CAM

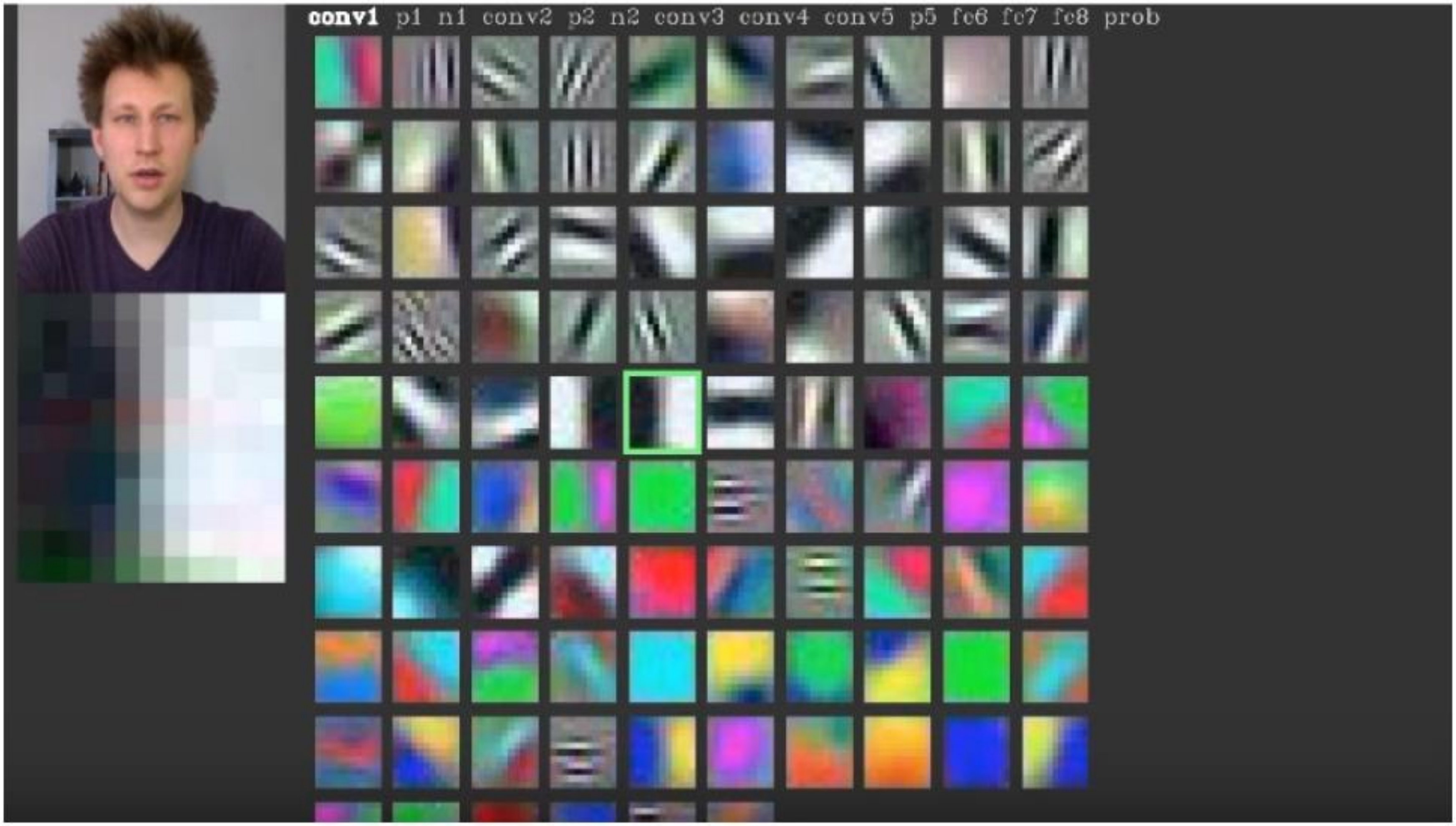
약한 지도학습

Dr Rhee
Jan 2021

CAM(Class Activation Map 클래스 활성맵)

