

---

## 14.9 객체 탐지

---

## 14.9.1 옛날 방법 - CNN + NMS



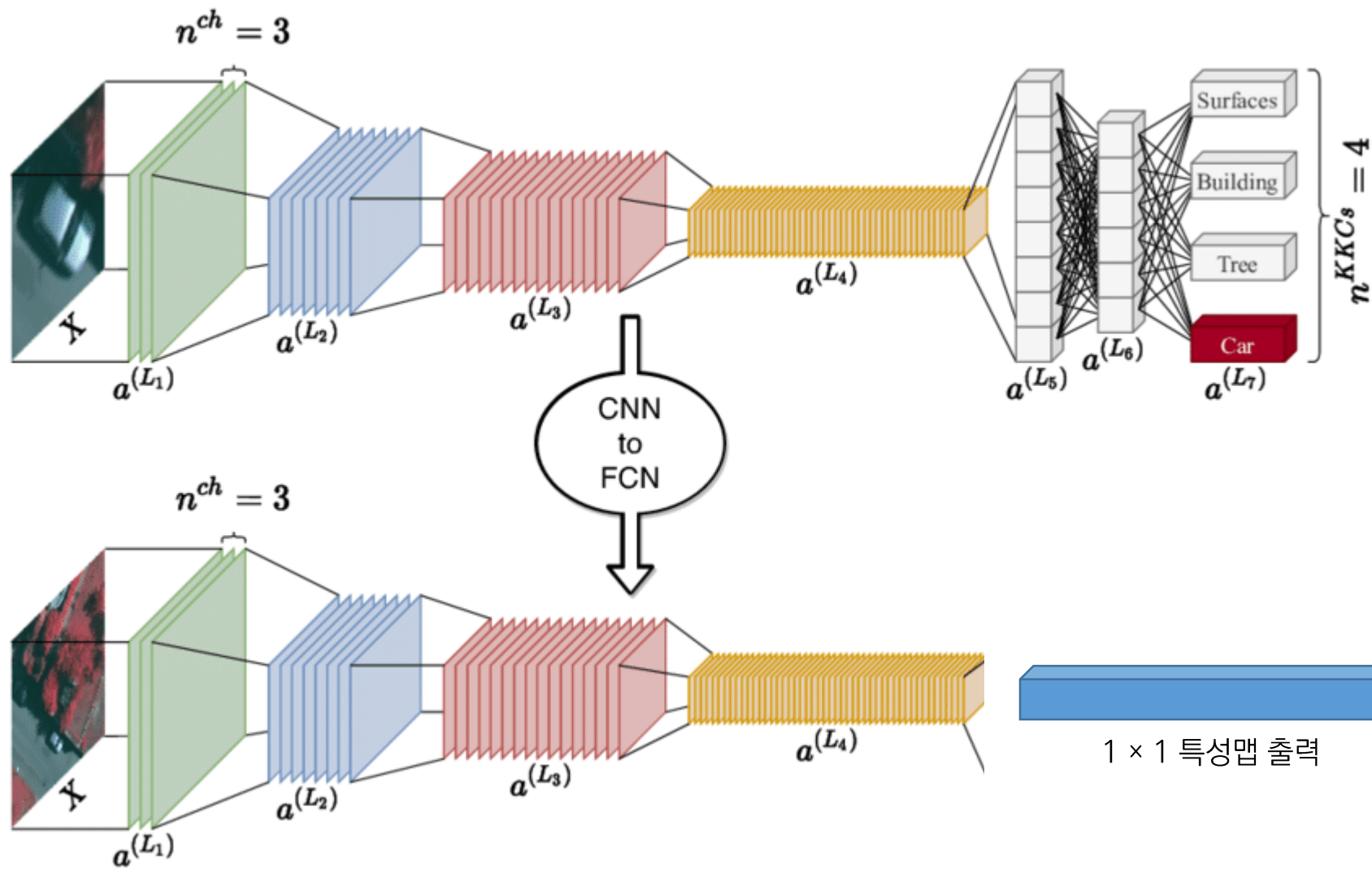
같은 꽃을 인지하더라도 IOU 차이가 있음

- 1) CNN이 이동하면서 순차적으로 꽃을 인지한다.  
동시에 여러 개 바운딩 박스를 생성하여 각 박스마다 IOU를 계산한다
- 2) 존재여부가 임계값 이하의 바운딩 박스를 모두 삭제한다  
(대체로  $\text{IOU} < 50\%$ )
- 3) 같은 꽃을 인지한 바운딩 박스에 대해 IOU가 가장 높은 박스만 살린다
- 4) 더는 제거할 바운딩 박스가 없을 때까지 반복한다

→ CNN 한번 끝...이제 CNN 옆으로 하나 이동시키는 것...

