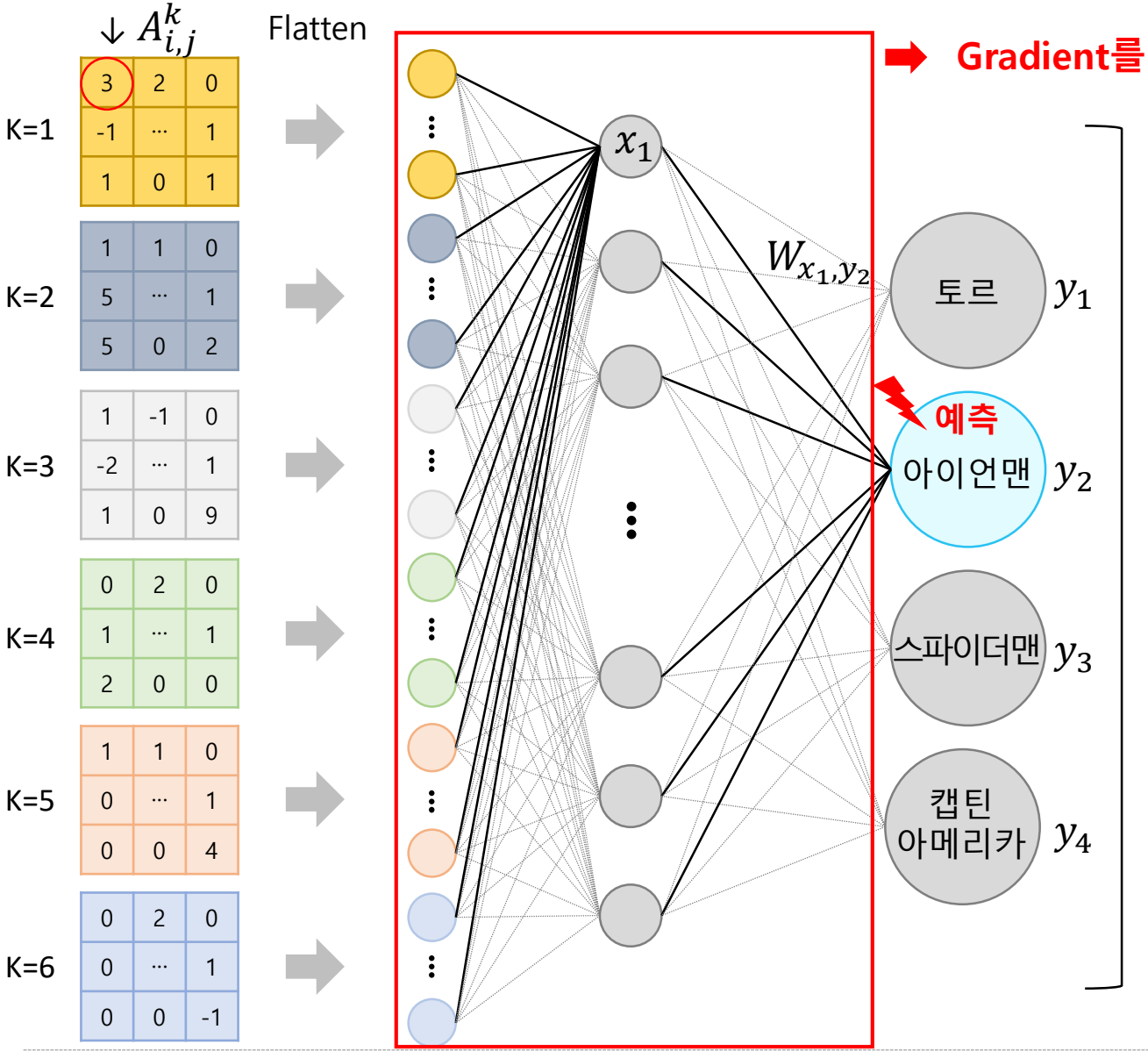
Grad_CAM



Grad_CAM



Grad_CAM



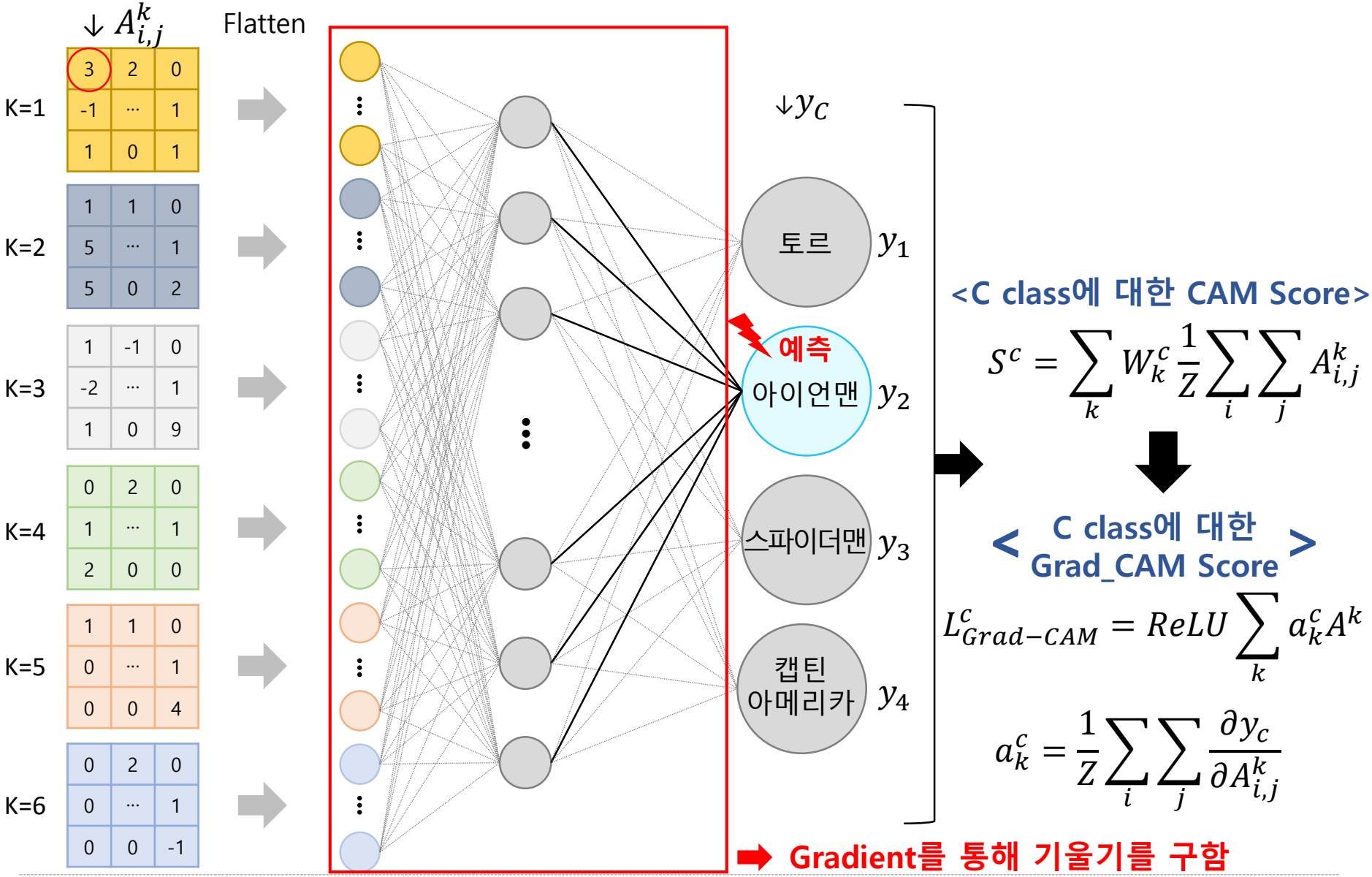
➡ **Gradient를 통해 기울기를 구하자**

$$\sum_i \sum_j \frac{\partial y_c}{\partial A_{i,j}^{k}}$$

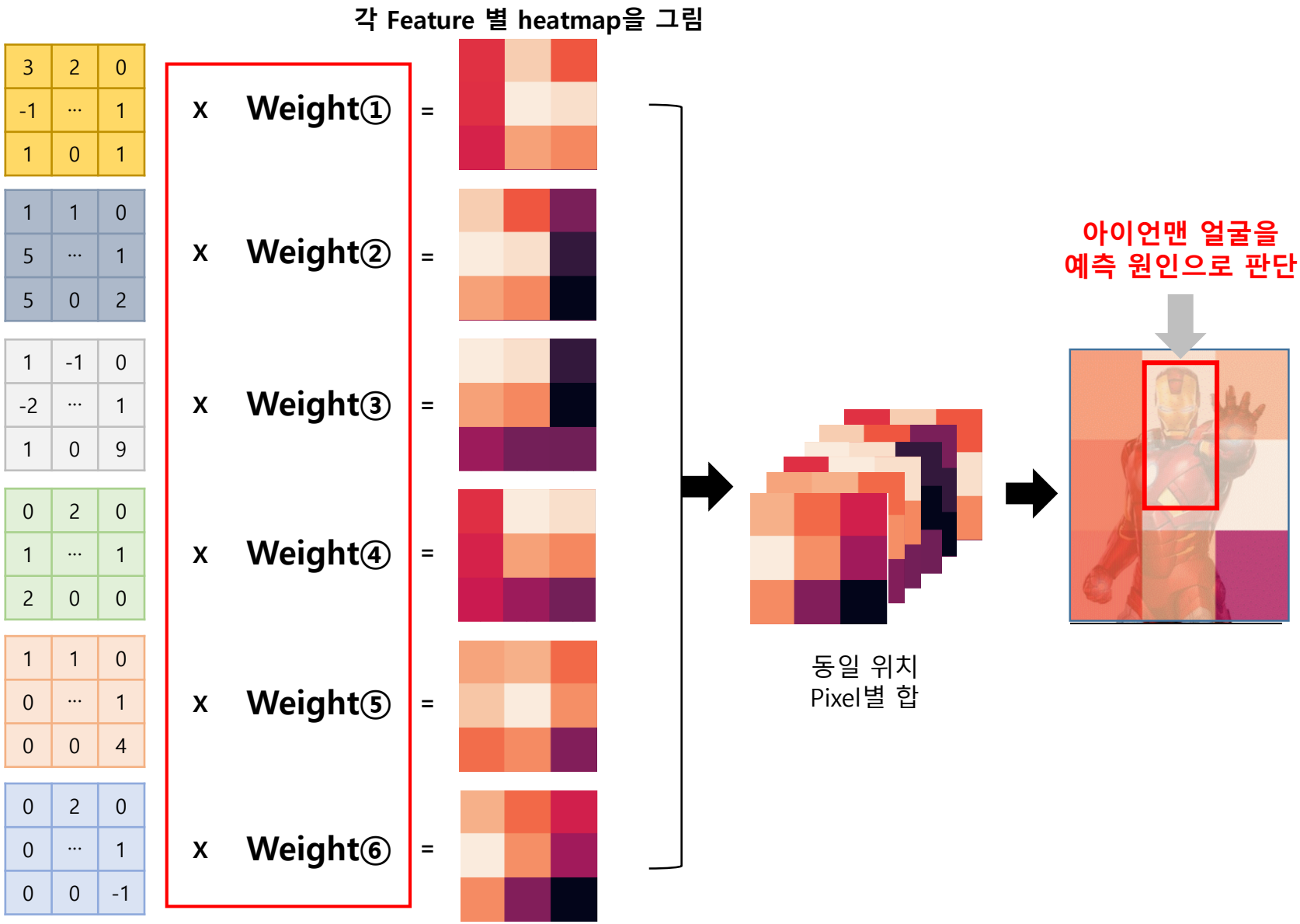