활성맵의 시각화

- 활성맵 시각화

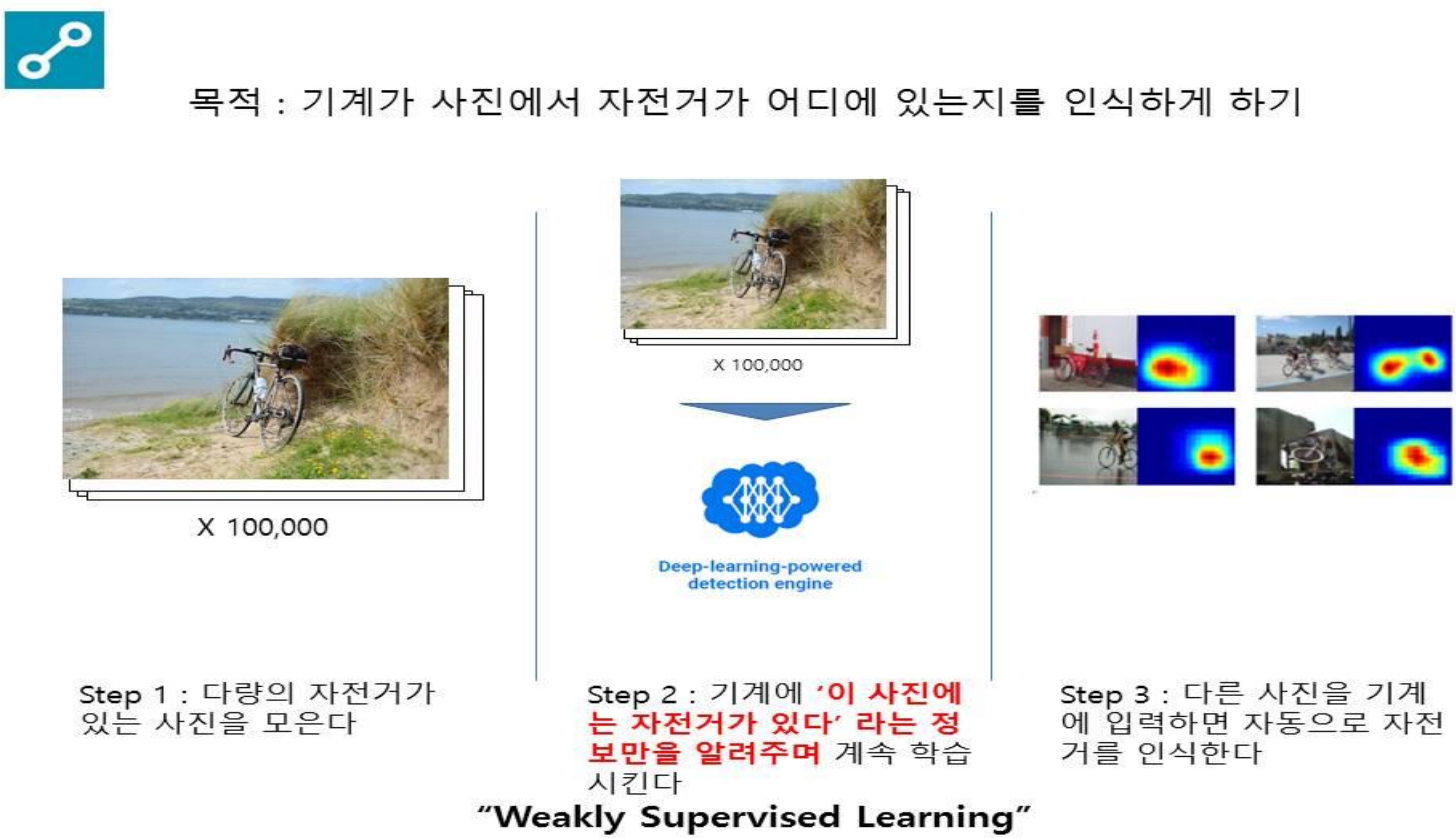


Class Activation Map (클래스 활성맵)

- 지도학습(Supervised Learning): 정답 즉 레이블이 필요하다.

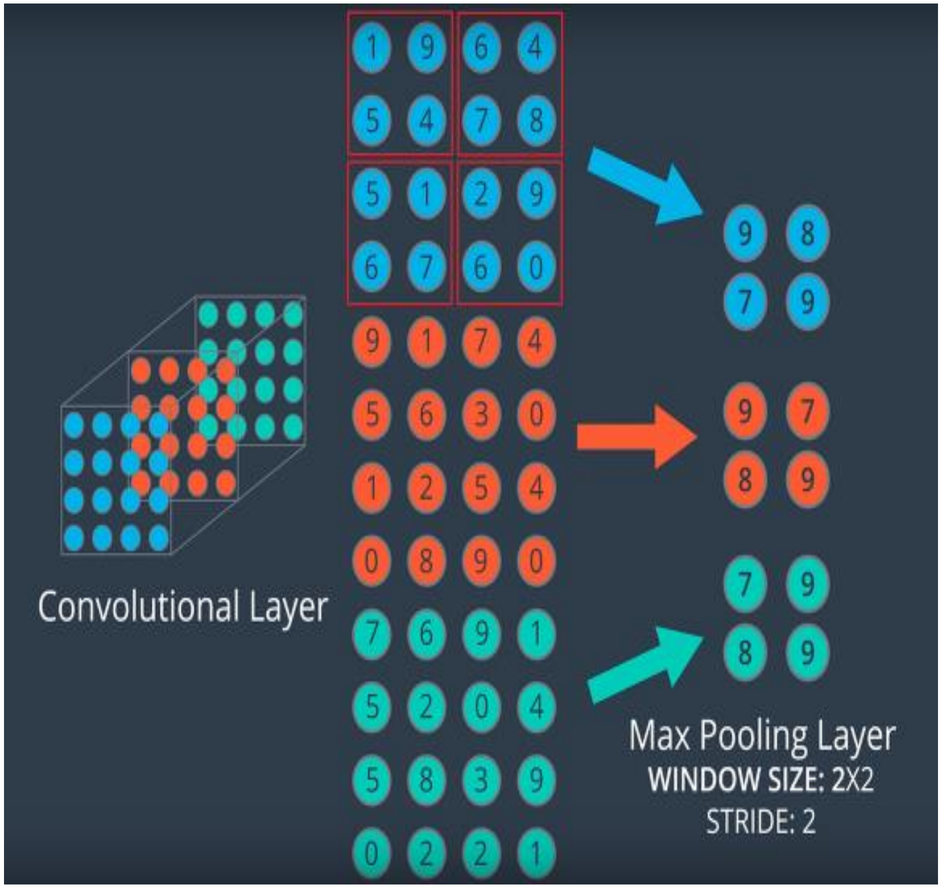
Class Activation Map (클래스 활성맵)