→ CNN을 몇십 몇백번 실행해야 해서 많이 느리다

## 14.9.2 좀 더 빠른 CNN? - 완전 합성곱 신경망 (FCN)



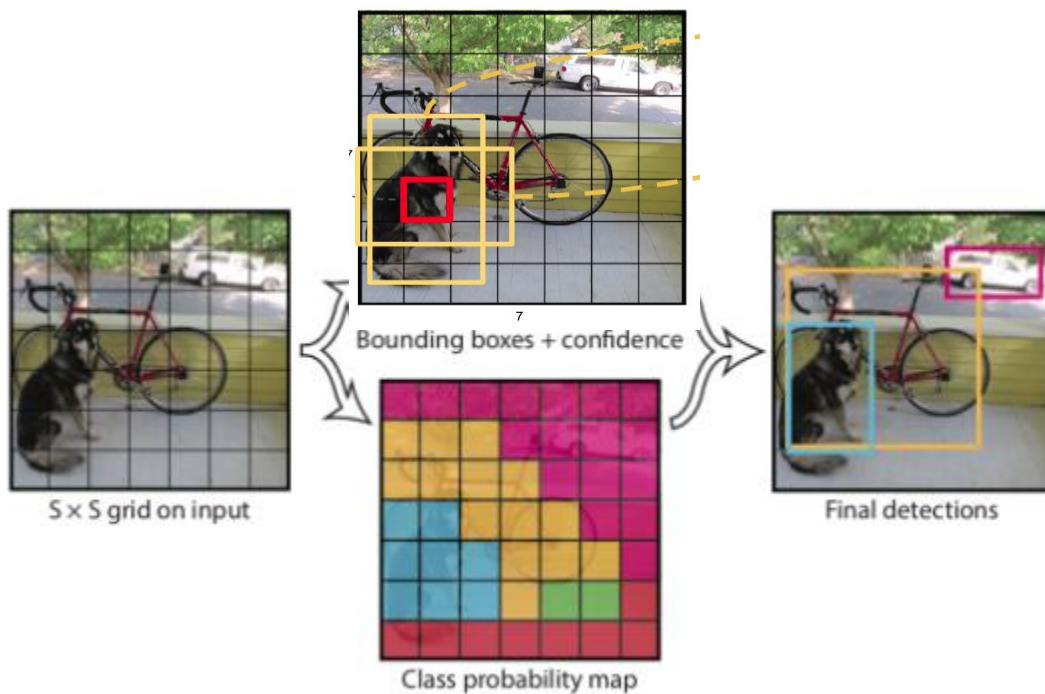
마지막 밀집층에서  
각 특성 맵에 대해 가중치 값 계산  
↓  
가중치 계산에 자원 소모

여러 필터를 사용하여  
'한번'의 합성곱 수행  
↓  
상대적으로 줄어든 자원 소모

# 14.9.3 한번의 합성곱 계산? You Only Look Once!!

## ■ YOLO와 기존 객체탐지 모델과 가장 다른 점

- ① Regional Proposal ② Classification 두 단계였던 것을
- ① 단계 제거하고 한번에 Object Detection 수행! (바운딩 박스, Classification 동시 수행)