$$y_c = Wx + b$$

$$y_2 = W_{x_1,y_2} x_1 + b$$

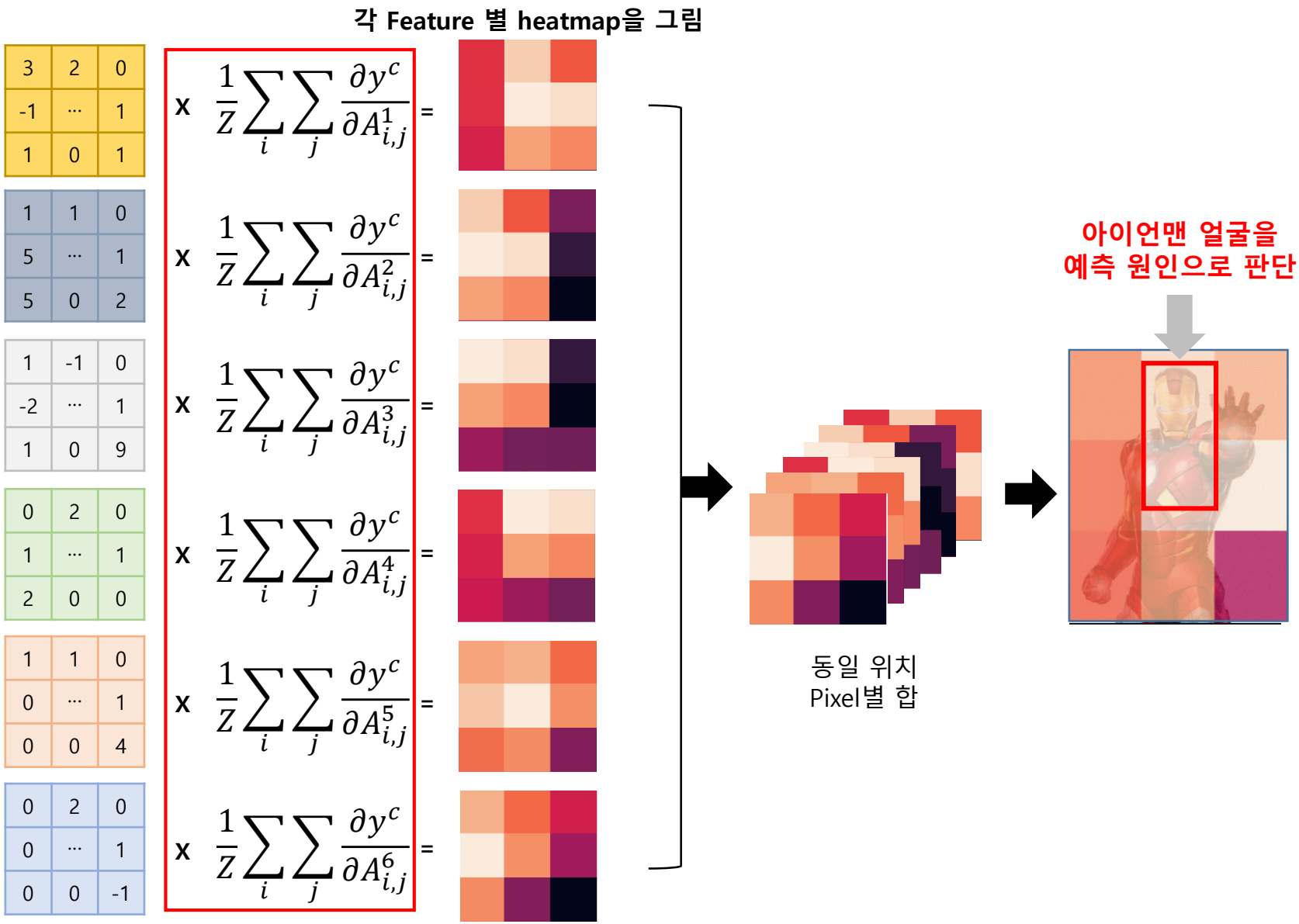
Grad_CAM



Grad_CAM



Grad_CAM



Grad_CAM

❖ Grad_CAM 추가 장점

- ➢ Gradient로 Weight를 구하다 보니 마지막 Layer 아니어도 적용이 가능