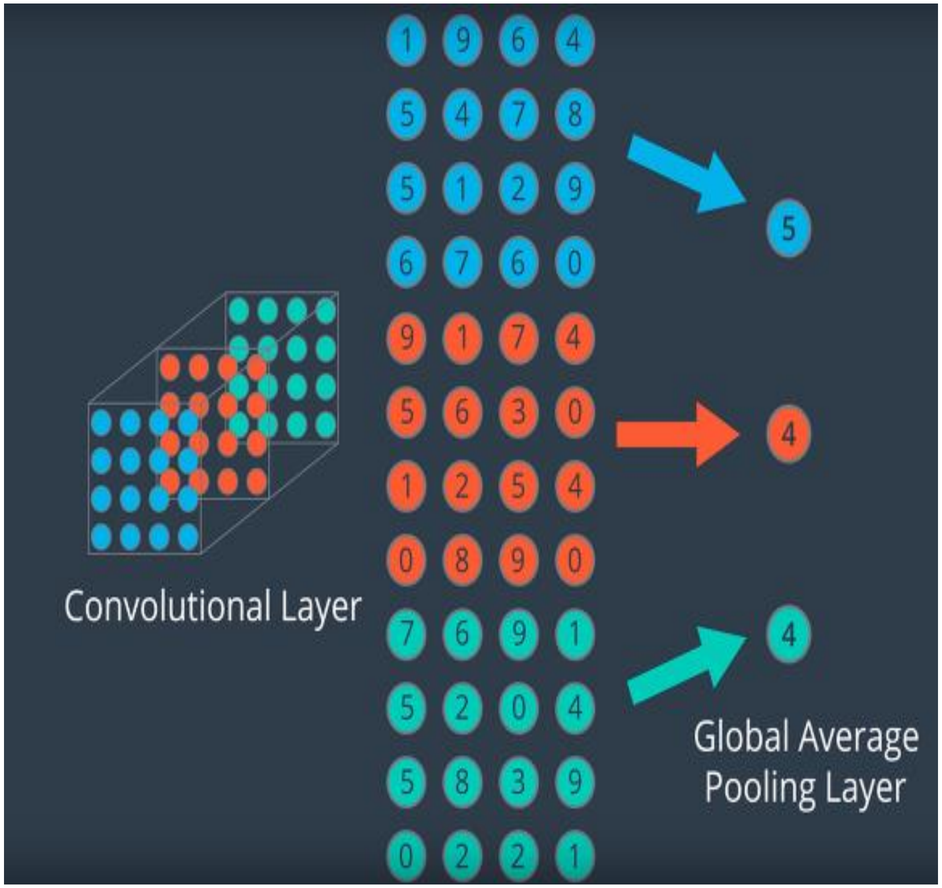
● 약한 지도학습(Weakly Supervised Learning): 약한 정답을 제시한다.



Class Activation Map의 원리

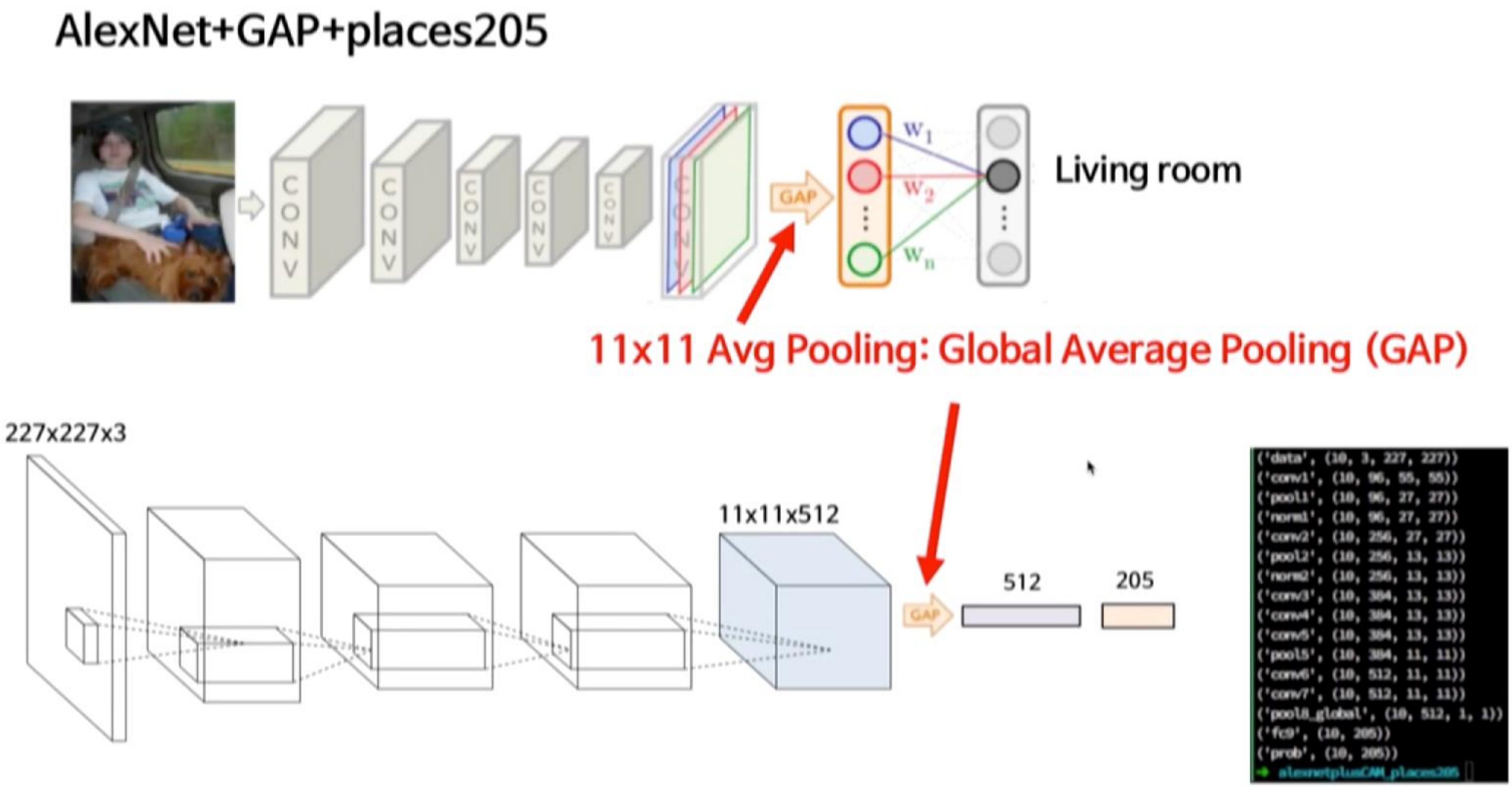
- 맥스풀링(MP)과 글로벌 평균풀링(GAP)





Class Activation Map

- 약지도학습(Supervised Learning): Global Average Pooling



Class Activation Map

- 출력층의 가중치를 다시 합성곱 특성맵에 투영함으로써 중요한 이미지 영역을 식별
- CAM은 각 클래스에 대해서 생성될 수 있다.
- 주어진 이미지에 대해 클래스가 강조되는 영역이 다르다.

