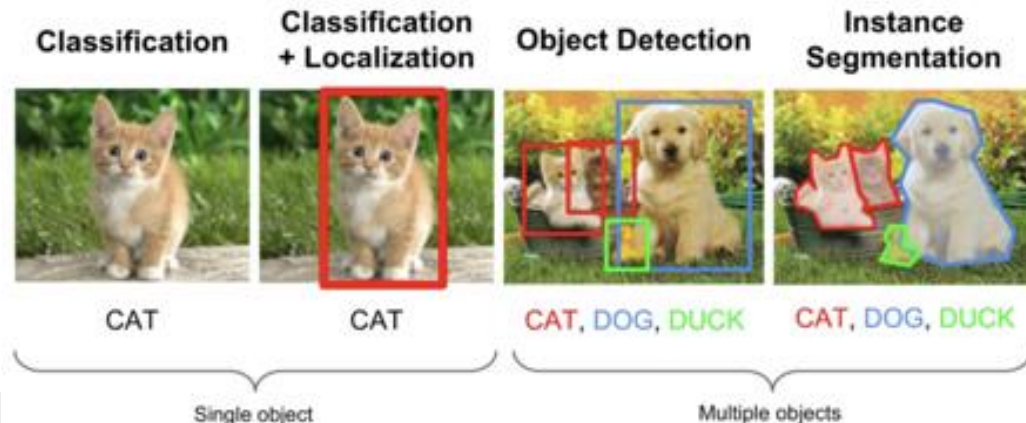


YOLO(You Only Look Once)

□ Object Detection

■ 정의

- 한 물체(single object)가 아닌 여러 물체(Multiple objects)에 대해 어떤 물체인지 클래스를 분류하는 Classification 문제와, 그 물체가 어디 있는지 박스를 (Bounding box) 통해 위치 정보를 나타내는 Localization 문제를 모두 포함한다.

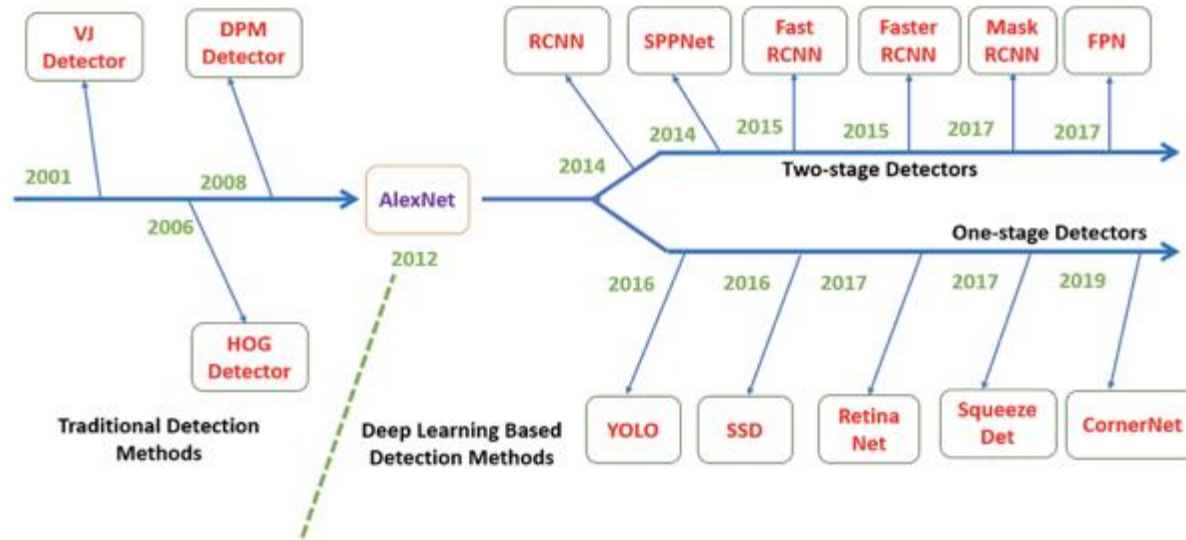


- 즉 Object Detection은 Multiple Object에 대한 Multi-Labeled Classification + Bounding Box Regression(Localization) 라고 정리할 수 있다.



YOLO(You Only Look Once)

- Deep Learning을 이용한 Object Detection은 2013년 11월을 기준으로 나타나기 시작하였으며 크게 1-stage Detector와 2-stage Detector로 구분할 수 있다.



- Object Detection문제는 물체를 식별하는 Classification 문제와, 물체의 위치를 찾는 Localization 문제를 합한 것인데, 1-stage Detector는 이 두 가지 task를 동시에 행하는 방법이고, 2-stage Detector는 이 두 문제를 순차적으로 행하는 방법이다.

Object Detection = Classification + Localization

- 1-stage Detector가 비교적 빠르지만 정확도가 낮고 2-stage Detector가 비교적 느리지만 정확도가 높다.
 - 2-stage Detector 논문 : RCNN, SPPNet, Fast RCNN, Faster RCNN, FPN
 - 1-stage Detector 논문 : YOLO, SSD, Retina Net, Squeeze Net, Corner Net



YOLO(You Only Look Once)

□ YOLO

■ 정의

- 네트워크의 최종 출력단에서 경계박스를 위치 찾기와 클래스 분류가 동시에 이루어지는 방식
- 단 하나의 네트워크가 한번에 특징도 추출하고, 경계박스도 만들고, 클래스도 분류한다.
- 그러므로 간단하고 빠르다.



- 좌측의 입력 영상이 네트워크를 통과하면 중앙의 2개의 데이터를 얻는다. 이것이 네트워크의 최종 출력이다.



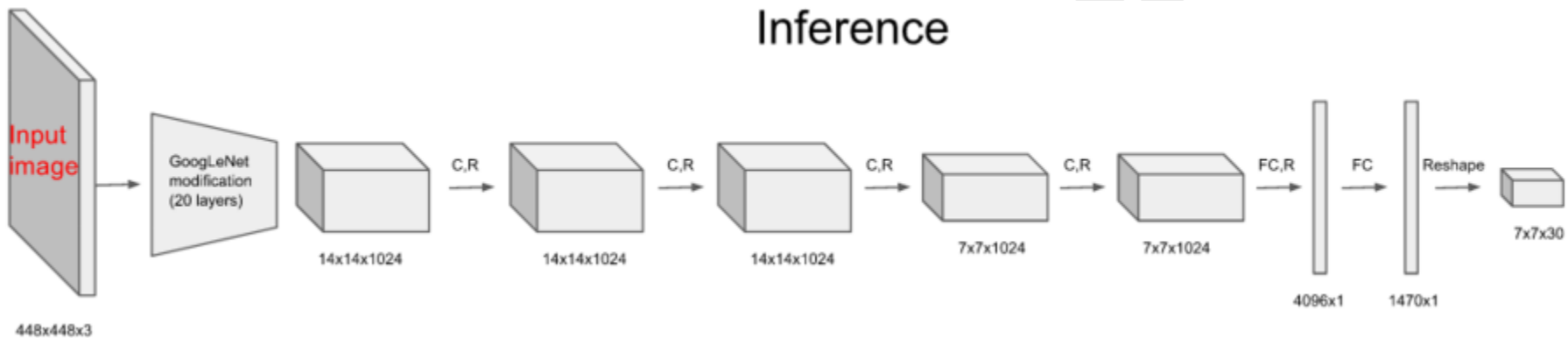
YOLO(You Only Look Once)

- 네트워크는 영상을 7x7 그리드로 나눈다. 각 그리드에서 중심을 그리드 안쪽으로 하면서 크기가 일정하지 않은 경계박스를 2개씩 생성한다. 그리드 셀이 $7 * 7 = 49$ 개이므로 경계 박스는 총 98개가 만들어 진다.
- 이 중 경계 박스 안쪽에 어떤 오브젝트가 있을 것 같다고 확신(confidence score)할수록 박스를 굵게 그려준다. 굵은 경계 박스들만 남기고 얇은 경계박스(어떤 오브젝트도 없는 것 같다고 생각되는 것들)을 지운다. 남은 후보 경계 박스들을 NMS(Nom-maximal suppression 비-최대값 억제) 알고리즘을 이용해 선별하면 우측의 이미지 처럼 3개만 남게 된다.
- 경계 박스의 색깔은 클래스를 의미한다. 중앙의 아래쪽에 이미지가 7x7의 격자(그리드)로 분할되어 총 49개의 그리드 셀이 만들어졌다. 각 그리드 셀은 해당 영역에서 제안한(proposal) 경계 박스안의 오브젝트가 어떤 클래스인지를 컬러로 표현하고 있다.
- 그러므로 최종적으로 남은 3개의 경계 박스 안에 어떠한 클래스가 있는지 알 수 있다. 그래서 우측의 최종 결과를 얻는다.
- 네트워크가 생성하는 경계박스의 숫자는 그리드 셀의 2배이다. 그리드 셀이 49개이므로 경계 박스는 총 98개가 만들어 진다. ROI 혹은 오브젝트 후보라고 할 수 있는 이 경계박스들은 스레시홀드(0.5)보다 작으면 지워준다.