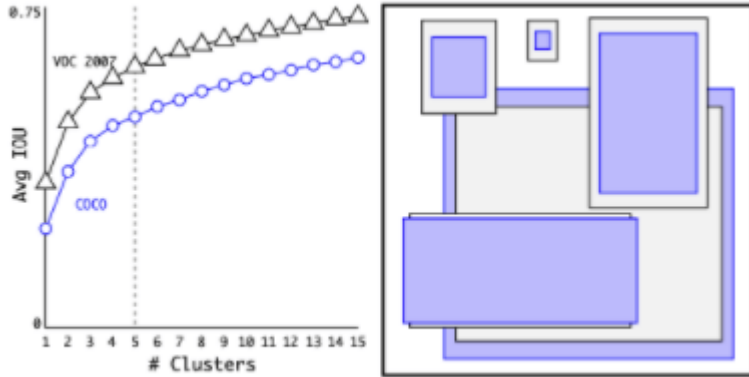


- 1) 입력 이미지를  $S \times S$  격자로 나눈다
- 2) 하나의 격자마다 B개의 Bounding Box 예측 (논문에서는 2개)
- 3) Confidence(신뢰도) 계산  
= 해당 그리드에 물체가 있을 확률  $\times$  예측한 Box와 Ground Truth 박스와 겹치는 영역  
=  $\text{Pr}(\text{Object}) \times \text{IOU}$
- 4) 각 그리드마다 C개의 클래스에 대해 각 클래스일 확률을 계산  
=  $\text{Pr}(\text{Class}_i | \text{Object})$
- 5) 신뢰도  $\times$  확률 계산 =  $\text{Pr}(\text{Object}) \times \text{IOU} \times \text{Pr}(\text{Class}_i | \text{Object})$   
=  $\text{IOU} \times \text{Pr}(\text{Class}_i)$  = 해당 박스가 특정 클래스일 확률!!!
- 6) NMS 수행 : 중복된 Box 제거  
(하나의 Box만이 하나의 클래스에 속할 때까지)

→ 각 클래스마다 Box가 만들어짐!!

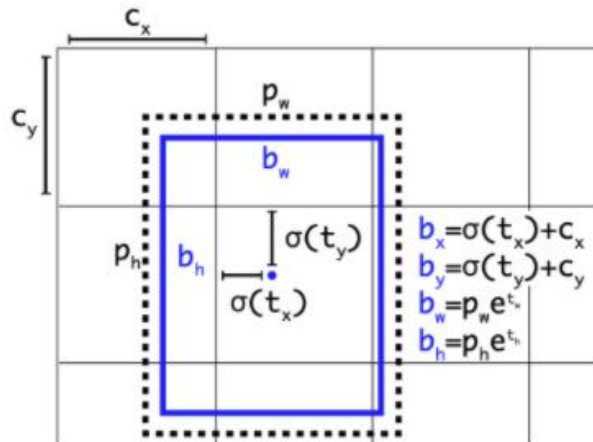
# 14.9.4 좀 더 나은 YOLOv3

## ■ Anchor Box 사용



- ① 사전에 크기와 비율이 모두 결정되어 있는 박스를 전제
- ② 학습을 통해 이 박스의 위치나 크기를 세부 조정
  - 처음부터 중심점의 좌표, 너비, 높이를 결정하는 것보다 훨씬 안정적
  - 대체로 5개 박스로 설정하는 것이 Precision, Recall 측면에서 유리

## ■ Bounding Box Regression



- ① Box에 대해 절대 좌표를 이용하는 것이 아니라 상대 좌표 이용 (학습에 유리)
- ② 상대 좌표에 대해 지수함수와 비율을 사용하여 Box 크기, 위치 조정
  - 왼쪽 식에서 결국  $\partial(t_x)$ ,  $\partial(t_y)$ ,  $e^{(t_w)}$ ,  $e^{(t_h)}$ 를 학습
  - 대체로 5개 박스로 설정하는 것이 Precision, Recall 측면에서 유리

# 참고. YOLO에 대해 설명을 잘 해둔 사이트

---

갈아먹는 Object Detection [5] Yolo: You Only Look Once : <https://yeomko.tistory.com/19?category=888201>

갈아먹는 Object Detection [8] yolov2, yolo9000 : <https://yeomko.tistory.com/47>

러시아 형님들의 친절한 YOLO 해설 :

[https://docs.google.com/presentation/d/1aeRvtKG21KHdD5lg6Hgyhx5rPq\\_ZOsGjG5rJ1HP7BbA/pub?start=false&loop=false&delayms=3000&slide=id.g137784ab86\\_4\\_4611](https://docs.google.com/presentation/d/1aeRvtKG21KHdD5lg6Hgyhx5rPq_ZOsGjG5rJ1HP7BbA/pub?start=false&loop=false&delayms=3000&slide=id.g137784ab86_4_4611)

YOLOv3 : [https://jjeamin.github.io/darknet\\_book/part1\\_paper/yolov3.html](https://jjeamin.github.io/darknet_book/part1_paper/yolov3.html)

YOLOv4도 있음 : [https://jjeamin.github.io/darknet\\_book/part1\\_paper/yolov4.html](https://jjeamin.github.io/darknet_book/part1_paper/yolov4.html)