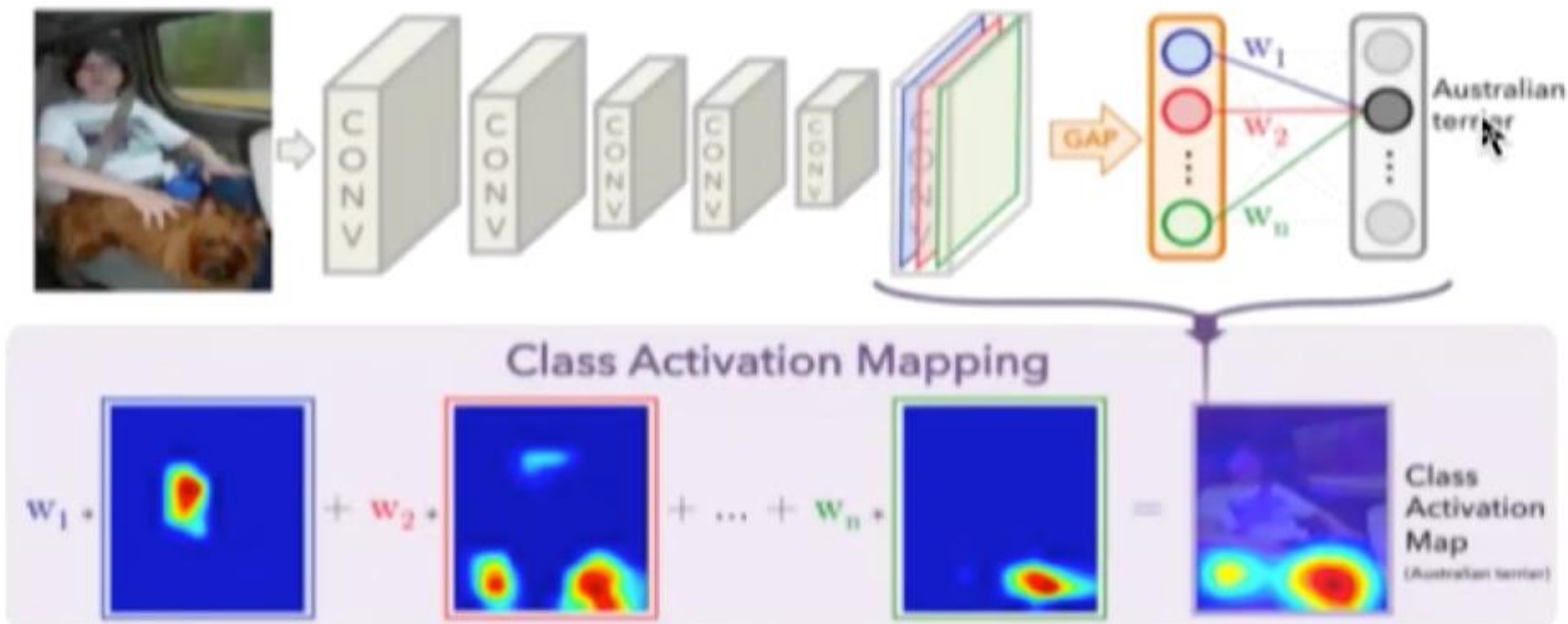
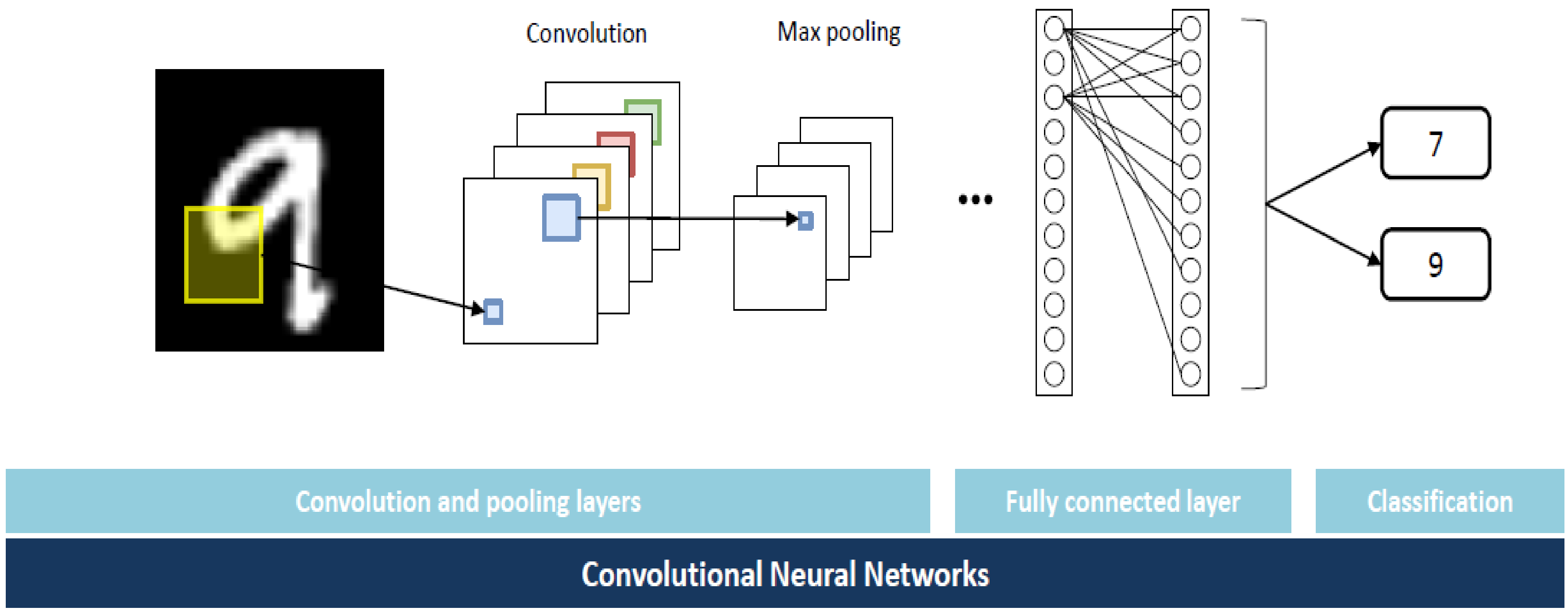