YOLO(You Only Look Once)

■ 네트워크 구조

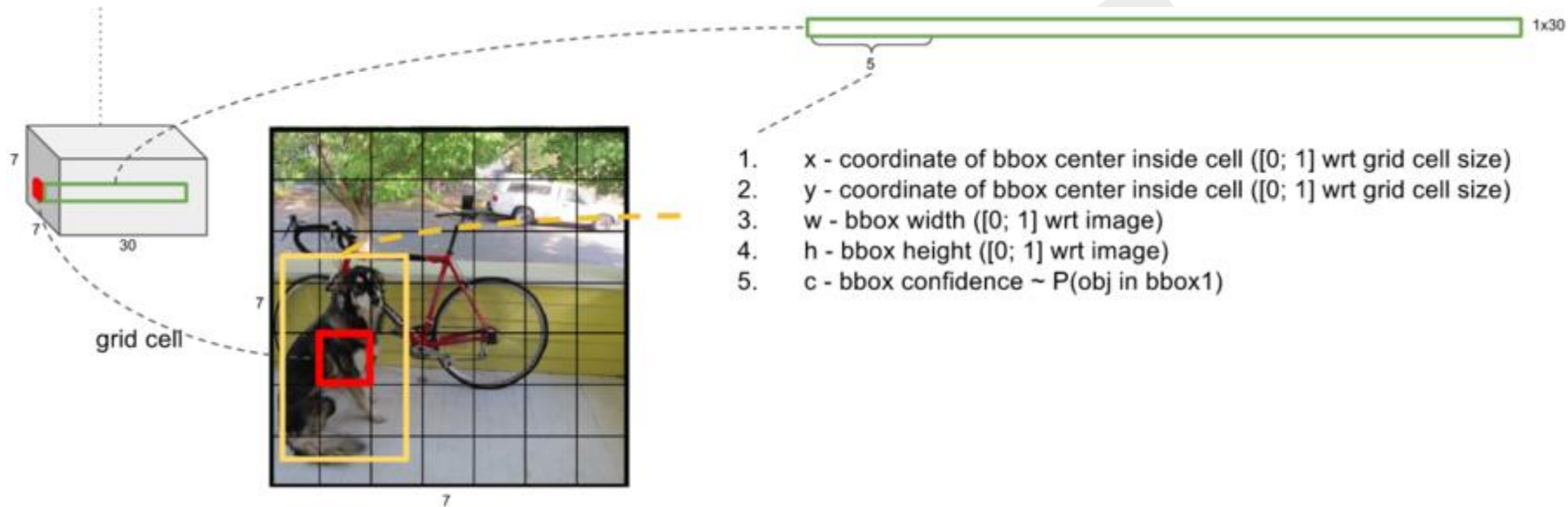


- GoogLeNet을 약간 변형시켜서 특징 추출기로 사용했다. 이후 컨볼루션 레이어 4회, 풀 커넥션 레이어 2번 하고 사이즈를 7x7x30으로 조정하면 끝난다.
- 이 마지막 특징 데이터 7x7x30 가 바로 예측 결과이며 이 안에 경계박스와 클래스 정보 등 모든 것이 들어 있다.



YOLO(You Only Look Once)

■ 첫번째 예측(첫번째 경계박스, 노란색)



- 왼쪽 빨간점으로 표시한 부분은 7x7 그리드셀중에 하나로 이미지에서 개의 중앙 부분에 해당한다. 그리고 빨간색 박스보다 큰 노란색 박스가 바로 빨간색 그리드셀에서 예측한 경계 박스이다.
- 7x7 은 영상을 7x7 의 격자로 나눈것이다. 30개의 채널은 (경계 박스의 정보 4개 , 경계 박스안에 오브젝트가 있을 확률(confidence)) x 2, 어떤 클래스일 확률 20개 로 구성된다.
 - 경계 박스 정보 x, y : 노란색 경계 박스의 중심이 빨간 격자 셀의 중심에서 어디에 있는가를 나타낸다.
 - 경계 박스 정보 w, h : 노란색 경계 박스의 가로 세로 길이가 전체 이미지 크기에 어느 정도 크기를 갖는가를 나타낸다.
- 만약 경계박스가 위의 그림처럼 되었다면 x, y는 모두 0.5 정도이고 w, h는 각각 2/7, 4/7 정도가 될 것이다.

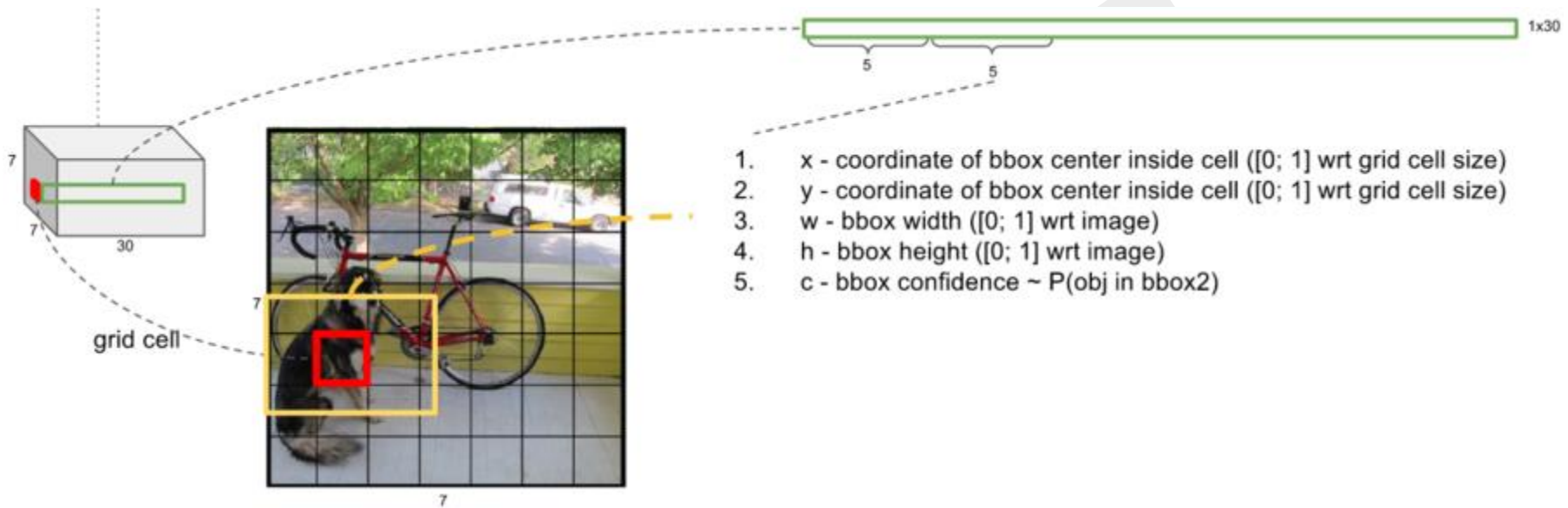
YOLO(You Only Look Once)

- 노란색 경계 박스는 반드시 그 중심이 빨간 그리드 셀 안에 있어야 하며, 가로와 세로길이는 빨간 그리드 셀보다 작을 수도 있고 그림처럼 클 수도 있다. 또한 정사각형일 필요도 없다. 빨간 그리드 셀 내부 어딘가를 중심으로 하여 근처에 있는 어떤 오브젝트를 둘러싸는 직사각형의 노란색 경계 박스를 그리는 것이 목표이다.
- 노란색 경계 박스가 바로 ROI , 오브젝트 후보이다. 이것을 2개 만든다.



YOLO(You Only Look Once)

■ 두번째 예측(두번째 경계박스, 노란색)



- 30개의 채널 중에서 앞에 5개는 1번째 경계박스의 정보가 있으며 다음의 5개에는 2번째 경계박스의 정보를 가지고 있다.
- 2번째 경계 박스는 1번째 것과 모양이 비슷할 수도 있고 다를 수도 있다. 네트워크가 학습하는 것이기 때문에 크기나 모양이 어떻게 나올지 모른다.



YOLO(You Only Look Once)

- 네트워크 예측데이터의 구조

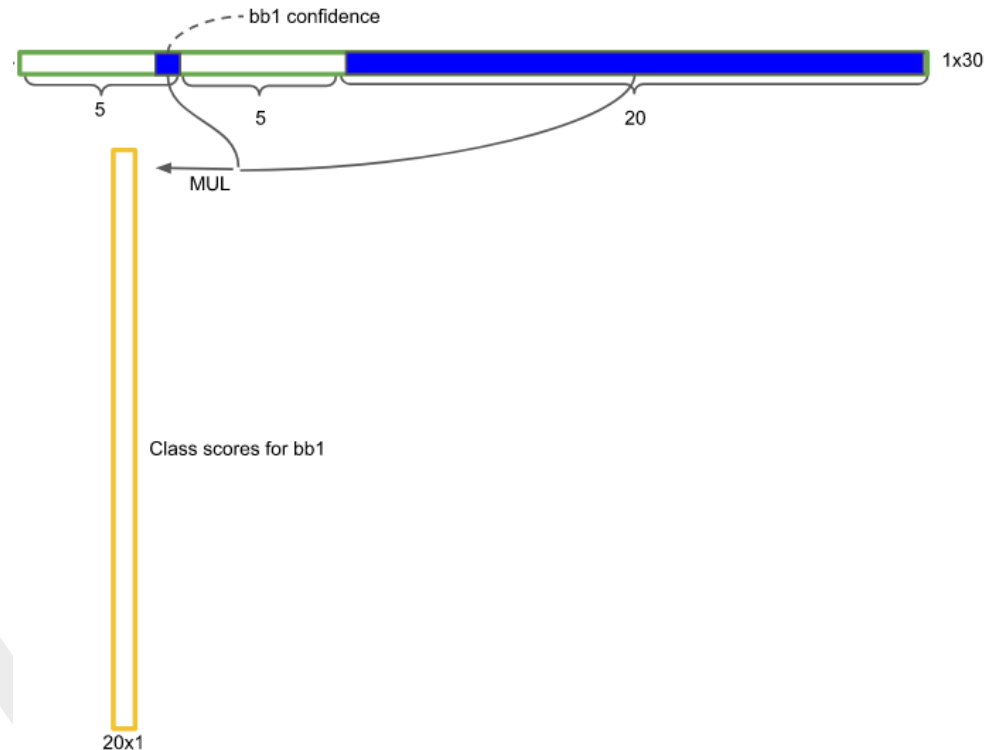


- 30개의 채널의 뒷부분에는 해당 그리드 셀(빨간색 박스)에 오브젝트가 만약 있다면 그것이 어떤 클래스일지에 대한 확률이 저장되어 있다.