- ➢ 아래 사례는 Activation_45 Layer부터 고양이 얼굴 영역을 잡기 시작

출처: https://jsideas.net/grad_cam/

5. Conclusion

Conclusion

❖ Conclusion

① 모델 구조를 변형하는 CAM과 달리 Grad_CAM에서는 모델 구조를 변형하지 않고도 예측 결과에 대한 주요 요인을 찾을 수 있음

② CNN 모델은 원본 데이터 내 정보를 지속해서 요약해 나가는 알고리즘으로 마지막 Layer에 있는 픽셀 한 개 값은 원본 이미지에 넓은 부분을 커버하고 있음

③ CNN 모델을 통해 요약된 마지막 정보와 그 정보로 예측 할 때 사용하는 Weight를 안다면 모델 구조 변경 없이 CAM 적용 가능

❖ 향후 계획

① 현재 진행 중인 Game-AI 프로젝트에 Grad_CAM 적용

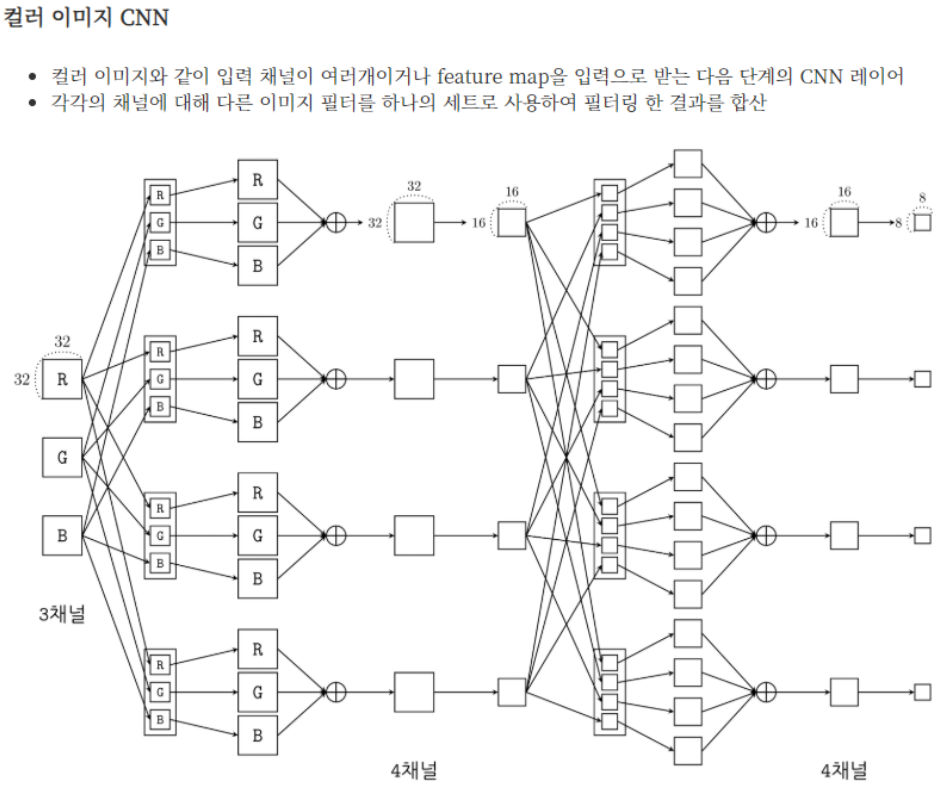
② Attention 기반 알고리즘에 대해 공부하고 CAM 알고리즘과 비교