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

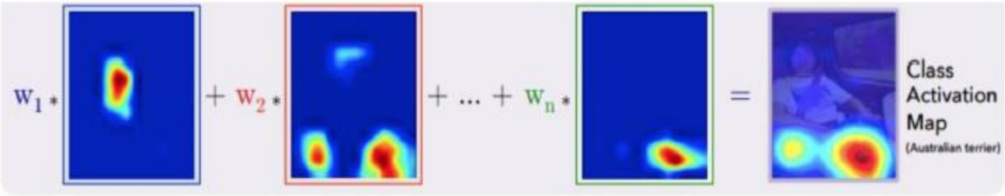
Class Activation Map

- 전결합층이 문제

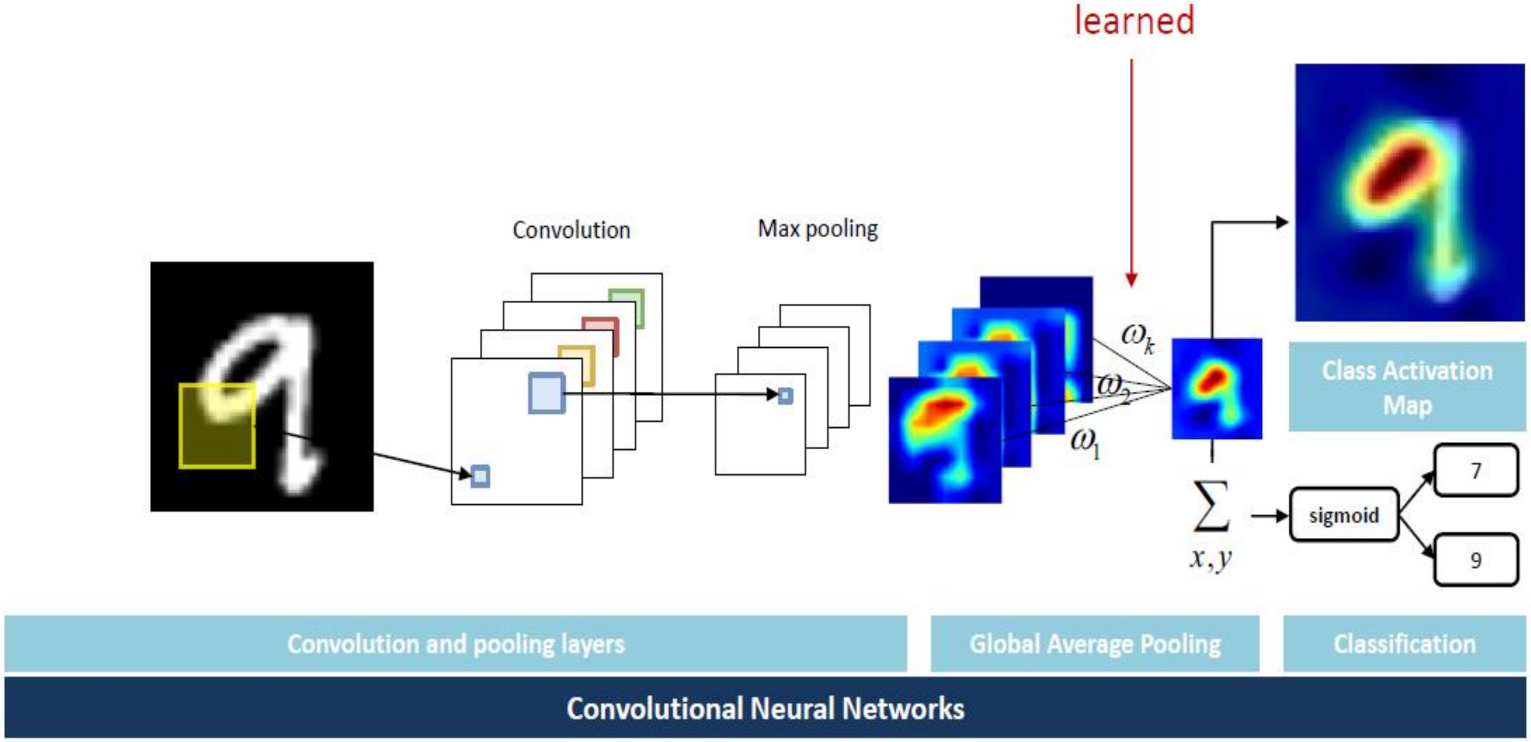


Class Activation Map

- 글로벌 평균 풀링



- Class Activation Map (CAM)
- (or Attention)

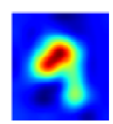


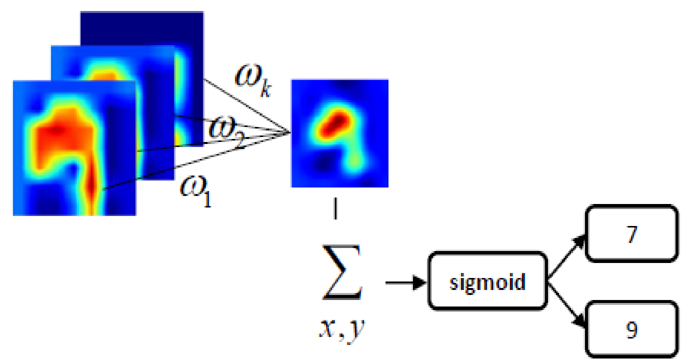
Class Activation Map

- 구현

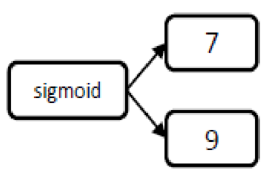
```
## global average pooling
SUM = tf.zeros([1,7,7,1])
for i in range(int(weights['w'].shape[0])):
    SUM = tf.add(weights['w'][i]*tf.reshape(maxp2[:,:,:,i],(-1,7,7,1)), SUM)

attention = tf.reduce_sum(SUM, axis = (3))
output = tf.reduce_sum(attention, axis = (1,2))
output = tf.nn.sigmoid(output)
output = tf.stack(((1-output), output),1)
```

$$S_c = \sum_k \omega_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \underbrace{\sum_k \omega_k^c f_k(x,y)}$$