YOLO(You Only Look Once)

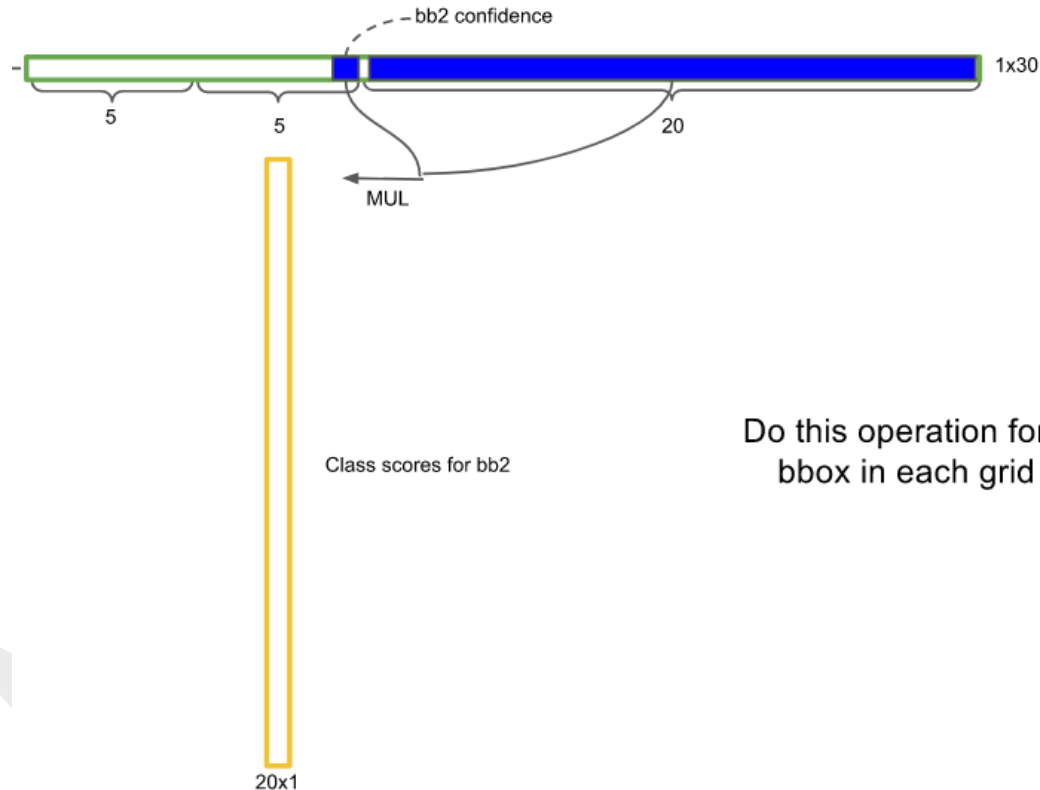
- 그리드 셀(빨간색 박스)에 오브젝트가 있다면 그것의 클래스를 분류하기(1번째 경계박스 분류)



- 각 그리드 셀에는 셀의 주위 혹은 셀 안쪽에 어떤 오브젝트가 있을 거라고 예측한 2개의 경계박스가 있다. 각 경계박스는 오브젝트가 있을 확률(confidence)을 가지고 있다. 이 스칼라 값(숫자 1개)을 그리드 셀의 클래스 분류 확률과 곱하면 경계박스의 클래스 분류 확률을 구할 수 있다. 이것이 위 그림의 긴 노란색 박스이다.
- bb1(bound box 1) 즉 첫번째 후보의 경계박스에 대한 클래스 점수표이다.
- 경계박스의 오브젝트가 있을 확률(confidence)가 매우 낮다면(0에 가깝다면) 그곳에 어떠한 클래스가 있는지에 대한 정보도 매우 낮아 지게 된다.

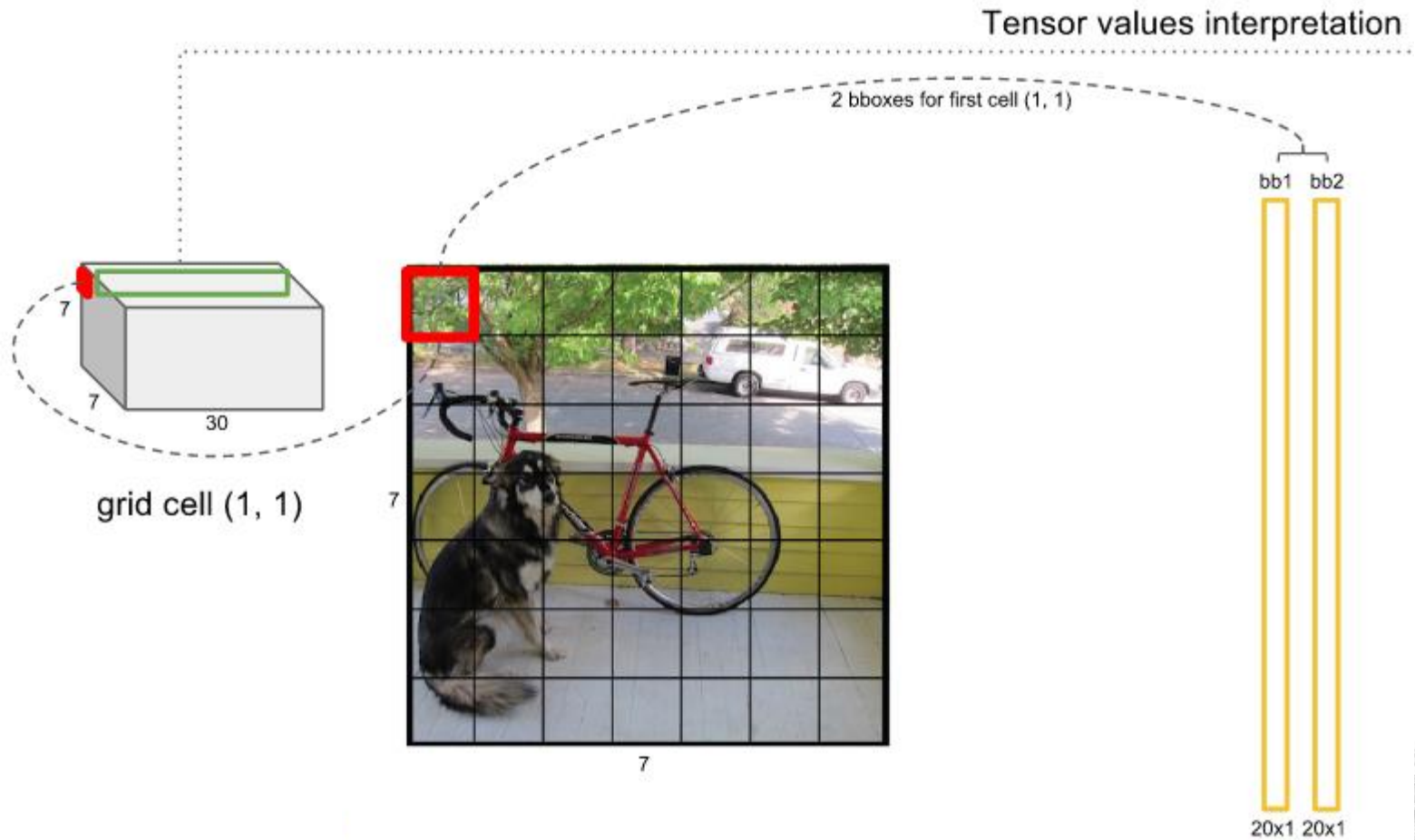
YOLO(You Only Look Once)

- 모든 그리드 셀은 경계박스가 2개씩 있으므로 2번째 경계박스에 대해서도 클래스를 분류한다.(2번째 경계박스 분류)