감사합니다.

Appendix

CNN

- ❖ CNN
  - ➢ CNN 구조



출처: https://datascienceschool.net/view-notebook/e2d30432911f4498b873232dd7d99cce/

CNN

- CNN
  - CNN 구조

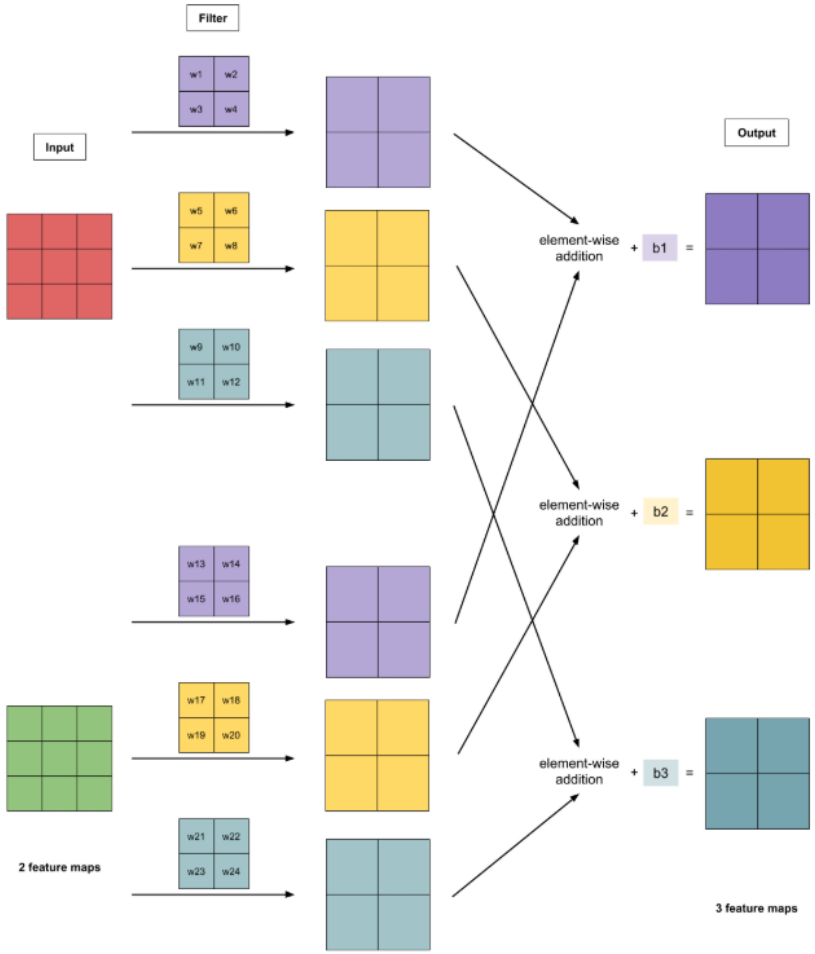


Fig. 3.1: Convolution of a 2-channel image with 2×2 filter to output 3 channels. Here, there are 27 parameters — 24 weights and 3 biases.

출처: https://towardsdatascience.com/counting-no-of-parameters-in-deep-learning-models-by-hand-8f1716241889

❖ 출처

- https://jsideas.net/grad_cam/
- https://you359.github.io/cnn%20visualization/GradCAM/
- https://hugrypiggykim.com/2018/03/28/grad-cam-gradient-weighted-class-activation-mapping/