$$P_c = \frac{\exp(S_c)}{\sum_c \exp(S_c)}$$



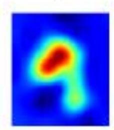
Class Activation Map

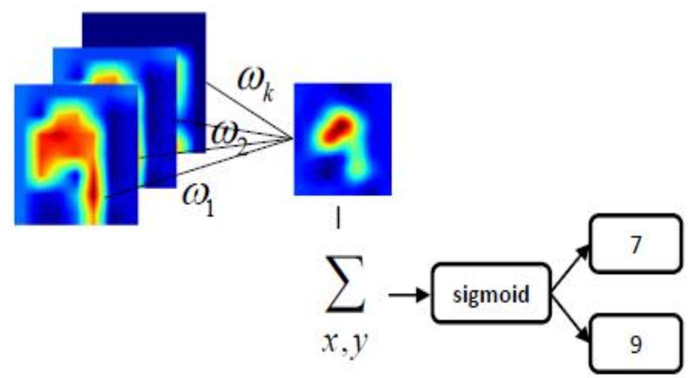
- 구현

```python
## global average pooling
avg = tf.reduce_mean(maxp2, axis = (1,2))
output = tf.matmul(avg, weights['output'])
```

```python
maps, pred = net(x, weights, biases)
loss = tf.nn.softmax_cross_entropy_with_logits(labels = y, logits = pred)
loss = tf.reduce_mean(loss)
```

$$S_c = \sum_k \omega_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k \omega_k^c \, f_k(x,y)$$





$$P_c = \frac{\exp(S_c)}{\sum_c \exp(S_c)}$$

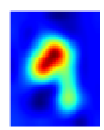
Class Activation Map

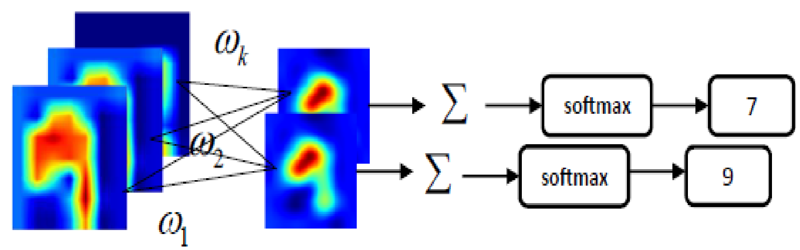
- 더 나은 구현법

```python
## global average pooling
avg = tf.reduce_mean(maxp2, axis = (1,2))
output = tf.matmul(avg, weights['output'])
```

```python
maps, pred = net(x, weights, biases)
loss = tf.nn.softmax_cross_entropy_with_logits(labels = y, logits = pred)
loss = tf.reduce_mean(loss)
```

$$S_c = \sum_k \omega_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \underbrace{\sum_k \omega_k^c \, f_k(x,y)}$$





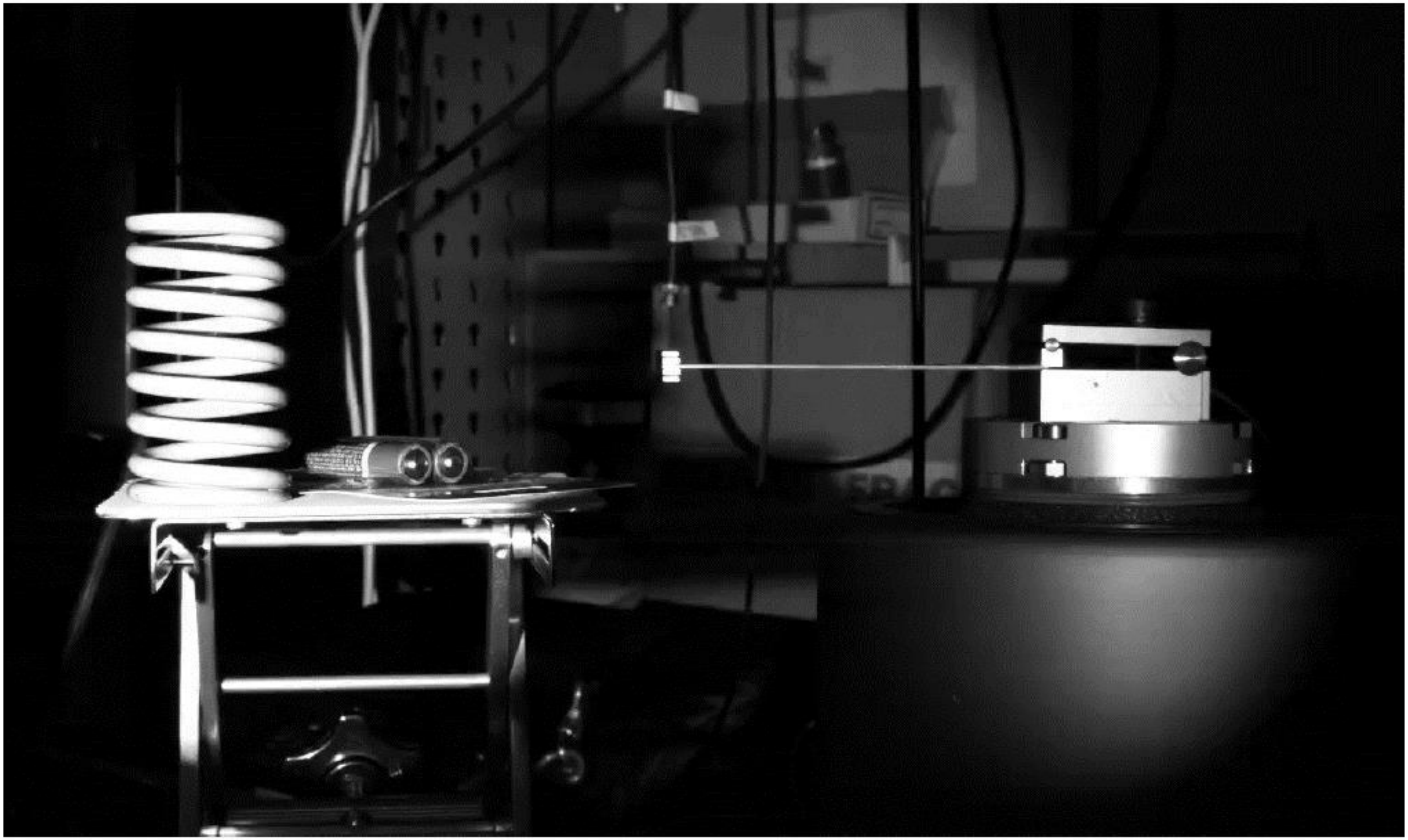
$$P_c = \frac{\exp(S_c)}{\sum_c \exp(S_c)}$$