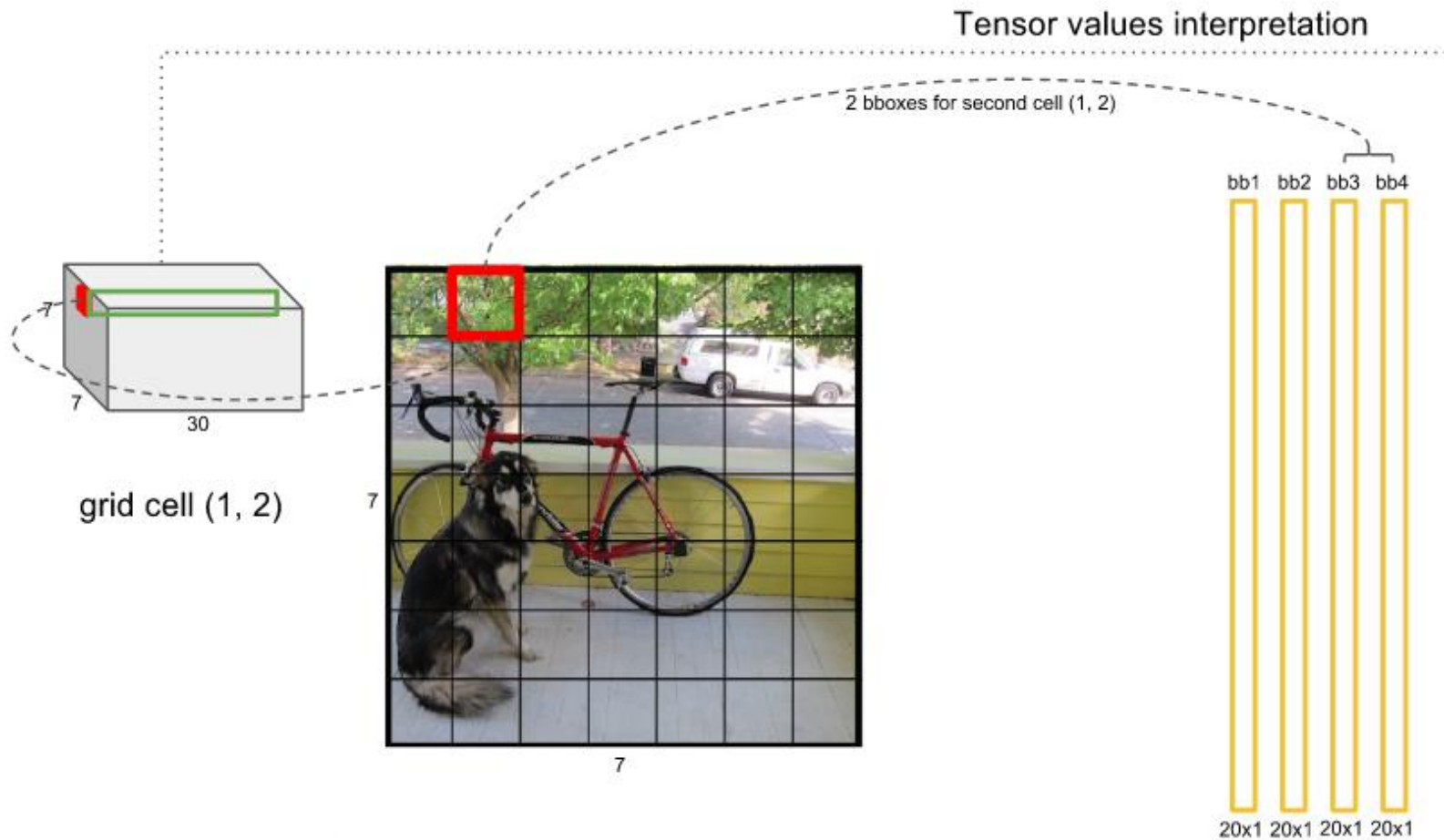
YOLO(You Only Look Once)

- 각 그리드 셀 하나에는 2개의 경계박스 후보가 있고 각 경계박스에 대한 클래스 분류 점수를 얻는다.
- 두번째 그리드 셀에서 클래스 분류 점수 2개를 얻는다.



YOLO(You Only Look Once)

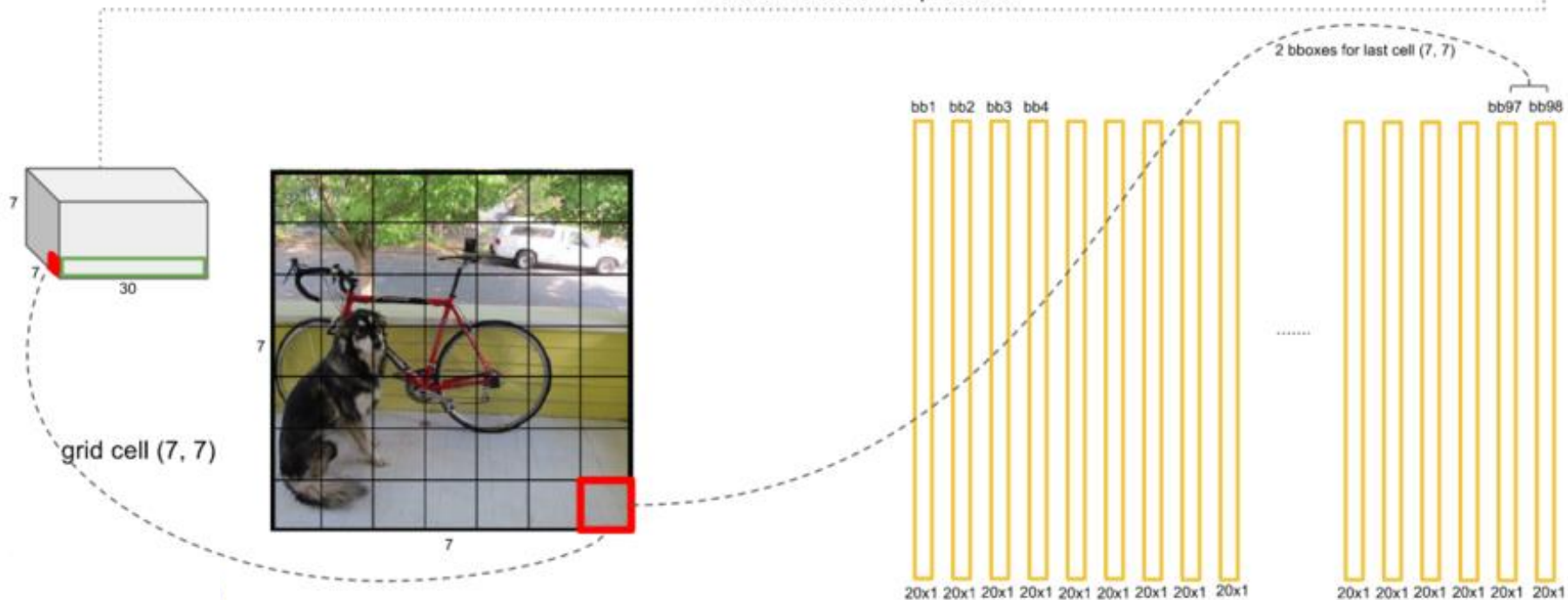
- 두번째 그리드 셀에서 클래스 분류 점수 2개를 얻는다.



YOLO(You Only Look Once)

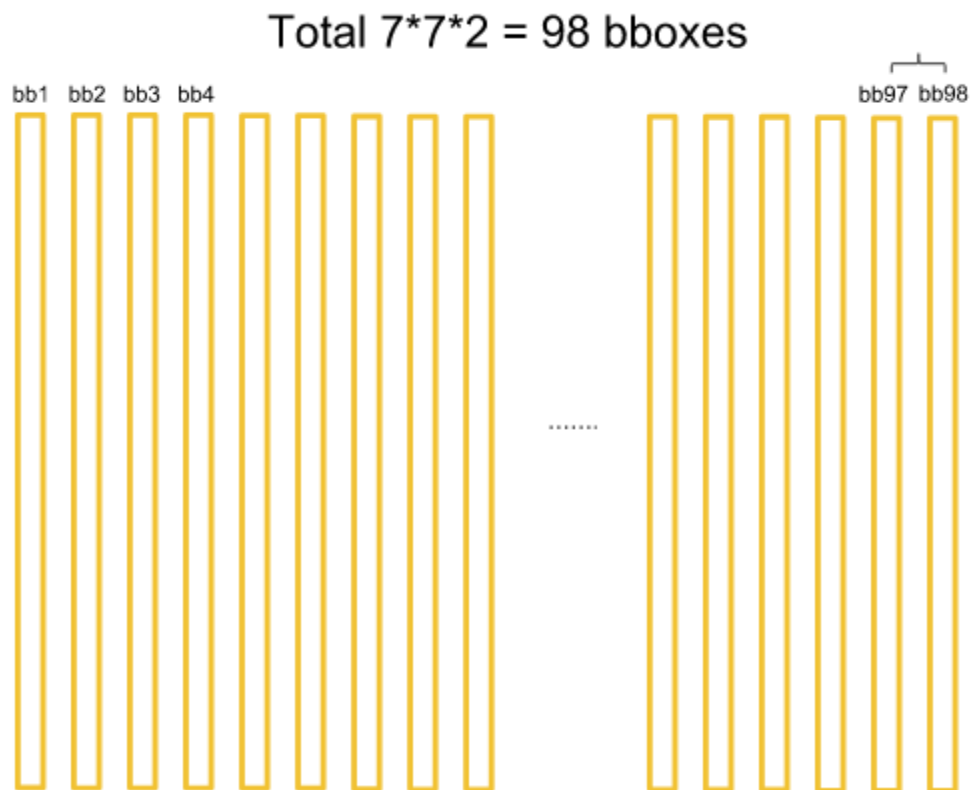
- 마지막 그리드 셀에서 클래스 분류 점수 2개를 얻는다.