Class Activation Map

- MNIST 예제

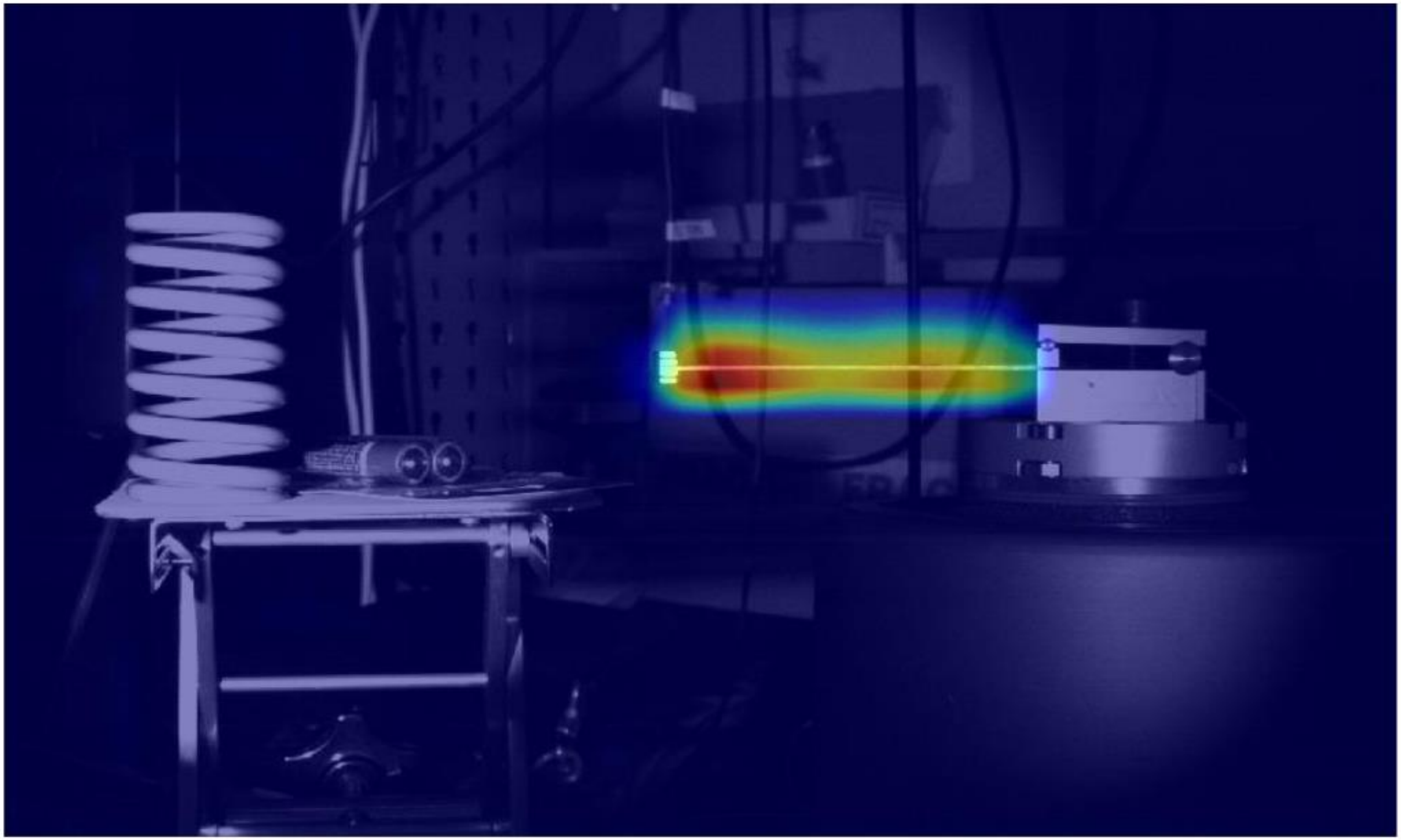
Class Activation Map

- 칸틸레버(외팔보): 한쪽 끝은 고정되고 다른쪽 끝은 자유로운 들보



Class Activation Map

- 칸틸레버(외팔보): 한쪽 끝은 고정되고 다른쪽 끝은 자유로운 들보



Class Activation Map

- 약한 지도 학습 (약한 레이블링으로 학습)

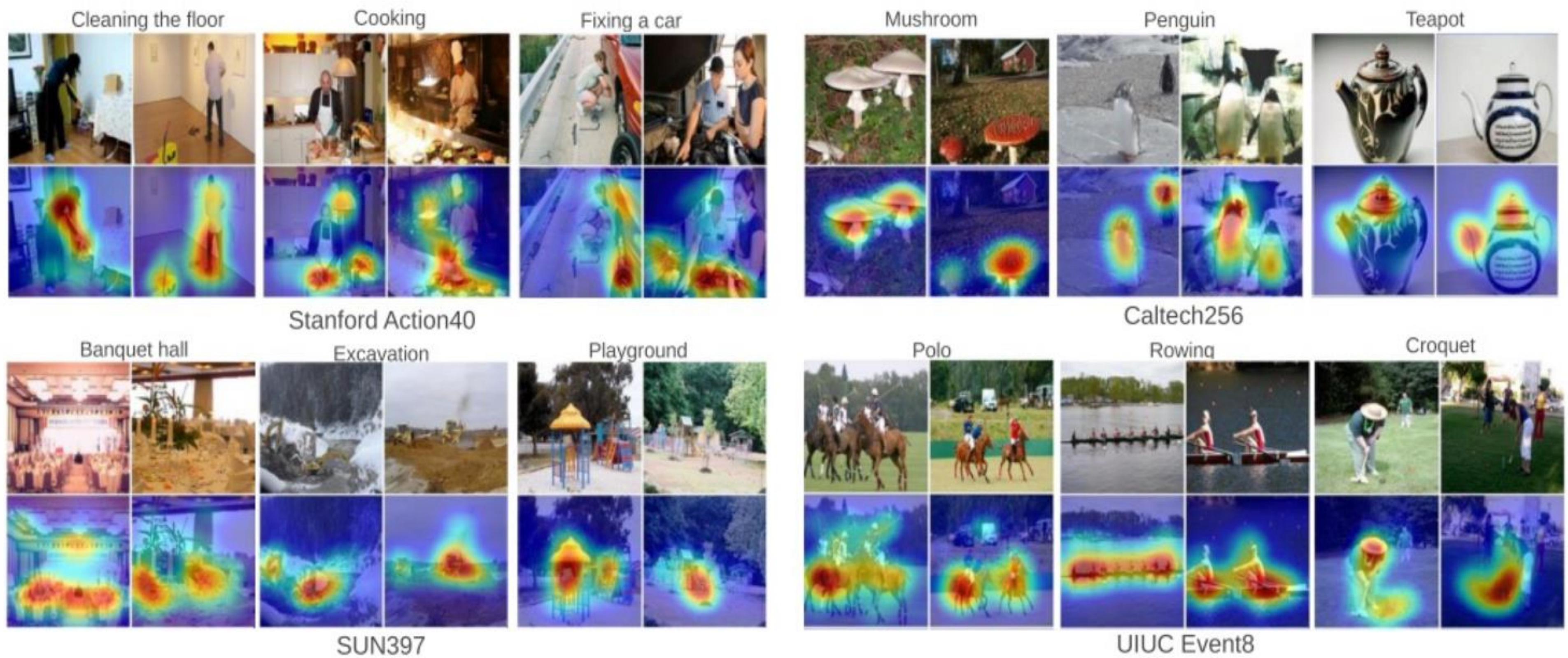


Figure 8. Generic discriminative localization using our GoogLeNet-GAP deep features (which have been trained to recognize objects). We show 2 images each from 3 classes for 4 datasets, and their class activation maps below them. We observe that the discriminative regions of the images are often highlighted e.g., in Stanford Action40, the mop is localized for *cleaning the floor*, while for *cooking* the pan and bowl are localized and similar observations can be made in other datasets. This demonstrates the generic localization ability of our deep features.

Class Activation Map

- 약한 지도 학습 (추상적 레이블링 또는 텍스트의 유무로 학습)

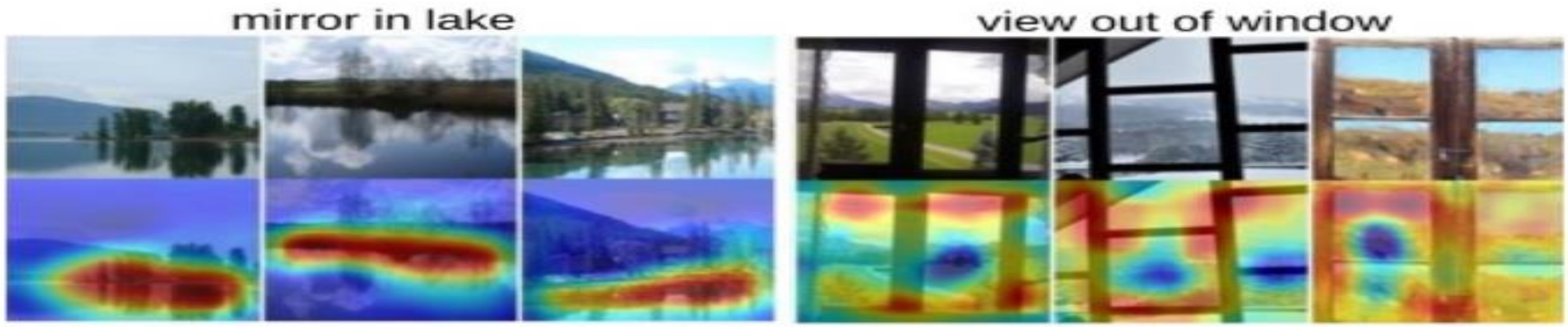


Figure 10. Informative regions for the concept learned from weakly labeled images. Despite being fairly abstract, the concepts are adequately localized by our GoogLeNet-GAP network.

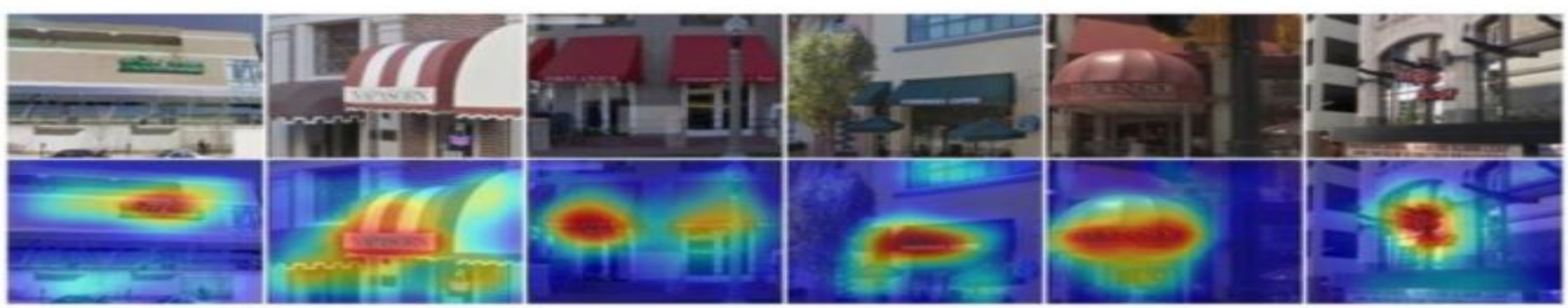


Figure 11. Learning a weakly supervised text detector. The text is accurately detected on the image even though our network is not trained with text or any bounding box annotations.