Tensor values interpretation



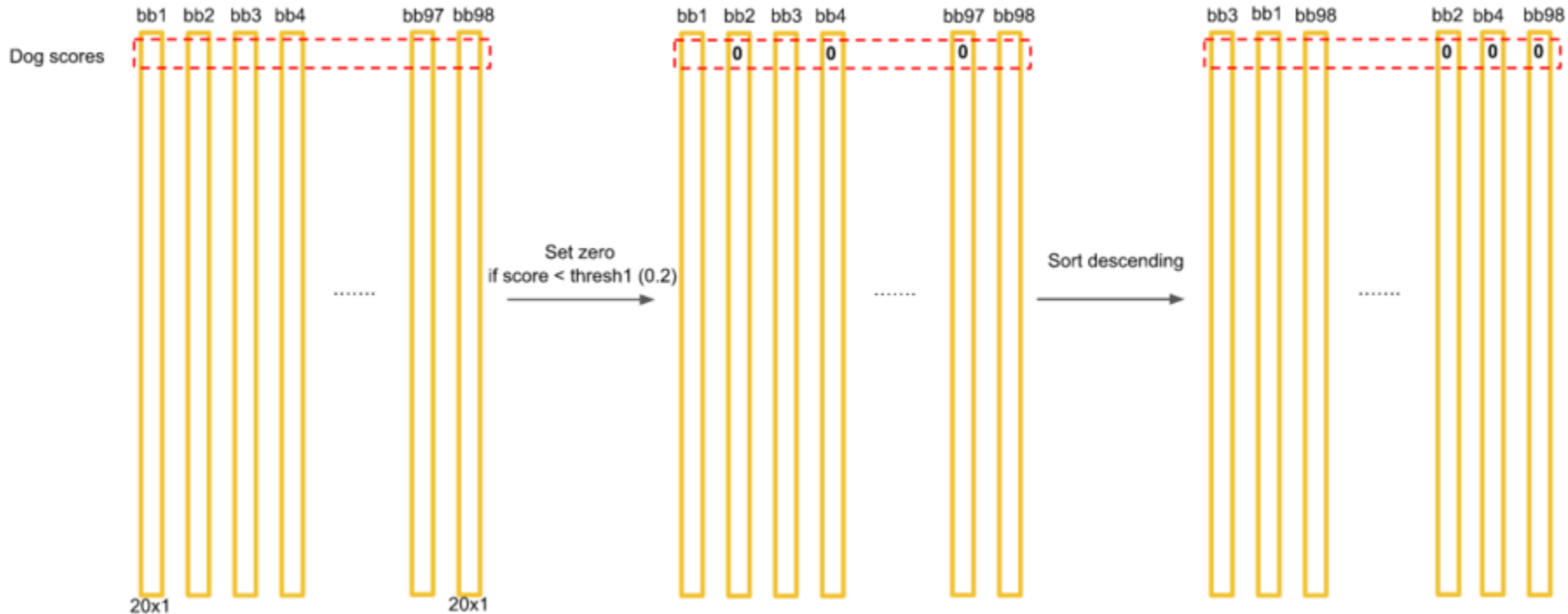
YOLO(You Only Look Once)

- 경계 박스가 7x7개이므로 클래스 분류 점수는 7x7x2 개로 총 98개가 된다.



YOLO(You Only Look Once)

- 98개의 클래스 분류 점수중에서 스레시홀드(0.2) 보다 작은 것은 전부 0으로 채워준다.

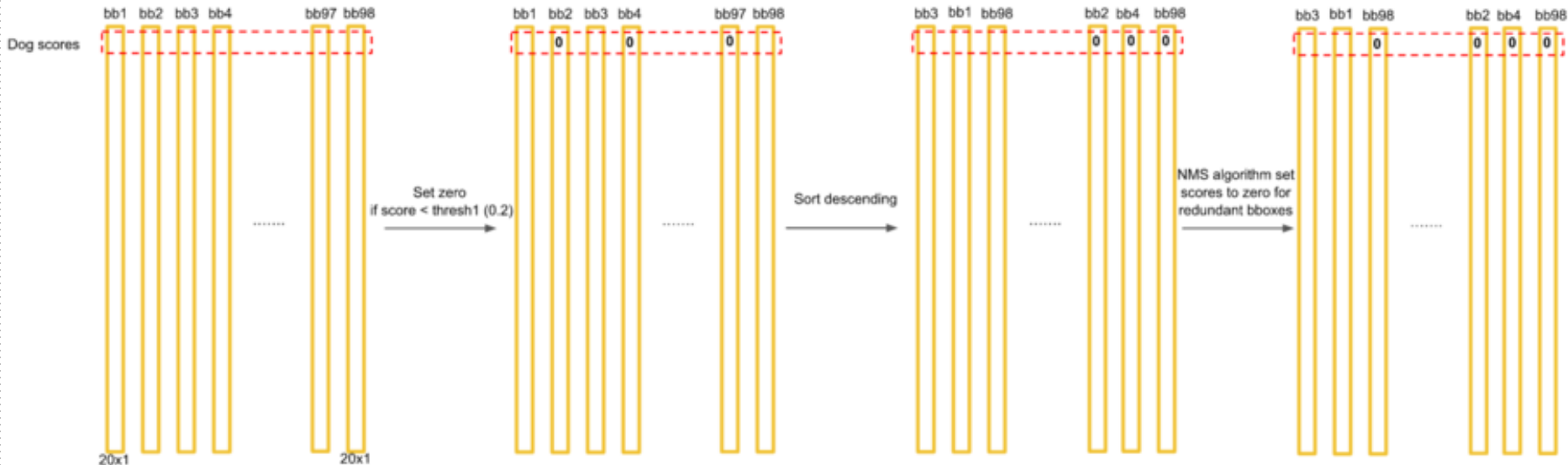


- 클래스 20개의 점수가 [0.1 , 0.1, 0.5, 0, 0, 0.25] 이렇게 나올 텐데 0.2 보다 작으면 그것은 해당클래스는 절대 아닐거라고 판단하였다.

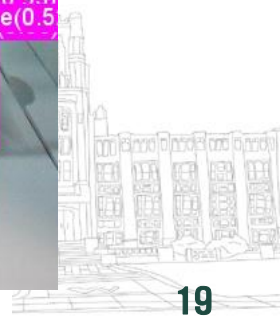


YOLO(You Only Look Once)

• 경계 박스 결정 과정

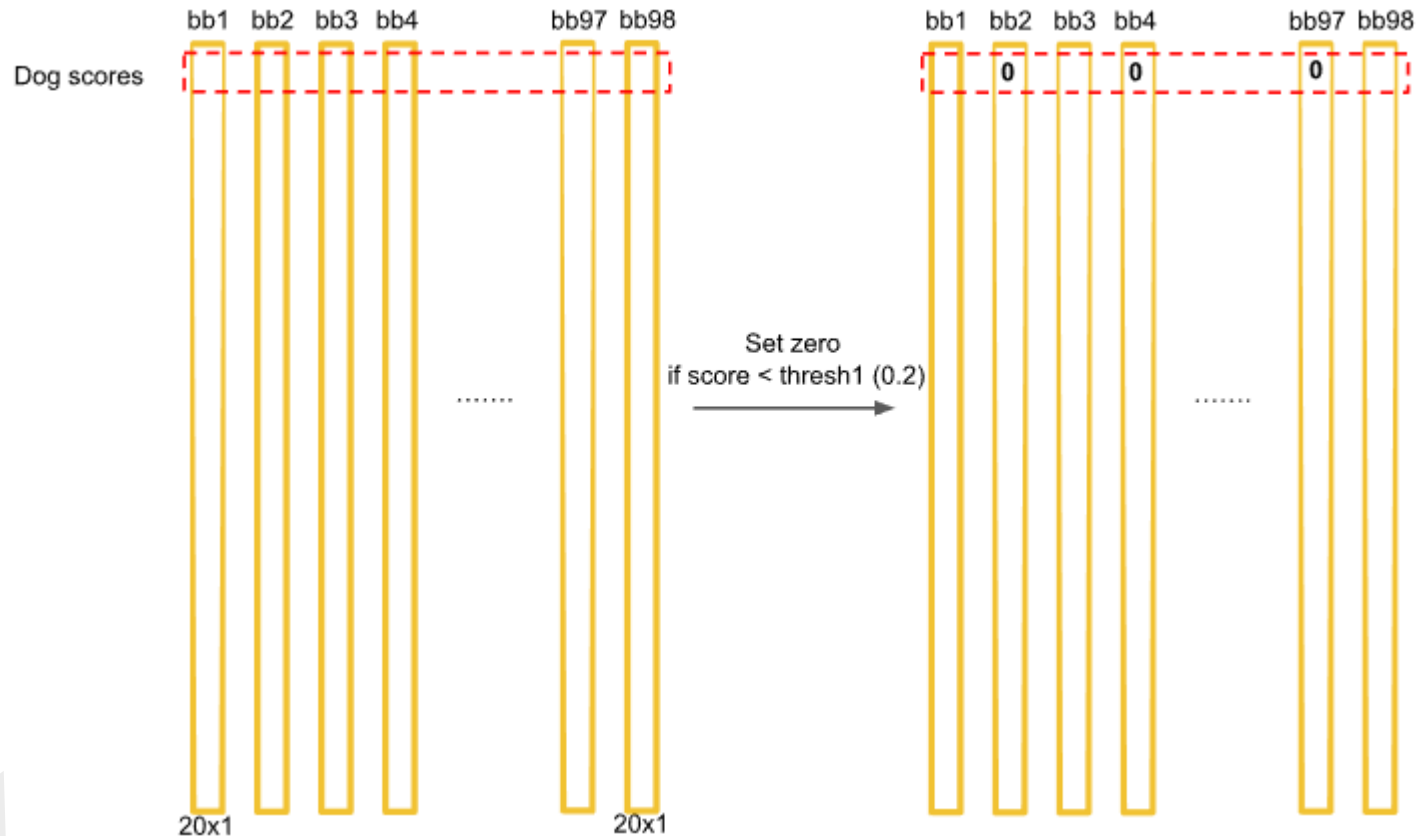


- 클래스 첫번째 자리는 Dog 일 확률이다. Dog일 확률을 높은 값부터 낮은 값으로 정렬한다.
- 만약 위의 98개의 클래스 점수 표 중에서 97개가 0이고 하나만 0이 아니면 해당 오브젝트가 그 위치에 있다는 것을 간단하게 알 수 있다. 하지만 0이 아닌 노란 박스가 여러개 있을 수 있다. 특히 오브젝트의 크기가 큰 경우 더욱 그러하다.
- 이것들을 제거하기 위해서 NMS(Nom-maximal suppression 비-최대값 억제) 알고리즘을 사용하여 중복이 되는 경계박스들을 제거하여 오직 하나의 경계 박스만 남긴다. NMS 는 여러 경계 박스들이 겹쳐있을 때에는 그중에서 최대값을 갖는 하나의 오브젝트만 빼고 나머지를 지운다(억제한다). 그러나 여러 경계 박스들이 겹쳐있지 않을 때에는 각 경계 박스들은 서로 다른 오브젝트의 경계 박스일 것이라고 생각하고 경계박스들을 그냥 둔다.
- 테스트 결과 NMS 을 사용하면 mAP 성능이 2-3% 상승한다. NMS를 사용하지 않더라도 충분한 성능이 나오므로 NMS가 반드시 필요한 것은 아니라고 한다.



YOLO(You Only Look Once)

- 스레시홀드 0.2를 사용하여 신뢰도가 최소 기준보다 낮은 클래스에 대해서는 0을 셋팅한다.

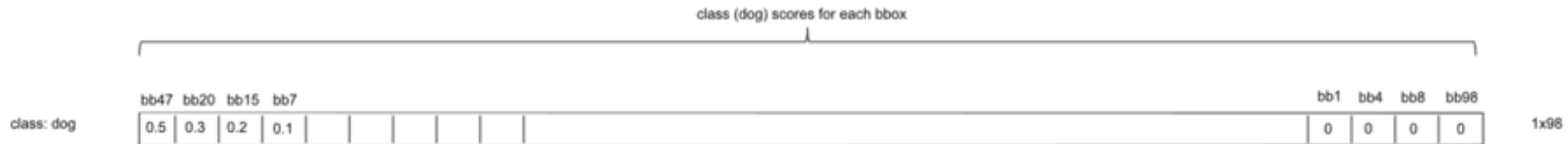


- YOLO 는 7*7 개의 그리드 셀에서 각각 2개의 경계박스 후보를 예측한다.
- 그러므로 총 98개의 경계박스를 예측한다.
- 20개의 클래스 중에서 우선 dog 만 살펴보자
- 우선 클래스의 신뢰도 점수가 0.2보다 작은 것은 모두 0으로 셋팅한다.
- 경계 박스의 신뢰도 점수가 어떤 클래스(예를 들어 개)에 대해서 신뢰점수가 낮다면 그 경계박스안에 뭐가 있는지는 아직 모르겠지만 최소한 그 클래스(개)는 아닐 것이라고 판단한다.

YOLO(You Only Look Once)

- 개에 대한 경계박스 중복 제거하기, 위그림의 빨간 점선 박스 부분

Non-Maximum Suppression: intuition

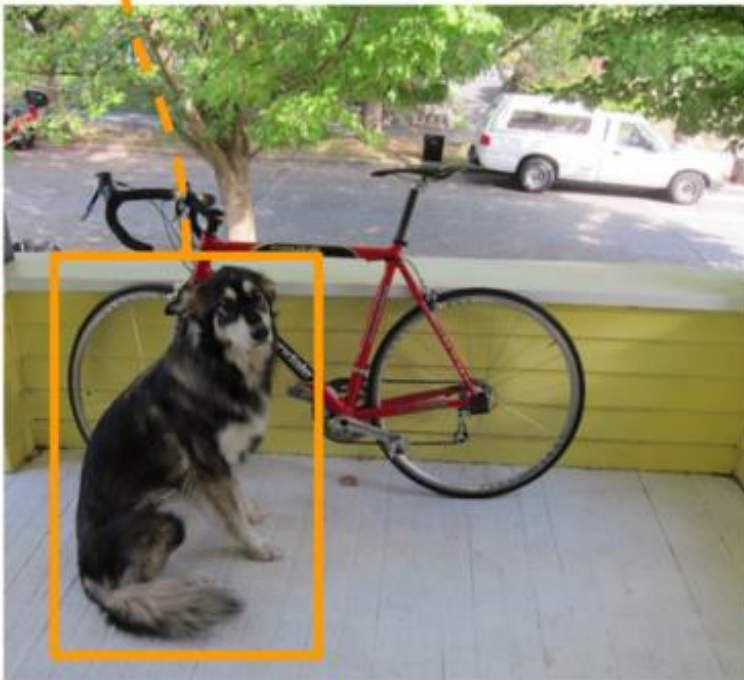


- dog 에 대한 bbox confidence 텀의 값을 큰것부터 작은것으로 내림차순 정렬하면 위와 같이 1x98 크기의 배열을 얻게 된다. 위 그림에서 0.2보다 작은 것은 모두 지웠다.(0.1은 참고 데이터로 생각됨)
- 신뢰도가 0.2보다 작은 것은 모두 지운 후에 개일 확률이 높은 경계 박스가 3개 남았다.



YOLO(You Only Look Once)

- NMS(개 후보 경계 박스 3개)

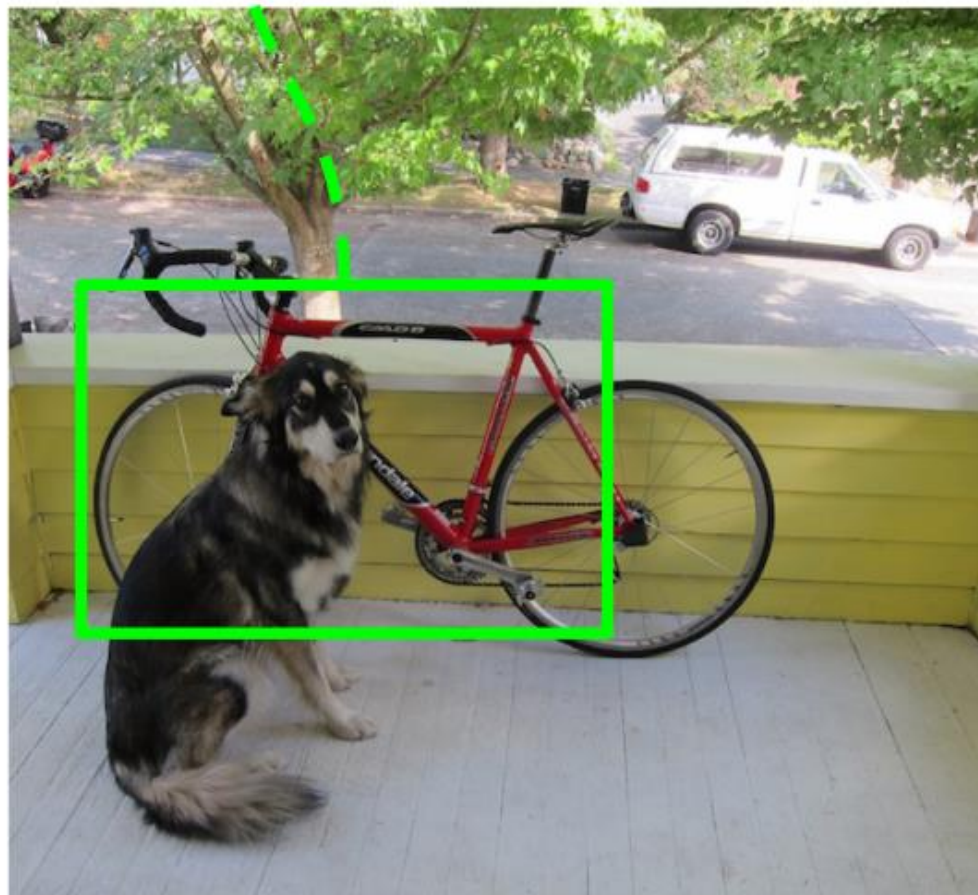


- 위 그림에서 개에 대해 가장 높은 경계박스 신뢰 점수를 갖는 그리드 셀은 bb47 이었다. 그 그리드 셀의 경계박스 예측 좌표를 오렌지 색으로 표시했다.

YOLO(You Only Look Once)

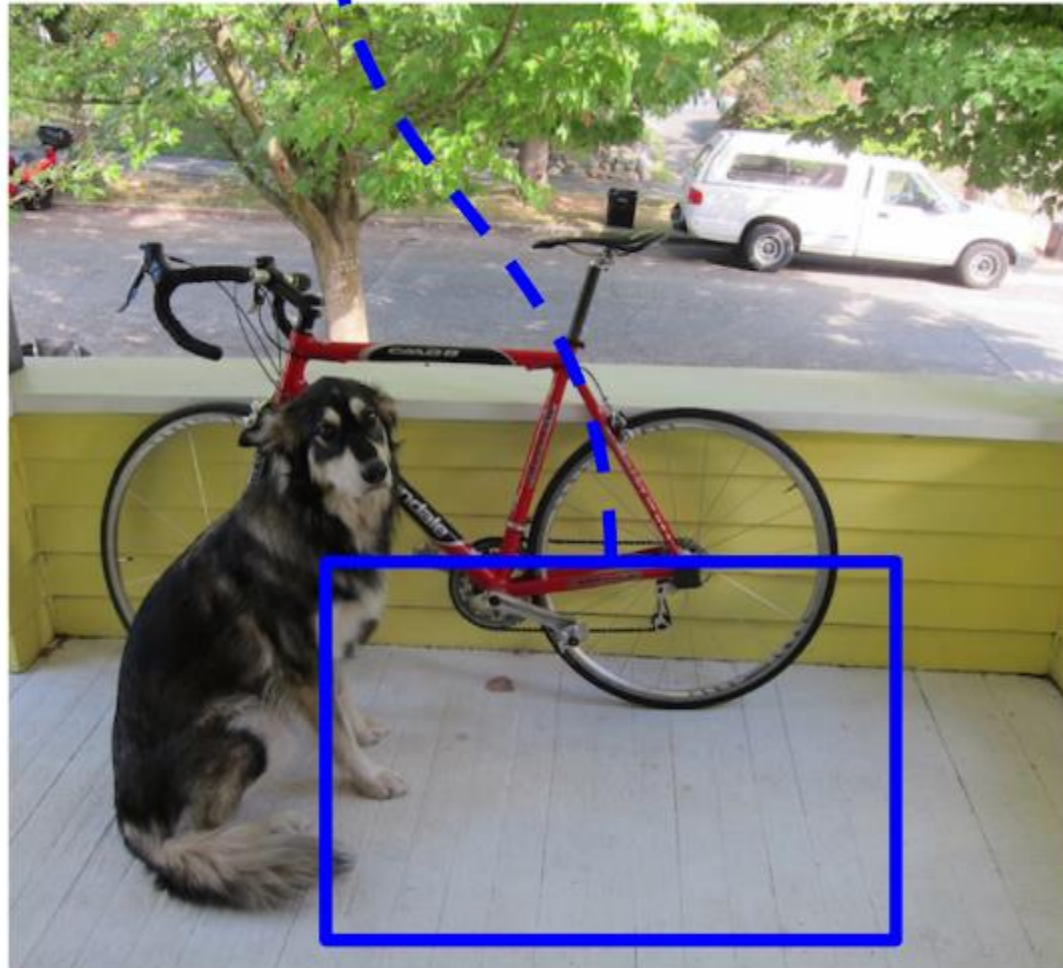
- 초록색 경계박스도 개를 어느 정도 잘 포착했다. 그러나 개의 다리와 엉덩이 부분은 포착하지 못했기 때문에 오렌지색 경계 박스보다는 나쁘다.

	bb47	bb20	bb15	bb7						
class: dog	0.5	0.3	0.2	0.1						



YOLO(You Only Look Once)

	bb47	bb20	bb15	bb7						
class: dog	0.5	0.3	0.2	0.1						



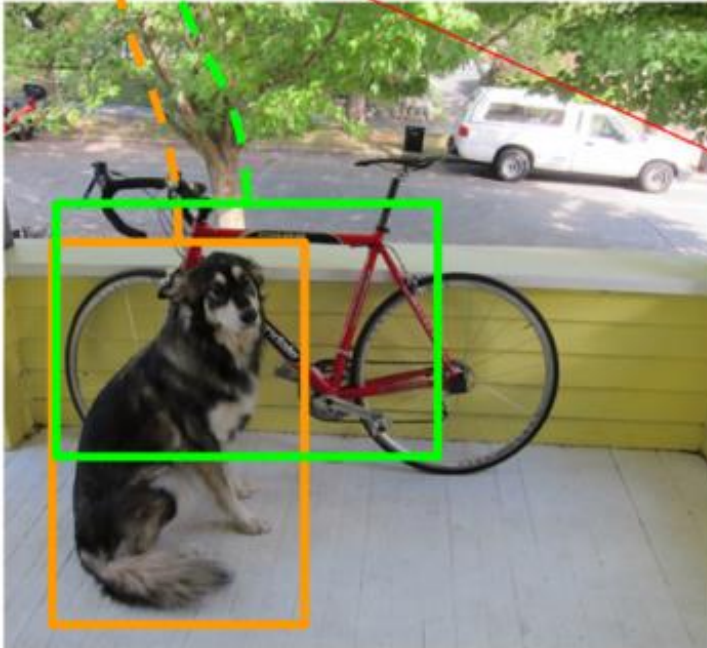
- 파란 경계 박스 안에는 겨우 개의 앞발만 나타나 있다.



YOLO(You Only Look Once)

- 중복 경계박스 지우기

	bb47	bb20	bb15	bb7						bb1	bb4	bb8	bb98		
class: dog	0.5	0	0.2	0.1							0	0	0	0	1x9



If $\text{IoU}(\text{bbox_max}, \text{bbox_cur}) > 0.5$ then set 0 score to bbox_cur .

In this case: set to 0.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

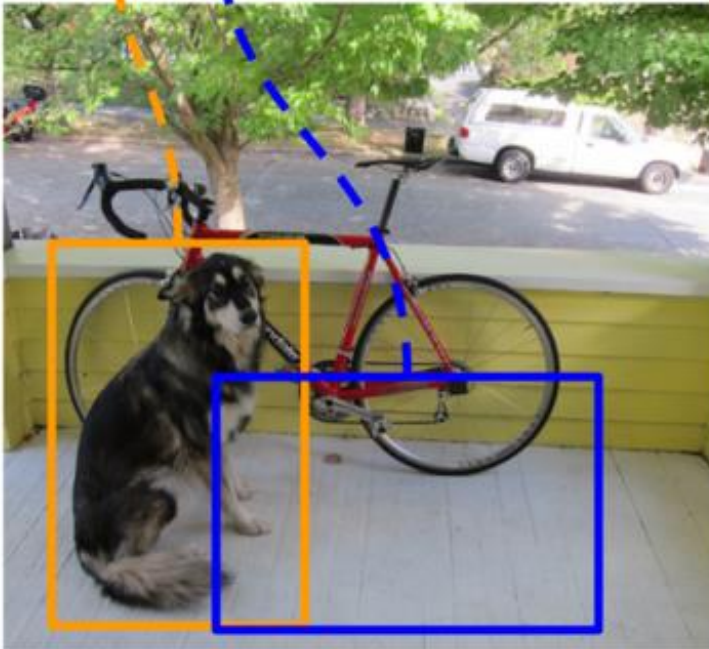
- 현재 경계박스 신뢰도가 가장 높은 첫번째 값 0.5 가 bbox max 이다. 두번째로 높은 신뢰도를 갖는 초록색 경계박스의 값은 원래 0.3 이었다. 그런데 두 경계박스의 겹치는 부분(IOU)이 50%가 넘는다. 이런 경우 둘 중 하나는 중복일 것이라고 판단, 둘 중 신뢰도가 낮은 것을 지워준다. 그래서 0.3의 신뢰도 값을 0 으로 설정한다.

YOLO(You Only Look Once)

class: dog

	bb47	bb20	bb15	bb7																		bb1	bb4	bb8	bb98
	0.5	0	0.2	0.1																		0	0	0	0

1x



Go to next `bbox_cur`.

If $\text{IoU}(\text{bbox_max}, \text{bbox_cur}) > 0.5$ then set 0 score to bbox_cur .

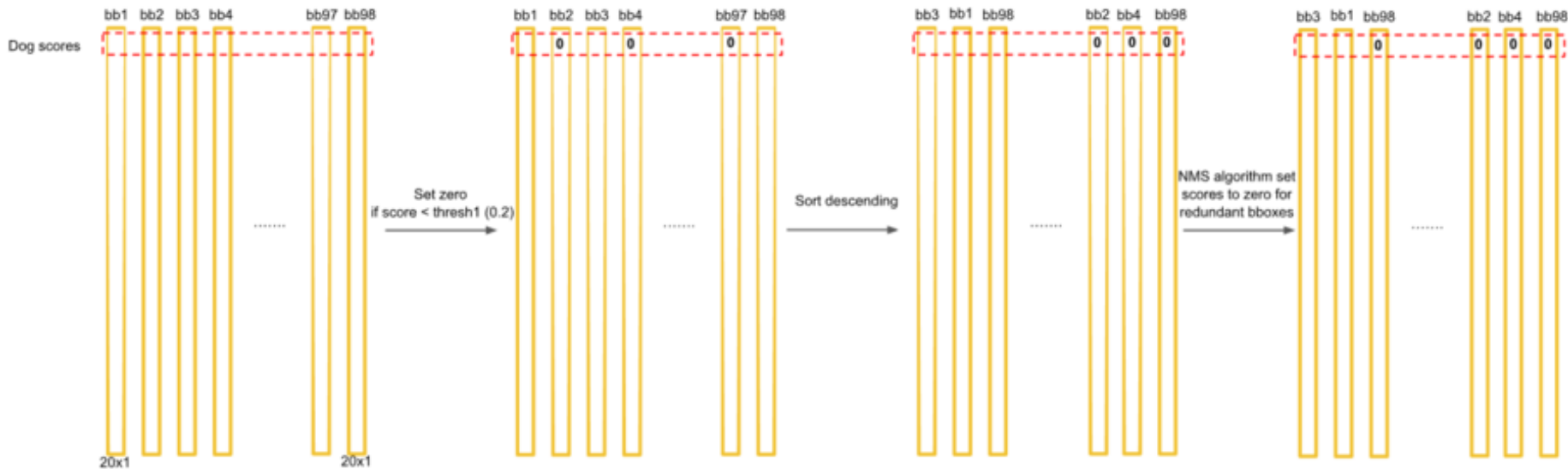
In this case: continue.

- 개에 대해서 3번째로 높은 신뢰도를 갖는 파란 경계 박스는 신뢰도가 가장 높은 오렌지색 경계 박스와 겹치는 부분이 50% 가 넘지 않는다. 그렇다면 파란색 경계 박스 안에는 좌측의 개가 아닌 또 다른 개가 있을 수도 있다고 판단한다. 이런 경우는 신뢰도 값을 건드리지 않고 그냥 건너 뛴다



YOLO(You Only Look Once)

- 개에 대해 중복 경계 박스 지우기

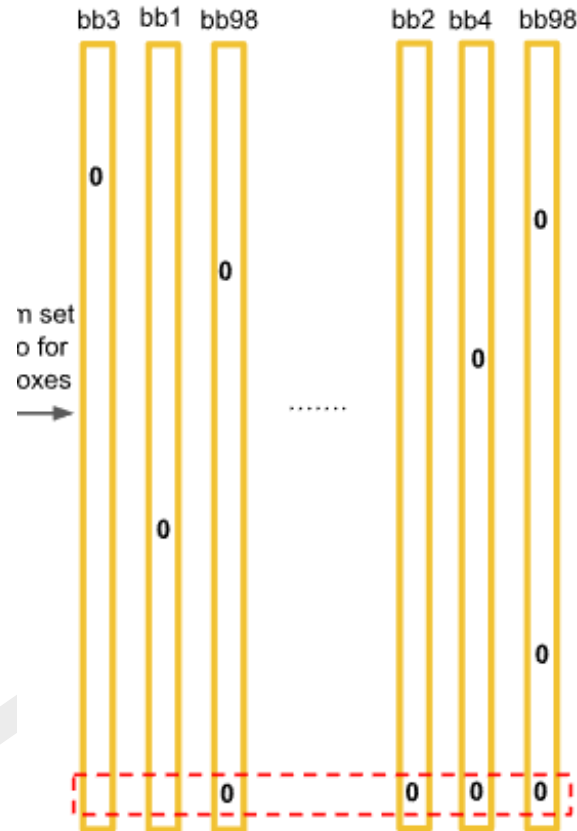


- 지금까지는 개에 대해서만 중복 경계 박스를 지웠다. 개에 대해 정렬한 1x20(첫번째 행)에서는 경계 박스 2개(오렌지색과 파란색)만 남고 나머지 18개는 모두 0이 되었다.



YOLO(You Only Look Once)

- 고양이 클래스에 대해 중복 경계 박스를 지운 결과
 - 이제 남은 19개의 클래스(두번째 행부터 마지막 행까지)에 대해서도 같은 작업을 해준다.

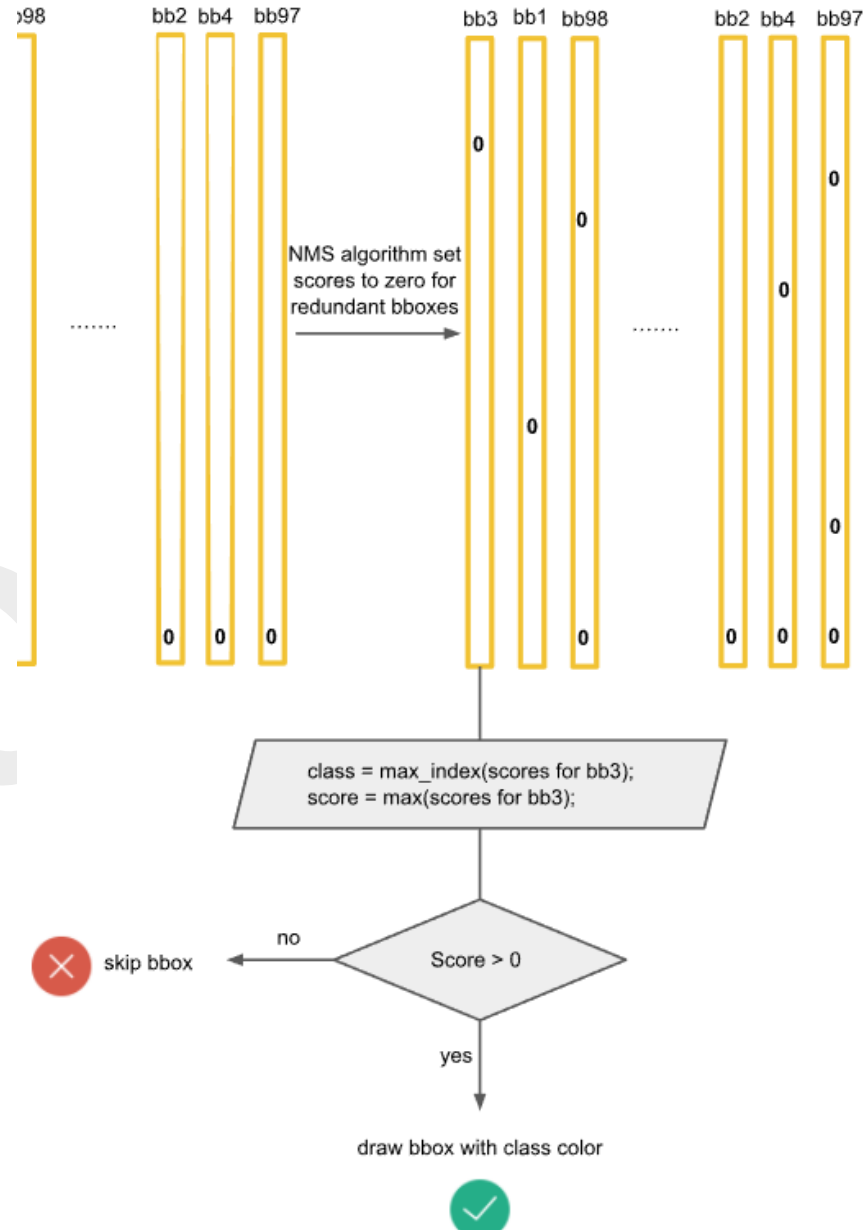


- 하지만 아직 한단계가 더 남았다. 사진에서는 개가 1마리 뿐이 없었는데, 개에 대한 경계박스가 오렌지색, 파란색 2개가 남아있다. 파란색 경계 박스에는 개가 없었다. 이것을 지워주어야 한다.



YOLO(You Only Look Once)

- 최종 경계 박스 결정하기



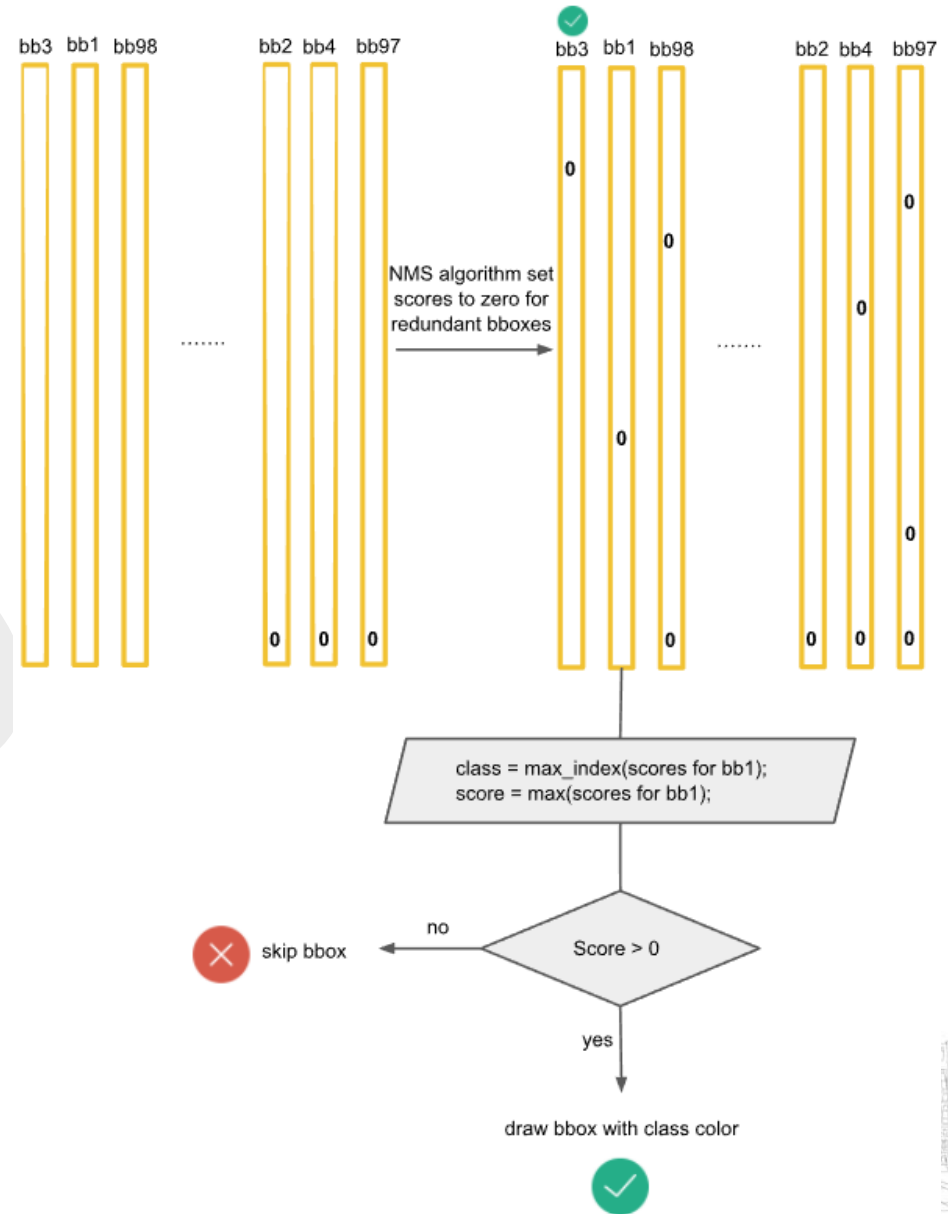
YOLO(You Only Look Once)

- bb3 는 20개 클래스에 대한 신뢰점수를 가지고 있다. 20개 신뢰점수 중 가장 큰 값을 가지는 클래스가 바로 경계 박스의 컬러이다. 만약 가장 큰 값이 0인 경우, 즉 해당 그리드 셀에서의 클래스 신뢰도가 전부 0이라면 bb3 에는 오브젝트가 없는 것이다. 그러면 당연히 경계 박스를 그리지 않는다.(skip)
- 이런 과정을 거쳐서 각 그리드 셀에서는 최대값을 갖는 클래스 하나와 또다른 클래스 하나 총 2개가 나타날 수 있다. 만약 한 그리드 셀에서 2개의 클래스가 검출되었다면 오브젝트들이 겹쳐있을 확률이 높을 것이다. 하나의 그리드 셀에서 클래스가 같은 2개의 오브젝트는 나올 수 없는 구조이다.
- 위의 예에서 개일 확률이 있는 2개의 경계박스중에서 오렌지색만 진짜이고 파란색 경계박스는 잘못된 것이었다. 파란색 경계박스는 신뢰점수가 0.5 이하이므로 나중에 지워진다.



YOLO(You Only Look Once)

- 전체 98개의 경계 박스에 대해서 같은 작업을 한다.



YOLO(You Only Look Once)

- 최종적으로 3개의 경계박스만이 남았다. 경계 박스의 클래스는 컬러로 나타냈다

